Faculteit Wetenschappen
Vakgroep Zuivere Wiskunde en Computeralgebra

Extending a first order predicate calculus with partially defined iota terms

Geert Vernaeve

Promotor:
Prof. Dr. A. Hoogewijs

Proefschrift voorgelegd aan de Faculteit Wetenschappen
tot het behalen van de graad van Doctor in de Wetenschappen: Wiskunde

Acknowledgements

There are so many people who have had a direct or indirect influence in the creation of this thesis, that every enumeration attempt is doomed to be incomplete: my promotor prof. Hoogewijs, colleagues, university staff, friends, teachers, parents, family, ...

Many thanks to all of you for steering me through a path full of coincidences and unexpected twists towards the fascinating field of formal logic, which is still is far less known than it deserves to be[1], and lies right in the intersection of two of my favourite interests: mathematics and computer science, and for your friendship and support during the whole endeavour.

[1] Our department *still* bulks with finite geometers.

Chapter 1

Introduction

1.1 An obstinate misconception

Since its advent in the 1940's, the programmable digital computer has been used for mathematical *numeric* calculations. Somewhat less known it is also perfectly possible to program such a device to perform *symbolic* calculations and even proofs.

For example, if we want to prove that $1 + 2 + 3 + \cdots + n = \frac{n \cdot (n+1)}{2}$, conventional wisdom seems to be that while we can write a computer program along the lines of

```
PRINT "Value of n? "
READ n
sum = 0
counter = 1
WHILE counter <= n DO
  sum = sum + counter
  counter = counter + 1
DONE
PRINT "1+2+...+n = " sum
PRINT "n*(n+1)/2 =" n*(n+1)/2
```

which yields

```
Value of n? 3
1+2+...+n = 6
n*(n+1)/2 = 6

Value of n? 10
1+2+...+n = 55
```

```
n*(n+1)/2 = 55

Value of n? 100
1+2+...+n = 5050
n*(n+1)/2 = 5050

...
```

using which we can verify the formula for each *individual* value of n, it is impossible to use a computer to check the formula for *all* values of n. To be certain that this formula holds for all n, still according to conventional wisdom, a *human* needs to supply a *proof*, an argument, to convince one that the formula indeed holds for all n. For example, one could observe

$$\begin{array}{rccccccccccc} & 1 & + & 2 & + & 3 & + & \ldots & + & (n-1) & + & n \\ + & n & + & (n-1) & + & (n-2) & + & \ldots & + & 2 & + & 1 \\ \hline & n+1 & + & n+1 & + & n+1 & + & \ldots & + & n+1 & + & n+1 \end{array}$$

and hence, $2 \cdot (1 + 2 + \cdots + n) = n \cdot (n + 1)$, from which the initial formula easily follows.

At first sight, this conventional wisdom seems correct: how would we be able to program a computer to process mathematical proofs, of which the example above is just a very simple one?

1.2 The QED Dream

However, one of the achievements of mathematics in the late 1800's is the realisation that almost all of mathematics can, at least in principle, be derived from a very small set of axioms using a very small set of so called deduction rules. In practice, this is rather tedious: Russell later said that his mind never fully recovered from the strain of writing *Principia Mathematica*, in which this is actually done for set theory, cardinal numbers, ordinal numbers, and real numbers.

With the advent of the computer, the venture of doing mathematics formally has become a practical possibility. There are currently tens of systems available to do this. (See for example `http://www.cs.ru.nl/~freek/digimath/index.html`, which lists 163 systems in the categories 'proof checker' and 'theorem prover'.) But why would one embark on such an arcane mission? The reasons can be manifold:

- *Correctness:* if we know that the program correctly implements the axioms and deduction rules, all theorems derived in it are correct. This is

important when one wants to formally verify that a computer program obeys a certain property (e.g. for use in medical or avionic devices). It is also pleasant when developing new proofs, since it is hard to be sure that a proof of a theorem is completely correct. Illustrations of the difficulty of validating mathematical proofs are for example

  - the Fundamental Theorem of Algebra, which purportedly was proven in 1746 by d'Alembert and later also by Euler, de Foncenex, Lagrange, Laplace, Gauss, ..., until finally only in 1806 Argand supplied the first correct proof,
  - the Four Color Theorem, of which incorrect proofs were given in 1879 by Kempe and 1880 by Tait, which were both believed to be correct for 11 years, and it was not until 1976 before a completely correct proof was found.

- *Rigour:* for most formal systems, a precise semantics is available, and hence the meaning of a mathematical statement in the system is unambiguously clear. This is in sharp contrast with computer algebra systems (CAS), such as Maple, Mathematica, Derive, ... which are notoriously sloppy in this regard. An interesting insight in the problems surrounding the construction of mathematically correct computer algebra systems can be found in [Fateman 1994].

- *Assistance:* if we use the computer to do mathematics, it could help us by suggesting existing possibly useful lemmas, doing routine checks and even automatically filling in some parts of the proof.

- *Retrieval:* currently, mathematical proofs in journals are increasingly electronically available, but these are informal proofs, and even formulae are in most of the cases stored in a form which is essentially a consecution of typographical glyphs from which it is difficult to programmatically extract the mathematical structure. For example, it is almost impossible to search for "a formula containing the integral of a polynomial in $\sin(x)$". If the theorems would be stored in a formal system, such queries would be significantly facilitated. In a world in which yearly 55000 mathematical articles [Impens] containing about 200000 theorems [Hazewinkel 2003] are published, it is increasingly interesting to be able to scan them effectively for theorems which would be of interest for the problem at hand.

- ... There are many more reasons; see for example the *QED Manifesto* [QED 1994] for a thorough examination of the usefulness of formalising mathematics.

1.3 An informal introduction to formal logic

1.3.1 An example of a formal proof

A small formal proof might look like this:

1.	$\forall y \neg p(x,y) \vdash \forall y \neg p(x,y)$	ass
2.	$\forall y \neg p(x,y) \vdash \neg p(x,y)$	$\forall$-elim (1)
3.	$\forall x p(x,y) \vdash \forall x p(x,y)$	ass
4.	$\forall x p(x,y) \vdash p(x,y)$	$\forall$-elim (3)
5.	$\forall y \neg p(x,y), \forall x p(x,y) \vdash \neg \forall x p(x,y)$	contra (2,4)
6.	$\neg \forall x p(x,y) \vdash \neg \forall x p(x,y)$	ass
7.	$\forall y \neg p(x,y) \vdash \neg \forall x p(x,y)$	rem (5,6)
8.	$\forall y \neg p(x,y) \vdash \forall y (\neg \forall x p(x,y))$	$\forall$-in (7)
9.	$\neg \forall y (\neg \forall x p(x,y)) \vdash \neg \forall y (\neg \forall x p(x,y))$	ass
10.	$\forall y \neg p(x,y), \neg \forall y (\neg \forall x p(x,y)) \vdash \neg \forall y (\neg p(x,y))$	contra (8,9)
11.	$\neg \forall y (\neg p(x,y)) \vdash \neg \forall y (\neg p(x,y))$	ass
12.	$\neg \forall y (\neg \forall x p(x,y)) \vdash \neg \forall y (\neg p(x,y))$	rem (10,11)
13.	$\neg \forall y (\neg \forall x p(x,y)) \vdash \forall x (\neg \forall y (\neg p(x,y)))$	$\forall$-intro (12)

where p is a unary predicate symbol (for a rigorous description of the syntax, see §2.1).

We see that a formal proof is a list containing a number of **sequent**s, which consist of a list of formulae called the **antecedent** and a single formula called the **conclusion** with a "$\vdash$" symbol in between. Note that the order of formulae in the antecedent is not significant, so line 5 could just as well have read

5.	$\forall x p(x,y), \forall y \neg p(x,y) \vdash \neg \forall x p(x,y)$	contra (2,4)

Each sequent is 'obtained' by applying a **deduction rule**. For example, in the first line of the proof, we obtained the sequent "$\forall y \neg p(x,y) \vdash \forall y \neg p(x,y)$" using the deduction rule called 'ass(umption)'—see §2.3 for a list of all available rules. In the second line, we obtained the sequent "$\forall y \neg p(x,y) \vdash \neg p(x,y)$" using the deduction rule '$\forall$-elim'. This rule needs as 'input' or **premise** a single sequent obtained earlier in the proof, in this case the sequent from line 1. In this fashion, we continue to produce sequent after sequent until the proof ends.

In this way, a formal system could be considered as a kind of game, where the challenge is to find a way to produce a given sequent (in our case "$\neg \forall y (\neg \forall x p(x,y)) \vdash \forall x (\neg \forall y (\neg p(x,y)))$") using the deduction rules.

1.3.2 Semantics

Of course, the rules of the calculus are not chosen at random: we can attach a *semantics*, *meaning* or **interpretation** to a formula in a certain **domain** ω.

For example, if we set ω to be the natural numbers, we could interpret $p(x, y)$ as "$x \geq y$". Having fixed ω and the interpretation of p in this way, using §2.2, we can interpret the formula $\neg\forall y(\neg\forall x p(x, y))$ as "it is not the case that for each natural number y, no natural number x is larger than or equal to y", i.e., "there exists a natural number y such that every natural number x is larger than or equal to y". In this interpretation, the formula is true or **valid** (the natural number 0 is a suitable candidate for y). Similarly, the formula $\forall x(\neg\forall y(\neg p(x, y)))$ is to be interpreted as "for each natural number x there exists a natural number y such that x is larger than or equal to y", which too turns out to be a valid formula (again zero is a suitable candidate for y).

We can choose another interpretation, where for example ω is the set of real numbers and $p(x, y)$ is interpreted as "x is the double of y". Then it turns out that the first formula is false (or **invalid**) since it interprets as "there exists a real number y such that every real number is equal to the double of y" and the second formula is valid since it interprets as "for each real number x there exists a real number y such that x is the double of y" ($\frac{x}{2}$ is a suitable candidate for y).

Hence, we observe that the interpretation of a formula depends on the interpretation chosen.

However, it turns out that the interpretation of some formulae can be independent of the interpretation. For example, the formula '$x = x$' is always valid, no matter which interpretation is used. Carrying this idea further, it is also possible that a formula is always valid, provided some other formula(e) is (are) valid too in the same interpretation. For instance, it turns out that $\forall x(\neg\forall y(\neg p(x, y)))$ is always valid whenever $\neg\forall y(\neg\forall x p(x, y))$ is valid, and we note this as

$$\neg\forall y(\neg\forall x p(x, y)) \models \forall x(\neg\forall y(\neg p(x, y)))$$

and we say that $\forall x(\neg\forall y(\neg p(x, y)))$ is a **consequence** of $\neg\forall y(\neg\forall x p(x, y))$. The corresponding sequent, $\neg\forall y(\neg\forall x p(x, y)) \vdash \forall x(\neg\forall y(\neg p(x, y)))$, is called a **sound sequent**.

The connection between the formal calculus and the semantics can now become apparent: it turns out that using the deduction rules, we only can produce sound sequents (the calculus is **sound**); conversely, each sound sequent can be derived (the calculus is **complete**).

1.3.3 Theories

The 'plain' first order logic as described above is able to reason about functions and predicates in general. Often, we have some specific functions and predicates in mind. For example, when discussing natural numbers, we will probably want to introduce a one-place function 'successor(x)' which we want to interpret as '$x + 1$', a constant (which we treat as 0-ary functions) '0', and so on. Of course, interpretations don't have to choose the set of natural numbers as their domain, and don't have to interpret 'successor' as the successor function. To try and force this, one adds **axiom**s, e.g., $\forall x(\neg(\text{successor}(x) = 0))$. Interpretations which interpret this particular axiom as valid, only allow interpretations of 'successor' for which the interpretation of '0' is not the successor of any element of the domain. This single axiom is not sufficient to fix the interpretation; for example, we need to add the axiom $\forall x \forall y((\text{successor}(x) = \text{successor}(y)) \Rightarrow x = y)$; intuitively, this expresses that each number has only a single predecessor.

Hence, in practice, we will add to the calculus extra sequents of the form $\vdash \alpha$ where α is an axiom; we call the resulting formal system a **theory**. One can for example consider the theory of natural numbers, the theory of sets, and so on.

1.4 Partial functions in mathematical practice

Earlier, we said there are tens of systems available in which one can do computerised formal mathematics. Why are these systems not commonplace among mathematicians if the advantages are so numerous? The reasons seem to be manifold (see e.g. Jones' *Critique of the QED Manifesto* [Jones 1995]). One of them is that we need the formal logic to reflect the common mathematical practice as close as possible. In systems to do computerised formal mathematics, there is a tendency *not to merely reconstruct existing mathematics in a formal framework, but to try and develop a new (better?) mathematics. [...] If we hope to interest many mathematicians [in actually using such a system], we need to accommodate existing mathematics. We are already trying to wreak one revolution by making the mathematicians formalise their work. Surely one revolution at a time is enough!* [Harrison 1996b]

An area in which formal logics are notoriously controversial is in their treatment of partially defined functions (see next section).

In mathematical practice, partially defined functions abound:

- Division in the real numbers, i.e., the function $\mathrm{div}(x, y) := \frac{x}{y}$ is only defined when $y \neq 0$.

- $\sum_{i=1}^{\infty} a_i$ is only defined when the series converges.

- $\lim_{n\to\infty} a_n$ is only defined when the sequence a_n is convergent.

- $\int f$ is only defined when f is (Riemann, Lebesgue, ...) integrable.

- $GF(q)$ is only defined when q is a prime power.

Note that even in a logic which does not explicitly cater for partially defined functions, we can express everything using relations instead (this is approach (b) in [Farmer 1990]). For example, instead of using the partial function mapping (x, y) to $\frac{x}{y}$, we can use the relation $\mathrm{Div}(x, y, z)$, which contains all triples (x, y, z) for which $\frac{x}{y}$ is defined and $\frac{x}{y} = z$. Hence, the formula $\mathrm{div}(x, y) = z$ becomes $\mathrm{Div}(x, y, z)$. This notation quickly becomes unwieldy. Hence, our task is not one of making the logic more expressive, but "to find a notationally efficient way of reasoning about partial functions that is reasonably faithful to mathematical practice and that upsets the framework of classical logic as little as possible" [Farmer 1990]. In other words, our aim is not being able to prove more theorems, but to prove them in a way that corresponds better to mathematical practice.

How does mathematical practice cope with potentially undefined expressions such as $\frac{1}{x}$ or $\sqrt{y}$? Almost all introductory mathematical textbooks present elementary mathematical logic as if it were two-valued; potentially undefined expressions are allowed only when one first proves that they are always defined. For example, one is allowed to talk about $\frac{1}{x}$ and $\sqrt{y}$ provided that one proves first that $x \neq 0$ and $y \geq 0$. Expressions such as $\frac{1}{0}$ are meaningless: one can write them down but is not allowed to use them or prove properties about them. The mathematical expression "$\frac{1}{0}$" is akin to the natural language expression "the brick is happy" in the sense that 0 possesses no inverse just like bricks have no mood; it makes no sense to try and discuss such nonsense statements.

We conclude that in mathematical practice one performs a balancing act between using potentially undefined terms (which seems to call for a three-valued logic in which expressions be true, false or undefined) and maintaining the illusion that one works in a two-valued setting.

1.5 Treating partially defined functions in a formal calculus

It will appear shortly that there are many different ways in which formal calculi have been adapted to handle partially defined functions. The multitude of different approaches already indicates that there is probably no "definitive" way in which once and for all partially defined functions can be handled, but the way in which they are handled depends on the application at hand. Each approach has its advantages and disadvantages—*there ain't no such thing as a free lunch with partial functions* as [Jones 1996] puts it.

For example, in computer science, it is very common that an operation is 'undefined', for example when trying to read a nonexistent file:

```
try {
  f = open_file_for_reading("nonexistent")
  do_something_with_file(f)
} catch (FileException) {
  PRINT "Could not read the file"
}
```

This differs quite from undefinedness in pure mathematics, where terms as $\frac{1}{0}$ are considered with suspicion. Here, the result of `open_file_for_reading()` can be 'undefined', but this is not a problematical situation since we explicitly state what has to happen when this occurs.

On the other hand, another kind of undefinedness can creep in computer programs: in most programming languages, the result of some operations is not specified. For example, in Java, `File.delete()` evaluates to `true` when the file is successfully deleted and to `false` otherwise, but if the file did not exist in the first place, the result depends on the particular Java implementation (so this corresponds to approach (c) in [Farmer 1990]). In case the specified file did not exist, we could consider this expression as 'undefined', since we do not want programs to depend on its exact value. (In theorem 23 we will prove formally that in our calculus, proofs don't 'depend on undefined values').

In this work, we extend the classical first order predicate calculus with identity (as formalised in e.g. [Hermes 1973]) by adding so-called ι-terms to the calculus, in a way which is analogous to [Hilbert & Bernays 1968]: the term '$\iota x(\varphi)$' is to be interpreted as 'the (unique) x for which φ is true'. Such ι-terms are only allowed to be used if one can show that the **uniqueness condition** $\exists! x(\varphi)$ holds. For example, in a theory describing real numbers, we can introduce the operation of subtraction given addition: '$x - y$' is just $\iota z(z + y = x)$; we can indeed show $\vdash \exists! z(z + y = x)$.

However, we extend this "classical" notion of ι-term to situations where there is not always a unique x satisfying φ but only when a certain condition ψ (the **domain formula**) holds. For example, if we want to introduce division in the example given above, we would like to proceed analogously and define $\frac{x}{y}$ as $\iota z(z \cdot y = x)$; however we can only show that $y \neq 0 \vdash \exists! z(z \cdot y = x)$; if $y = 0$ then either $x = 0$ and every z satisfies $z \cdot y = x$, or $x \neq 0$ and no z satisfies $z \cdot y = x$. We will show that allowing such ι-terms to be added to the calculus introduces no contradictions, provided one considers the obtained ι-term as undefined when the uniqueness condition is not met (i.e., we consider $\iota z(z \cdot y = x)$ as undefined if $y = 0$). We will denote this partially defined ι-term as $\iota z_{y \neq 0}(z \cdot y = x)$: in the subscript, we note when the ι-term is defined (i.e., its domain formula). The interpretation of this ι-term is still 'the unique z for which $z \cdot y = x$' when $y \neq 0$, but when $y = 0$, this ι-term will be interpreted as 'undefined'.

We will now discuss various ways in which formal calculi have been adapted to cope with partial functions.

- As already indicated in the previous section, one can avoid the problem by using a total $n+1$-ary relation instead of a partial n-ary function. (approach (b) in [Farmer 1990], approach 4 in [Jones 1996])

- One can move the problem into the syntax by considering expressions containing applications of functions outside their domain as not well-formed. This method has drawbacks (see approach (a) in [Farmer 1990]), one of which is that it is not in line with mathematical practice, in which expressions as $\frac{1}{x}$ are well-formed. In our calculus, a partially defined ι-term is always syntactically well-formed, even if we cannot deduce its uniqueness condition, e.g., $\iota x_{y>5}(x < y)$. However, we cannot prove anything about it, since otherwise, we would be able to infer its uniqueness condition (using the UC rule).

- One can model partial functions as total, but keep its value outside its domain unspecified (approach (c) in [Farmer 1990], approach 1 in [Jones 1996]). For example, terms such as $\frac{1}{0}$ are given a value but one cannot know which one it is, say $\frac{19}{17}$, and hence we would get the pathological theorem $\frac{0}{0} = 0 \cdot \frac{1}{0} = 0 \cdot \frac{19}{17} = 0$. The popular HOL system (`http://www.cl.cam.ac.uk/research/hvg/HOL/`) uses this technique. In our calculus, terms as $\frac{1}{0}$ are considered as undefined, avoiding the pathological theorems.

- One can make a partial function total by giving it a 'convenient' value on points outside its domain. For example, we could set $\frac{x}{0} := 0$. Here

we get the same risk of introducing pathological theorems as in the previous approach. ACL2 is a system using this technique.

- One can use a many-sorted logic, where a sort is introduced for each domain ((d) in [Farmer 1990]). This can lead to a proliferation of sorts ; however, given a sufficiently rich type theory, this approach seems manageable, as witnessed by the well known PVS, Coq and NuPRL systems. Our calculus is one-sorted, keeping the calculus simple. This has the advantage that we can use standard ZFC set theory instead of type theory. The use of standard set theory makes a system possible which adheres more to common mathematical practice than type theories. Interestingly, PVS (`http://pvs.csl.sri.com/`) has also a notion of 'definedness in context' and hence a notion of evaluation order, which in fact inspired the handling of undefinedness in our system. PVS has a separate mechanism of Type Correctness Conditions (TCC's) to ensure that undefinedness does not occur when evaluating. For example, when one sets out to derive $\vdash x \neq 0 \Rightarrow x \cdot \frac{y}{x} = y$, the PVS system generates the TCC $\vdash x \neq 0 \Rightarrow x \neq 0$, which we have to derive first. This parallels the requirement of the premise $\Sigma; \vdash_\iota \mathbf{\Delta}(\alpha)$ for the assumption rule in our calculus; note that $\mathbf{\Delta}\big(x \neq 0 \Rightarrow x \cdot \frac{y}{x} = y\big)$ is precisely the formula in the TCC. In contrast to PVS, which is a higher order logic, our calculus shows that we can apply similar ideas to a first order logic (this is also done in [Wiedijk & Zwanenburg 2003]—see below).

- One can extend the range of each partial function with a special 'error' value ((e) in [Farmer 1990]). For example, the division function would be modeled as a function from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R} \cup \{\text{error}\}$. This approach seems more suitable to reason about computer programs than abstract mathematics. For example, $\frac{1}{\frac{1}{x}}$ is problematical because $(1, \text{error})$ is not in the domain of our division function as sketched earlier; we would have to modify it into a function from $\mathbb{R} \cup \{\text{error}\} \times \mathbb{R} \cup \{\text{error}\}$ to $\mathbb{R} \cup \{\text{error}\}$ and it is easy to see that other functions such as addition, ... would be 'infected' rapidly too.

- Use 'error' values, but do not quantify over them ((f) in [Farmer 1990]). The 'error' value is given an inferior status: free variables range over the whole domain including error values, but quantified variables do not range over error values. This alleviates some of the problems with the previous approach, but the different treatment of free and bound variables is contrary to mathematical practice. In our calculus, free and bound variables are treated equally and never are undefined.

- Use a three-valued logic in which terms and formulae are allowed to have the value 'undefined'; if a formula or term contains an undefined term, it is itself undefined ((g) in [Farmer 1990]). This is akin to our approach, but in our calculus, undefinedness does not have to 'propagate upwards'. For example, in our calculus, the formula $x \neq 0 \Rightarrow \frac{1}{x} \neq 0$ is true, even when x is interpreted as 0, which causes the formula to contain the undefined subterm $\frac{1}{0}$.

- Use a three-valued logic in which terms are undefined when they contain an undefined subterm but formulae are always defined. An atomic formula containing an undefined subterm is considered false ((h) in [Farmer 1990]). The IMPS system uses this approach. This approach has a number of interesting features. For example, over the reals, $\sqrt{x} = 2 \Leftrightarrow x = 4$ is a theorem, even if $x < 0$, since for example $\sqrt{-2} = 2$ is considered false because it contains the undefined term $\sqrt{-2}$, and $-2 = 4$ is of course false too. Another example is the theorem $x = \frac{z}{y} \Rightarrow x \cdot y = z$, which holds even when $y = 0$, since then we have that $x = \frac{z}{0}$ is false because $\frac{z}{0}$ is undefined and hence the whole implication is true. However, we also get pathological theorems such as $\neg(\frac{1}{0} = \frac{1}{0})$. In our calculus, formulae can be undefined, avoiding the $\neg(\frac{1}{0} = \frac{1}{0})$ pathological theorem.

- Use a three-valued logic in which terms can be undefined but formulae are always defined, introducing two forms of equality: existential equality (for which undefined $=_\exists$ undefined) and strong equality (for which undefined $\neq$ undefined) (approach 2 in [Jones 1996]). Not only is having two different notions of equality confusing, other relational operators get infected too and also need two variants in the logic.

- Use a three-valued logic in which terms and formulae can be undefined. The propositional operators are the McCarthy conditional operators. These operators have slightly different properties compared to their two-valued counterparts; for example, conjunction is no longer commutative. Hence, both a commutative and a conditional variant of the propositional operators is used (approach 3 in [Jones 1996]).

  Our calculus uses the McCarthy operators, but we do not see the need to introduce extra commutative variants: the conjunction does not commute only when undefined terms are involved (see property 35) so we do not see this as a major problem.

- Use a three-valued logic in which terms and formulae can be undefined. The propositional operators can be thought of as evaluating

their operands in parallel and delivering a result as soon as enough information is available (this is the LPF approach [Jones 1996]).

The drawback of this approach is that for example $1+\frac{1}{0} = 1+\frac{1}{0} \Rightarrow 1 = 1$ is a theorem, which is contrary to the idea that undefined terms are nonsensical; we don't like to have theorems depend on properties of undefined terms.

- Use a three-valued logic with an extra 'is defined' operator (as in PPC [Hoogewijs 1977]), on equal footing with the other logical symbols such as $\forall$, &, $\neg$, ... This complicates the logic by introducing an extra operator. Undefined values are treated more on an equal footing with true and false values and there is no notion of evaluation order (the conjunction of PPC is symmetric in both arguments). For example, with the usual definitions, $y = \frac{1}{x} \vdash y = \frac{1}{x}$ is a valid sequent in PPC, whereas it is invalid in our calculus (when $\mathcal{I}(x) = 0$, we encounter an undefined value).

  In our calculus, the operator 'is defined' ($\mathbf{\Delta}$) is a metalogical operator (so it is not present in proofs or formulae), which makes the logic simpler for the user; undefined values are not allowed to occur when evaluating a valid sequent and there is a notion of evaluation order.

  Interestingly, in our calculus, both the Herbrand deduction rule (DdRu1 and DdRu2) and the Modus Ponens rule hold, which is not possible in PPC.

- Add PVS-like domain conditions to standard first order logic [Wiedijk & Zwanenburg 2003]. This approach is similar to our calculus. However, our calculus is an extension of the standard first order calculus (as given in [Hermes 1973]); domain conditions are part of the formal proof itself ([Wiedijk & Zwanenburg 2003] requires one to prove those separately) and functions and predicates do not have to be **strict** with respect to undefinedness: we can define a function or predicate that is defined when some of its arguments are undefined.

See also [Harrison 1996b] §2.5 where some of these approaches are discussed.

1.6 Introduction to our calculus

The logic which we will develop, which we will call the PITFOL calculus (for Partial Iota Terms in First Order Logic), is a superset of the well-known

classical first order predicate calculus with identity (see §2 for a short description). We then add, as already announced, terms of the form $\iota x_\psi(\varphi)$ to the calculus, adapt the deduction rules as necessary and add new rules. The resulting system has 17 deduction rules: 6 are transferred unmodified from the classical calculus, 5 are slightly extended and 6 are new. The extension is done in such a way that the rules do not change when applied to formulae of the classical calculus; only when ι-terms are used, the rules deviate from the classical rules. Moreover, the extension is **conservative**: if we obtain a theorem using the new rules and terms which is also expressible in the classical calculus, this theorem must already have been provable in the classical calculus. In this way, we have a calculus that "upsets the framework of classical logic as little as possible". This illustrates that our aim is not to be able to prove more theorems; the introduction of ι-terms however helps in finding proofs that follow mathematical practice better.

To illustrate how our PITFOL calculus handles partially defined terms, let us reconsider the term $\iota z_{y\neq 0}(z \cdot y = x)$, which as we showed above can be interpreted as $\frac{x}{y}$. If we were to use the classical equality rule, we would obtain the sequent

$$\vdash \iota z_{y\neq 0}(z \cdot y = x) \;=\; \iota z_{y\neq 0}(z \cdot y = x)$$

which seems harmless enough: it expresses that $\frac{x}{y} = \frac{x}{y}$ is a theorem. If we now interpret this sequent in a model where y is interpreted as 0, the term $\iota z_{y\neq 0}(z \cdot y = x)$ is interpreted as "undefined", and so is the formula $\iota z_{y\neq 0}(z \cdot y = x) \;=\; \iota z_{y\neq 0}(z \cdot y = x)$. This is a situation which of course does not occur in the classical first order predicate calculus, which is two-valued (formulae are either true or false, never undefined). However, in our PITFOL calculus, sequents can only be sound when one *never* has to consider undefined terms or formulae when interpreting it, so the given sequent is not sound.

The modified equality rule yields

$$y \neq 0; \vdash_\iota \iota z_{y\neq 0}(z \cdot y = x) \;=\; \iota z_{y\neq 0}(z \cdot y = x)$$

provided we are able to produce the uniqueness condition $y \neq 0 \vdash_\iota \exists! z(z \cdot y = x)$, which in a reasonable theory about real numbers should pose no problems. The meaning of the semicolon will become apparent later; for the time being, we just note that in this particular case we are allowed to drop the semicolon (the impatient reader may consult the fromCtxt and toCtxt rules in §3.4), yielding

$$y \neq 0 \vdash_\iota \iota z_{y\neq 0}(z \cdot y = x) \;=\; \iota z_{y\neq 0}(z \cdot y = x)$$

Is the latter sequent sound? Let us again evaluate it in a model where y is interpreted as 0. Crucial for our PITFOL calculus is the notion of a left-to-right **short circuit evaluation**, which we will now see in action: we *first* interpret all formulae on the left hand side of the sequent. When we find that at least one of them interprets as 'undefined', the whole sequent is considered unsound, since we agreed that in sound sequents, we never have to consider undefined formulae. When we find that no left hand side formulae interpret as 'undefined' and at least one of them interprets as 'false', the whole sequent is interpreted as 'true' (i.e., sound), without considering the formula on the right hand side of the sequent (its **consequent**)—hence the name 'short circuit' evaluation. We are in this situation here: the interpretation of $y \neq 0$ is clearly 'false', so the sequent is 'true' for this interpretation. For all other interpretations, where y is interpreted as a real number different from 0, the formula $y \neq 0$ is interpreted as 'true' and we have to consider the consequent, which clearly interprets as 'true', so in this case too, the whole sequent remains 'true'.

We believe this approach is "faithful to mathematical practice": one is allowed to prove theorems about potentially undefined terms such as $\frac{x}{y}$, but only in contexts where one has already shown that they cannot be undefined (here, that y must necessarily be nonzero).

To continue our illustration, let us consider the term

$$\iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y),$$

which can be interpreted as $\sqrt{y}$. Another application of the substitution rule to the sequent we obtained thus far, yields the difficult to decipher sequent

$$\begin{aligned} y \geq 0; \iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y) \neq 0 \\ \vdash_\iota \iota z_{y\geq 0 \& \iota w_{y\geq 0}(w\geq 0 \& w\cdot w = y)\neq 0}(z \cdot \iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y) = x) \\ = \iota z_{y\geq 0 \& \iota w_{y\geq 0}(w\geq 0 \& w\cdot w = y)\neq 0}(z \cdot \iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y) = x) \end{aligned}$$

If we consider $\sqrt{y}$ for a moment as a shorthand for $\iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y)$, the sequent becomes somewhat more readable:

$$y \geq 0; \sqrt{y} \neq 0 \vdash_\iota \iota z_{y\geq 0 \& \sqrt{y}\neq 0}(z \cdot \sqrt{y} = x) = \iota z_{y\geq 0 \& \sqrt{y}\neq 0}(z \cdot \sqrt{y} = x)$$

and if we consider in turn $\frac{x}{\sqrt{y}}$ as a shorthand for $\iota z_{y\geq 0 \& \sqrt{y}\neq 0}(z \cdot \sqrt{y} = x)$, we finally get

$$y \geq 0; \sqrt{y} \neq 0 \vdash_\iota \frac{x}{\sqrt{y}} = \frac{x}{\sqrt{y}}$$

If nothing else, from this example we learn that a facility for introducing shorthands (**defined symbol**s) is a necessity; we will illustrate it shortly

but for now consider notations like $\frac{x}{y}$, $\sqrt{y}$, $\frac{x}{\sqrt{y}}$, ... to be just abbreviations for the rather lengthy terms given above.
Moreover, this example also illustrates why we needed to extend the notion of sequent to the form

$$\sigma_1, \sigma_2, \ldots, \sigma_m; \gamma_1, \gamma_2, \ldots, \gamma_m \vdash_\iota \alpha$$

where $\gamma_1, \gamma_2, \ldots, \gamma_m$ is still called the **antecedent** and we will call $\sigma_1, \sigma_2, \ldots, \sigma_m$ the **context** of the sequent. (If the context is empty, we drop the semicolon and just write $\gamma_1, \gamma_2, \ldots, \gamma_m \vdash_\iota \alpha$.)
Indeed, the sequent

$$y \geq 0, \sqrt{y} \neq 0 \vdash_\iota \frac{x}{\sqrt{y}} = \frac{x}{\sqrt{y}}$$

is not a sound sequent. Consider an interpretation where y is interpreted as -1. Interpreting the sequents of the antecedent yields "false" resp. "undefined". If we interpret $y \geq 0$ first, we would be tempted to conclude that the interpretation of the whole sequent would be 'true', but in the antecedent, the order of the formulae is unimportant and we could just as well have started with the interpretation of $\sqrt{y} \neq 0$. Since it is possible to encounter an undefined formula when interpreting the sequent, it cannot be sound.

Hence, it is necessary to consider the version of the sequent with the semicolon. In contrast to the antecedent, the order of formulae in the context *is* significant; we are required to first interpret σ_1; if it is false then the whole sequent is automatically true (without considering the rest of the sequent, hence again short circuit evaluation). If it is undefined, the sequent is invalid. If it is true, then we have to consider σ_2 in a similar fashion. Only when all σ_i interpret as true, we have to consider the interpretation of the formulae in the antecedent.

Let us now illustrate the introduction of defined symbols, the need for which has been amply demonstrated. It turns out (see chapter 5) that we can indeed add the following sequents to the calculus:

$$y \neq 0 \vdash_\iota \mathrm{div}(x, y) = \iota z_{y \neq 0}(z \cdot y = x)$$
$$y \geq 0 \vdash_\iota \mathrm{sqrt}(y) = \iota w_{y \geq 0}(w \geq 0 \,\&\, w \cdot w = y)$$

Note that there are no domain formulae present in the defined symbols (i.e., we do not have to write something like "$\mathrm{div}_{y \neq 0}(x, y)$" or "$\mathrm{sqrt}_{y \geq 0}(y)$"); we will establish in chapter 5 that from the sequents above, one can extract all that is needed to keep the calculus consistent.

Finally, we remark that pathological theorems we found in other systems such as $\vdash \frac{1}{0} \neq \frac{1}{0}$ simply are not sound (and hence not deducible) in our calculus.

The price we have to pay for the simple semantics ("evaluate from left to right and you will never encounter an undefined term") and the absence of pathological theorems is a certain amount of verbosity in the logic. For example, suppose we are working in a suitable developed theory of real numbers in which the function $\lim_{n\to\infty} x_n$ is only defined for convergent sequences x_n. Then the sequent

$$\lim_{n\to\infty} x_n = a, \lim_{n\to\infty} y_n = b \vdash \lim_{n\to\infty} x_n + y_n = a + b$$

is not a valid sequent in our calculus, but

$$\text{convergent}(x_n), \text{convergent}(y_n); \lim_{n\to\infty} x_n = a, \lim_{n\to\infty} y_n = b \vdash \lim_{n\to\infty} x_n + y_n = a + b$$

is. It remains to be seen whether this "considerable clutter with definabality assumptions" [Harrison 1996] can be controlled enough to outweigh the advantages.

In practice, it seems this clutter is bearable enough; for example the corresponding theorem in the Mizar library reads

```
theorem :: SEQ_2:20
  seq is convergent & seq' is convergent implies
             lim (seq + seq')=(lim seq)+(lim seq');
```

Even more 'clutter' is to be seen in the corresponding theorem for real functions:

```
theorem :: LIMFUNC3:37
f1 is_convergent_in x0 & f2 is_convergent_in x0 &
(for r1,r2 st r1<x0 & x0<r2 ex g1,g2 st
  r1<g1 & g1<x0 & g1 in dom(f1+f2) & g2<r2 & x0<g2 & g2 in dom(f1+f2))
implies
f1+f2 is_convergent_in x0 & lim(f1+f2,x0)=lim(f1,x0)+lim(f2,x0);
```

So probably in practice, one can live with the extra clutter in theorems—the Mizar library is one of the largest library of formally proven mathematics in the world.

Chapter 2

First order predicate calculus with identity

As a starting point for our logic, we use the well known first order predicate calculus with identity, as formalised in [Hermes 1973]. We will refer to this formalisation as the **Hermes calculus**.

2.1 Syntax

The Hermes calculus uses a first order language consisting of

- **variable symbol**s, consisting of a number of letters: $x, y, z, xyzzy, \ldots$ We suppose that we have countably many variable symbols to our disposal.
- **function symbol**s, consisting of a number of letters. We suppose that we have countably many function symbols and that we can distinguish them somehow from the variable symbols. Each function symbol has an **arity** which is a natural number. Function symbols with arity zero are also called **constant symbol**s.
- **predicate symbol**s, with analogous conditions. Each predicate symbol has an **arity** which is a natural number.
- the connectives '$\neg$' and '&'
- the quantifier '$\forall$'
- the equality symbol '='

- punctuation symbols: the brackets '(' and ')' and the comma symbol ','. For clarity, we will sometimes omit some brackets.

From these symbols, **term**s are constructed as follows:

- A variable symbol is a term.
- If f is a n-ary function symbol and t_1, t_2, ..., t_n are terms, then $f(t_1, t_2, \ldots, t_n)$ is a term.

Next, **formula**e are built up following these rules:

- If t_1 and t_2 are two terms, then $t_1 = t_2$ is a formula. We call this kind of formula **atomic**.
- p is a n-ary predicate symbol and t_1, t_2, ..., t_n are terms, then $p(t_1, t_2, \ldots, t_n)$ is a formula. We call this kind of formula **atomic** too. For syntactical purposes, we will often treat $=$ as if it were a 2-ary predicate symbol. Whenever we proceed as such, we will mention that 'the case $t_1 = t_2$ is treated analogously'.
- If α is a formula, then $\neg\alpha$ is a formula too.
- If α and β are formulae, then so is $(\alpha \& \beta)$. We will not always explicitly write down the enclosing brackets around this kind of formula.
- If α is a formula and x a variable symbol, then $\forall x(\alpha)$ is a formula.

We will use the metalogical symbols $\equiv$ and $\not\equiv$ to express that two sequences of symbols are equal or not equal. We will often use variable names as x where we actually mean names for a variable name (just like we did when we said '$\forall x(\alpha)$ is a formula', where x can be any variable symbol).

With each formula and term, we associate a finite set of variable symbols, called the **free variable**s of that formula or term, as follows:

- A term x has x as its only free variable.
- A term $f(t_1, \ldots, t_n)$ has as free variables the union of the free variables of t_1, ..., t_{n-1} and t_n.
- An atomic formula $t_1 = t_2$ has as free variables the union of the free variables of t_1 and the free variables of t_2.
- An atomic formula $p(t_1, \ldots, t_n)$ has as free variables the union of the free variables of t_1, ..., t_{n-1} and t_n.

- A formula of the form $\neg\alpha$ has exactly the same free variables as α.
- A formula of the form $(\alpha \,\&\, \beta)$ has as free variables the union of the free variables of α and the free variables of β.
- The free variables of a formula of the form $\forall x(\alpha)$ are the same as those of α, except x. We say that x is a **bound variable** of $\forall x(\alpha)$.

Note that a variable can occur both free and bound in an expression: in

$$\forall x(x = y) \,\&\, z = x$$

the x in $x = y$ is bound but the x in $z = x$ is free.
Note also that nothing forbids a variable to be 'bound twice':

$$\forall x(x = y \,\&\, \forall x(z = x))$$

is a legitimate formula.

Given a term τ, a variable x and a term t, we call $[t/x]\tau$ the **substitution** of t for x in τ, which is the term obtained by replacing in τ all occurrences of x by t. More explicitly, we can define $[t/x]\tau$ like this:

- If τ is a variable symbol and $\tau \equiv x$, then $[t/x]\tau \equiv t$.
- If τ is a variable symbol and $\tau \not\equiv x$, then $[t/x]\tau \equiv \tau$.
- If $\tau \equiv f(\tau_1, \ldots, \tau_n)$, then $[t/x]\tau \equiv f([t/x]\tau_1, \ldots, [t/x]\tau_n)$. If c is a constant symbol, then $[t/x]c \equiv c$, in line with our definition of constant symbols as 0-ary function symbols.

Given a formula α, a variable x and a term t, we call $[t/x]\alpha$ the **substitution** of t for x in α, which is a formula obtained as follows by induction on the structure of α.

- If α is atomic and of the form $\alpha \equiv t_1 = t_2$, then $[t/x]\alpha \equiv [t/x]t_1 = [t/x]t_2$.
- If α is atomic and of the form $\alpha \equiv p(t_1, \ldots, t_n)$ then $[t/x]\alpha \equiv p([t/x]t_1, \ldots, [t/x]t_n)$.
- $[t/x]\neg\alpha \equiv \neg\,[t/x]\alpha$.
- $[t/x](\alpha \,\&\, \beta) \equiv [t/x]\alpha \,\&\, [t/x]\beta$.

- If x is not a free variable of $\forall y(\alpha)$, then $[t/x]\forall y(\alpha) \equiv \forall y(\alpha)$; in particular, $[t/x]\forall x(\alpha) \equiv \forall x(\alpha)$.
  If x is a free variable of $\forall y(\alpha)$ and y is not a free variable of t, then $[t/x]\forall y(\alpha) \equiv \forall y([t/x]\alpha)$.
  If x is a free variable of $\forall y(\alpha)$ and y is a free variable of t, then the substitution $[t/x]\forall y(\alpha)$ is **undefined** (in the sense that there is no such formula as $[t/x]\forall y(\alpha)$); we say that the substitution would **capture** the free variable y of t.

As we can see, substitution is not defined when the substitution would cause a free variable of t to become bound ('captured').

We remark that if y does not occur in α, the substitution $[y/x]\alpha$ is always defined.

A **sequent** is an expression of the form $\gamma_1, \gamma_2, \ldots, \gamma_n \vdash \alpha$. The list $\gamma_1, \ldots, \gamma_n$ is called the **antecedent** of the sequent; α is its **consequent** (in the literature also called **succedent**). We suppose that the antecedent does not contain the same formula more than once; if that would be the case, we silently delete its other instances. The order and multiplicity of the formulae in the antecedent is not important, i.e., we consider two sequents whose antecedents contain the same formulae identical even when they are in a different order or are repeated. For example, we consider

$$x = y, y = z \vdash x = z$$

and

$$y = z, y = z, x = y, y = z \vdash x = z$$

as two different notations for the same sequent.

In the sequel, we will use a few abbreviations:

- $\alpha \vee \beta$ is an abbreviation for $\neg(\neg\alpha \,\&\, \neg\beta)$
- $\alpha \Rightarrow \beta$ is an abbreviation for $\neg(\alpha \,\&\, \neg\beta)$
- $\alpha \Leftrightarrow \beta$ is an abbreviation for $(\alpha \Rightarrow \beta) \,\&\, (\beta \Rightarrow \alpha)$
- $\exists x(\alpha)$ is an abbreviation for $\neg\forall x(\neg\alpha)$

2.2 Semantics

We start from a **domain** ω, which must be a non-empty set. An **interpretation** is a function $\mathcal{I}$ defined on the variable, function and predicate symbols, such that

- for each variable symbol x we have $\mathcal{I}(x) \in \omega$,
- for each n-ary function symbol f, $\mathcal{I}(f)$ is an n-place function over ω,
- for each n-ary predicate symbol p, $\mathcal{I}(p)$ is an n-place predicate over ω.

Intuitively, an interpretation fixes a 'meaning' for each symbol of the formal language.

Remark that all functions and predicates above are total, i.e., defined for all n-tuples of elements of ω where n is the arity of the function or predicate.

An interpretation depends on the domain chosen, so actually we should write $\mathcal{I}_\omega$, but in the sequel we suppose the domain to be a fixed set and just write $\mathcal{I}$ to ease the notation.

Given an interpretation $\mathcal{I}$, a variable symbol x and an element $a \in \omega$, we define a new interpretation $\mathcal{I}^a_x$ as follows:

- $\mathcal{I}^a_x(x) := a$
- $\mathcal{I}^a_x(y) := \mathcal{I}(y)$ if $x \not\equiv y$.

We write $\mathcal{I}^{ab}_{xy}$ as an abbreviation of $(\mathcal{I}^a_x)^b_y$.

Starting from $\mathcal{I}$, we can associate an element of ω to each term t. As not to burden the notation, we will note this element too as $\mathcal{I}(t)$. It is defined inductively by setting

$$\mathcal{I}(f(t_1, \ldots, t_n)) := \mathcal{I}(f)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))$$

We say that a formula α is **valid** in an interpretation $\mathcal{I}$, or synonymously that $\mathcal{I}$ is a **model** of α, if

- If α is atomic, then we say that $t_1 = t_2$ is valid in $\mathcal{I}$ if and only if $\mathcal{I}(t_1) = \mathcal{I}(t_2)$, and that $p(t_1, \ldots, t_n)$ is valid in $\mathcal{I}$ if and only if $\mathcal{I}(p)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))$ holds.
- $\neg\alpha$ is valid in $\mathcal{I}$ if α is not valid in $\mathcal{I}$.
- $\alpha \,\&\, \beta$ is valid in $\mathcal{I}$ if both α and β are valid in $\mathcal{I}$.
- $\forall x(\alpha)$ is valid in $\mathcal{I}$ if for each $a \in \omega$, α is valid in $\mathcal{I}^a_x$.

We say that a list of formulae Γ is **valid** in an interpretation $\mathcal{I}$, or synonymously that $\mathcal{I}$ is a **model** of Γ, when all formulae of Γ are valid in $\mathcal{I}$ (i.e., $\mathcal{I}$ is a model of all formulae of Γ).

We say that a formula α is a **consequence** of a list of formulae Γ when for any $\mathcal{I}$ we have that $\mathcal{I}$ is a model of α whenever it is a model of Γ. We note this as $\Gamma \models \alpha$ and call the corresponding sequent $\Gamma \vdash \alpha$ a **sound sequent**.

2.3 Deduction rules

The Hermes calculus is a sequent calculus which uses the following deduction rules. The Greek letters α and β stand for formulae; Γ and Δ stand for lists of formulae; x stands for a variable and t for a term. **Premise**s are above the horizontal line and the **conclusion** below. As indicated above, double formulae are to be removed from the antecedent of the conclusion and the order of the formulae in the antecedent is not important.

We present abbreviated names of the rules in a box at the top of the rule, to be able to concisely refer to them in the sequel.

Assumption introduction: $\dfrac{}{\alpha \vdash \alpha}\ \boxed{\text{ass}}$

&-introduction: $\dfrac{\Gamma \vdash \alpha \quad \Delta \vdash \beta}{\Gamma, \Delta \vdash \alpha \,\&\, \beta}\ \boxed{\text{\&-intro}}$

&-elimination: $\dfrac{\Gamma \vdash \alpha \,\&\, \beta}{\Gamma \vdash \alpha}\ \boxed{\text{\&-elim}} \qquad \dfrac{\Gamma \vdash \alpha \,\&\, \beta}{\Gamma \vdash \beta}\ \boxed{\text{\&-elim}}$

Removal: $\dfrac{\Gamma, \alpha \vdash \beta \quad \Delta, \neg\alpha \vdash \beta}{\Gamma, \Delta \vdash \beta}\ \boxed{\text{rem}}$

Contradiction: $\dfrac{\Gamma \vdash \alpha \quad \Delta \vdash \neg\alpha}{\Gamma, \Delta \vdash \beta}\ \boxed{\text{contra}}$

$\forall$-introduction: $\dfrac{\Gamma \vdash \alpha}{\Gamma \vdash \forall x(\alpha)}$ provided x is not free in Γ

$\forall$-elimination: $\dfrac{\Gamma \vdash \forall x(\alpha)}{\Gamma \vdash \alpha}\ \boxed{\forall\text{-elim}}$

Substitution: $\dfrac{\Gamma \vdash \alpha}{[t/x]\Gamma \vdash [t/x]\alpha}\ \boxed{\text{subst}}$ where $[t/x]\Gamma$ means substituting t for x in

all formulae of Γ. This rule can only be applied if all the substitutions are defined.

Equality rules: $\dfrac{}{\vdash t = t}\,\boxed{\text{eq}}$ $\qquad$ $\dfrac{\Gamma \vdash \alpha}{\Gamma, x = t \vdash [{}^{t}\!/_{x}]\,\alpha}\,\boxed{\text{eqSubst}}$

The latter rule can only be applied if the substitution $[{}^{t}\!/_{x}]\,\alpha$ is defined.

We call a sequent **derivable** if it can be obtained by applying a deduction rule with derivable sequents as premises. From now on, we will use the symbol $\vdash$ only for derivable sequents; with the phrase "we have $\Gamma \vdash \alpha$" we mean that this sequent is a derivable one.

A **proof** is a finite list of sequents $s_1, s_2, \ldots, s_n$ where each s_i is obtained by applying one of the deduction rules with as antecedents sequents obtained earlier in the proof; i.e., if s_j is used as an antecedent, then $j < i$.

We call two formulae α and β **equivalent** when $\alpha \vdash \beta$ and $\beta \vdash \alpha$ are derivable. We note this as $\alpha \dashv\vdash \beta$. We call two formulae α and β **equivalent under the condition** γ when $\alpha, \gamma \vdash \beta$ and $\beta, \gamma \vdash \alpha$.

We call a formula α a **validity** if $\vdash \alpha$ is derivable.

One proves that the predicate calculus is **sound**: when $\Gamma \vdash \alpha$, then $\Gamma \models \alpha$. In other words, each derivable sequent is a sound sequent: we cannot derive unsound sequents using the deduction rules.

One proves also that the predicate calculus is **complete**: when $\Gamma \models \alpha$, then $\Gamma \vdash \alpha$. In other words, all sound sequents are derivable.

Chapter 3

Partially defined iota terms

3.1 Introduction

Whenever we prove that only one object x satisfying a certain property given by a formula φ exists, we want to be able to give it a name so we can reason about it. Hence we want to introduce a new kind of term in our logic of the form

$$\iota x(\varphi)$$

meaning 'the (unique) x for which φ holds'.

For example, in a theory about real numbers (i.e., ω is the set of reals), one can prove that there exists only one x such that

$$x \cdot x = 5 \,\&\, x > 0$$

(where we suppose that '5' and '0' are constants). This is of course the real number more customarily denoted with '$\sqrt{5}$'.

In that case, we want to introduce a new term

$$\iota x(x \cdot x = 5 \,\&\, x > 0)$$

read as 'the x such that $x \cdot x = 5 \,\&\, x > 0$', the interpretation of which will correspond to $\sqrt{5}$. This kind of terms has been introduced in the literature as **ι-terms** ([Hilbert & Bernays 1968]).

To deal with such terms, we will introduce a new calculus [Vernaeve & Hoogewijs 2007a] [Vernaeve & Hoogewijs 2007a]which we will coin the PITFOL calculus (for Partial Iota Terms in First Order Logic). It will be a superset of the first order predicate calculus with identity as presented earlier (which we will also call the 'Hermes calculus') in the sense that each term, formula, proof, ... of the Hermes calculus is a term, formula, proof, ... of the PITFOL calculus.

We will only allow terms of the form $\iota x(\varphi)$ to appear in a proof when we have a proof of

$$\vdash \exists x(\varphi) \,\&\, \forall x \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y)$$

where y is a variable not occurring in φ. We will call this the **uniqueness condition** for $\iota x(\varphi)$ and use the shorthand notation

$$\vdash \exists! x(\varphi)$$

These terms can also contain free variables:

$$\iota x(x \cdot x = z \,\&\, x > 0)$$

is a term whose interpretation will correspond to $\sqrt{\mathcal{I}(z)}$. However, this is a bad example because we cannot prove the uniqueness condition (there is no such x when $z < 0$).

We see from the last example that not everything we would like is a valid ι-term. To fix the definition of '$\sqrt{z}$', we could write

$$\iota x((z \geq 0 \Rightarrow (x \cdot x = z \,\&\, x > 0)) \,\&\, (z < 0 \Rightarrow x = 0))$$

which interprets as $\sqrt{\mathcal{I}(z)}$ whenever the square root is defined (i.e., when $\mathcal{I}(z) \geq 0$—we suppose for the sake of the example that $\mathcal{I}(\geq)$ is the predicate $\geq$ on the real numbers) and as 0 in the other cases.

However, such definitions are rather clumsy and give rise to strange side effects; for example, from $\sqrt{z} = 0$ we cannot conclude any more that $z = 0$ since z could have been any negative number too.

Another classic example is '$\frac{x}{y}$', defined as

$$\iota z((y \neq 0 \Rightarrow x = z \cdot y) \,\&\, (y = 0 \Rightarrow z = 0))$$

where we again encounter a side effect: $\vdash \frac{0}{0} = 0$ would be provable. This of course causes no inconsistencies, but it does not mirror the common mathematical practice, which forbids one from considering square roots of negative numbers or divisions by zero.

To meet this need, we introduce **partially defined iota terms**. These terms have the form

$$\iota x_{\psi}(\varphi)$$

and are to be interpreted in the same way as $\iota x(\varphi)$, but are only defined whenever ψ holds: if ψ is not valid then the interpretation of $\iota x_{\psi}(\varphi)$ is undefined; otherwise it is the unique x for which φ holds. We see that interpretations become three-valued: the interpretation of a term or formula

can be 'undefined'. If a term or formula is not undefined in an interpretation (i.e. it denotes an element of the domain resp. is valid or invalid) we call it **defined** in that interpretation.

A partially defined iota term $\iota x_{\psi}(\varphi)$ is only allowed to be used inside a proof whenever its uniqueness condition has been proved first:

$$\psi \vdash \exists x(\varphi) \,\&\, \forall x \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y))$$

where y is a variable not occurring in φ, or in shorthand:

$$\psi \vdash \exists! x(\varphi)$$

How will we handle undefined terms? Our solution is to make sure that when we are to interpret a term and find that it is undefined, the sequent in which it appears is automatically unsound. For example,

$$\Gamma \vdash \iota y_{x \neq 0}(x \cdot y = 1) \neq 0 \qquad (\text{meaning } \text{“}\Gamma \vdash \frac{1}{x} \neq 0\text{”})$$

is only sound when $\Gamma \vdash x \neq 0$. More specifically,

$$\vdash \iota y_{x \neq 0}(x \cdot y = 1) \neq 0 \qquad (\text{meaning } \text{“} \vdash \frac{1}{x} \neq 0\text{”})$$

is unsound since $\iota y_{x \neq 0}(x \cdot y = 1)$ is undefined when we interpret x as 0. However, we want

$$x \neq 0 \vdash \frac{1}{x} \neq 0 \quad , \quad x > 0 \vdash \frac{1}{x} > 0 \quad \text{and} \quad \vdash x > 1 \Rightarrow \frac{1}{x} < 1$$

(where $\frac{1}{x}$ is a fancy notation for the iota term $\iota y_{x \neq 0}(x \cdot y = 1)$—we will handle defined function symbols later) to be sound sequents. We see that in order to determine whether a term is defined or not, we need to take the 'context' into account. In the three last examples, from each of the 'contexts' '$x \neq 0$', '$x > 0$' and '$x > 1$', we can deduce that '$x \neq 0$', which is the condition for the definedness of $\frac{1}{x}$. The notion of 'context' entails that when interpreting a sequent, we *first* interpret the left hand sides of $\vdash$ and & (and hence, of $\Rightarrow$ too) and can make use of this information to deduce that terms in their right hand sides are defined.

It will become clear that the condition ψ itself must be defined in any interpretation, i.e., in a valid sequent, the term

$$\iota x_{\frac{1}{y}=5}(\ldots)$$

cannot occur because $\frac{1}{y} = 5$ is undefined in any interpretation that interprets y as 0; however,

$$\iota x_{y \neq 0 \,\&\, \frac{1}{y} = 5}(\dots)$$

will turn out to be a usable ι-term, because $y \neq 0 \,\&\, \frac{1}{y} = 5$ will turn out to be always defined (see below).

Note that even pathological terms like '$\iota x_{\frac{1}{y}=5}(\dots)$' are syntactically well-formed; we just cannot prove anything about them, because we cannot derive the uniqueness condition

$$\frac{1}{y} = 5 \vdash \exists! x(\dots)$$

Indeed, this sequent has to be unsound because in an interpretation where y is interpreted as 0, the interpretation of $\frac{1}{y} = 5$ is undefined.

Another consequence of the 'first interpret the left hand side of &' rule is that the conjunction is no longer commutative when one of its arguments is undefined.

To illustrate this, we look again at the formula $y \neq 0 \,\&\, \frac{1}{y} = 5$. In an interpretation where y is interpreted as 0, the left hand side of the conjunction is invalid and hence we conclude that the whole formula is invalid, without considering the interpretation of the right hand side.
If we consider another interpretation where y is not interpreted as 0, the left hand side of the conjunction is valid and we have to determine the interpretation of $\frac{1}{y} = 5$, which may be valid or invalid, but never undefined since $\frac{1}{y}$ is defined when y is not interpreted as 0.

On the other hand, the formula $\frac{1}{y} = 5 \,\&\, y \neq 0$ is undefined whenever the interpretation of y is 0. Indeed, we first have to determine the interpretation of $\frac{1}{y} = 5$ which is undefined since we have to interpret the term $\frac{1}{y}$.

3.2 Syntax

The syntax of the calculus with partially defined iota terms (the PITFOL calculus) is an extension of the syntax of the calculus without iota terms (the Hermes calculus).

The terms and formulae of the PITFOL calculus are constructed using the same rules as those of the Hermes calculus with the addition of

- If φ and ψ are formulae of the PITFOL calculus and x is a variable symbol then $\iota x_\psi(\varphi)$ is a term. This kind of terms will be called **ι-terms**. We call ψ the **domain formula** and φ the **definiens** of the ι-term.

Note that the terms of our new calculus can be divided into three categories:

- ι-less terms, not containing any ι symbol. These terms correspond to the terms of the Hermes calculus.
- Terms of the form $\iota x_\psi(\varphi)$, which we call ι-terms.
- Other terms of the PITFOL calculus, which are not ι-terms themselves but contain one or more ι-terms inside them, for example $f(\iota x_\psi(\varphi))$.

Remark that ι-terms are allowed to contain free variables (in the literature, sometimes such terms are called 'ι-expressions' and only get the name 'ι-term' when they do not contain free variables, e.g. [Hilbert & Bernays 1968]).

The **uniqueness condition** corresponding to a ι-term $\iota x_\psi(\varphi)$ is the sequent

$$\psi \vdash_\iota \exists! x(\varphi).$$

Note that even if the uniqueness condition $\psi \vdash_\iota \exists! x(\varphi)$ has not been derived, we still consider $\iota x_\psi(\varphi)$ to be a syntactically well-formed ι-term, although in the sequel it will turn out that they cannot appear in proofs.

The formulae of the PITFOL calculus are built up in the same way as those of the Hermes calculus, but of course their terms are terms of the PITFOL calculus. From now on, a 'formula' will stand for a formula of the PITFOL calculus unless we explicitly note otherwise.

Remark that in the PITFOL calculus, not only do formulae contain terms, but terms also can contain formulae (domain formula and definiens of ι-terms).

Given a formula α or term t, we denote the list containing all occurrences of its **top-level ι-term**s (i.e., those ι-terms that are themselves not contained into another ι-term) as $TLI(\alpha)$ resp. $TLI(t)$. Formally,

- $TLI(x) \equiv ()$, the empty list
- $TLI(\iota x_\psi(\varphi)) \equiv (\iota x_\psi(\varphi))$
- $TLI(f(t_1, t_2, \ldots, t_n)) \equiv TLI(t_1) : TLI(t_2) : \ldots : TLI(t_n)$
- $TLI(t_1 = t_2) \equiv TLI(t_1) : TLI(t_2)$
- $TLI(p(t_1, t_2, \ldots, t_n)) \equiv TLI(t_1) : TLI(t_2) : \ldots : TLI(t_n)$
- $TLI(\neg\alpha) \equiv TLI(\alpha)$
- $TLI(\alpha \,\&\, \beta) \equiv TLI(\alpha) : TLI(\beta)$

- $TLI(\forall x(\alpha)) \equiv TLI(\alpha)$

where ':' denotes the list concatenation operator and (e) the list with the single element e.

Given a formula α or term t, we denote the list containing the uniqueness conditions corresponding to all its **top-level ι-term**s as $UC(\alpha)$ resp. $UC(t)$. In the sequel, we will also refer to this list as "the uniqueness conditions for a term or formula". Formally,

- $UC(x) \equiv ()$, the empty list
- $UC(\iota x_{\psi}(\varphi)) \equiv (\psi \vdash_{\iota} \exists! x(\varphi))$
- $UC(f(t_1, t_2, \ldots, t_n)) \equiv UC(t_1) : UC(t_2) : \ldots : UC(t_n)$
- $UC(t_1 = t_2) \equiv UC(t_1) : UC(t_2)$
- $UC(p(t_1, t_2, \ldots, t_n)) \equiv UC(t_1) : UC(t_2) : \ldots : UC(t_n)$
- $UC(\neg\alpha) \equiv UC(\alpha)$
- $UC(\alpha \,\&\, \beta) \equiv UC(\alpha) : UC(\beta)$
- $UC(\forall x(\alpha)) \equiv UC(\alpha)$

We will denote $\iota x(\varphi)$ as an abbreviation for $\iota x_{\forall x(x=x)}(\varphi)$, i.e., a ι-term for which the definedness condition is always satisfied.

The notion of free variable of a term is extended with

- A term $\iota x_{\psi}(\varphi)$ has as free variables the union of the free variables of φ except x (which we call a bound variable of the term) and the free variables of ψ. Symbolically, if we denote the set of free variables of a term t as $FV(t)$, then

$$FV(\iota x_{\psi}(\varphi)) = (FV(\varphi) \setminus \{x\}) \cup FV(\psi).$$

  Note that any free occurrences of x in ψ are *not* bound by the ι.

A **pitfol sequent** is an expression of the form

$$\sigma_1, \sigma_2, \ldots; \gamma_1, \gamma_2, \ldots \vdash_{\iota} \alpha$$

where the σ's, γ's and α are formulae of the PITFOL calculus. If there are no σ's, then the leading semicolon is to be dropped. Note that we add a ι subscript to the $\vdash$ symbol to indicate the difference with sequents of the Hermes calculus.

The finite and possibly empty list $\sigma_1, \sigma_2, \ldots$ is called the **context** of the sequent; the finite and possibly empty list $\gamma_1, \gamma_2, \ldots$ is called the **antecedent** and the mandatory formula α is the **consequent**.
As before, the order and multiplicity of the formulae of the antecedent is not important. However, the order of the formulae in the context is significant. When the context contains double formulae, we consider the sequent identical to the same sequent in which we only keep the first copy of the double formula in the context. Note that a formula is allowed to occur both in the antecedent and the context of the conclusion; we are not allowed to remove those double formulae.

For example, we consider the sequent

$$\alpha, \alpha, \beta, \alpha, \gamma, \beta; \gamma, \gamma, \alpha \vdash_\iota \delta$$

identical to the sequent $\alpha, \beta, \gamma; \gamma, \alpha \vdash_\iota \delta$ and, since we noted that the order of the formulae in the antecedent is not important, the sequent $\alpha, \beta, \gamma; \alpha, \gamma \vdash_\iota \delta$. However, it is different from the sequent $\alpha, \beta, \gamma; \gamma \vdash_\iota \delta$.

If α is a formula, then informally, $\mathbf{\Delta}(\alpha)$ is defined as the formula that states when α is defined. However, $\mathbf{\Delta}(\alpha)$ may also be the symbol $\top$ in case α is always defined. The logic will be constructed in such a way that the symbol $\top$ will never occur inside sequents; it is just a syntactical device to aid in the construction of $\mathbf{\Delta}(\alpha)$. Semantically, $\top$ will behave as if its interpretation were 'true'.

For example, $\mathbf{\Delta}(\iota x_{y \neq 0}(x \cdot y = 1) > 0)$ will be $y \neq 0$. Note that the symbol $\mathbf{\Delta}$ is not part of our formal language; $\mathbf{\Delta}(\alpha)$ is a metalogical operator that maps the formula α to another formula or the symbol $\top$. We will again use the notation $\alpha \equiv \beta$ to express that two formulae are identical (where $\equiv$ is also a metalogical operator).
Likewise, if t is a term, $\mathbf{\Delta}(t)$ will be the formula (or the symbol $\top$) that states when t is defined, so $\mathbf{\Delta}(\iota x_{y \neq 0}(x \cdot y = 1)) \equiv y \neq 0$.

Formally, $\mathbf{\Delta}(t)$ is defined as follows:

- If t is a variable symbol, then $\mathbf{\Delta}(t)$ is $\top$.
- $\mathbf{\Delta}(f(t_1, t_2, \ldots, t_n)) \equiv \mathbf{\Delta}(t_1) \,\&\, (\mathbf{\Delta}(t_2) \,\&\, (\ldots \,\&\, (\mathbf{\Delta}(t_{n-1}) \,\&\, \mathbf{\Delta}(t_n))))$ and for 1-ary function symbols $\mathbf{\Delta}(f(t_1)) \equiv \mathbf{\Delta}(t_1)$. If c is a constant symbol, we define $\mathbf{\Delta}(c)$ also as $\top$, in line with our definition of constant symbols as 0-ary function symbols.
- $\mathbf{\Delta}(\iota x_\psi(\varphi)) \equiv \psi$.

One could also express these rules less formally as $\mathbf{\Delta}(t) \equiv \psi_1 \& \psi_2 \& \ldots \& \psi_n$ up to associativity, where $\psi_1, \psi_2, \ldots, \psi_n$ are the domain formulae of the top-level ι-terms of t in left-to-right order.

Note that the associativity of $\mathbf{\Delta}(t)$ depends on the structure of t. For example, $\mathbf{\Delta}(f(g(\iota x_{\psi_1}(\varphi_1), \iota x_{\psi_2}(\varphi_2)), \iota x_{\psi_3}(\varphi_3))) \equiv (\psi_1 \,\&\, \psi_2) \,\&\, \psi_3$.

> Note that from the sequel, it will appear that the conjunction is associative (property 34), and moreover, in this case also commutative (since from the uniqueness conditions $\psi_i \vdash_\iota \exists! x(\varphi_i)$ we can conclude $\vdash_\iota \mathbf{\Delta}(\psi_i)$; see property 35). So there's no real reason to insist on this specific form of $\mathbf{\Delta}(t)$.

As indicated above, we don't want the symbol $\top$ to occur in sequents. To achieve this, we will assume that formulae containing $\top$ are automatically simplified with the rules

$$\begin{array}{rcl} \alpha \,\&\, \top & \rightarrow & \alpha \\ \top \,\&\, \alpha & \rightarrow & \alpha \\ \forall x(\top) & \rightarrow & \top \end{array}$$

We see that, using these rules, either the symbol $\top$ does not occur in $\mathbf{\Delta}(t)$ or $\mathbf{\Delta}(t) \equiv \top$.

For formulae, the formal definition of $\mathbf{\Delta}(\alpha)$ is

- $\mathbf{\Delta}(t_1 = t_2) \equiv \mathbf{\Delta}(t_1) \,\&\, \mathbf{\Delta}(t_2)$
- $\mathbf{\Delta}(p(t_1, t_2, \ldots)) \equiv \mathbf{\Delta}(t_1) \,\&\, (\mathbf{\Delta}(t_2) \,\&\, (\mathbf{\Delta}(t_3) \,\&\, \ldots))$; for 0-ary predicate symbols we have again $\mathbf{\Delta}(p(t_1, t_2, \ldots)) \equiv \top$
- $\mathbf{\Delta}(\neg\alpha) \equiv \mathbf{\Delta}(\alpha)$.
- If $\mathbf{\Delta}(\beta) \not\equiv \top$, then we define $\mathbf{\Delta}(\alpha \,\&\, \beta) \equiv \mathbf{\Delta}(\alpha) \,\&\, (\alpha \Rightarrow \mathbf{\Delta}(\beta))$, which is an abbreviation for $\mathbf{\Delta}(\alpha) \,\&\, \neg(\alpha \,\&\, \neg(\mathbf{\Delta}(\beta)))$. Note that we have given no simplification rule for $\neg\top$, so we have to define explicitly $\mathbf{\Delta}(\alpha \,\&\, \beta) \equiv \mathbf{\Delta}(\alpha)$ in case $\mathbf{\Delta}(\beta) \equiv \top$.
- $\mathbf{\Delta}(\forall x(\alpha)) \equiv \forall x(\mathbf{\Delta}(\alpha))$.

For atomic formulae α, we see again that up to associativity, $\mathbf{\Delta}(\alpha) \equiv \psi_1 \,\&\, \psi_2 \,\&\, \ldots \,\&\, \psi_n$ where $\psi_1, \psi_2, \ldots, \psi_n$ are the domain formulae of the top-level ι-terms of α.

Finally, we define $\mathbf{\Delta}(\top) \equiv \top$.

As an example, let us reconsider the term

$$t \equiv \iota z_{y \geq 0 \,\&\, \iota w_{y \geq 0}(w \geq 0 \,\&\, w \cdot w = y) \neq 0}(z \cdot \iota w_{y \geq 0}(w \geq 0 \,\&\, w \cdot w = y) = x).$$

We already met this term in the introduction and noticed that one could interpret it as $\frac{x}{\sqrt{y}}$.

Then we have $\mathbf{\Delta}(t) \equiv y \geq 0 \,\&\, \iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y) \neq 0$, and

$$\begin{aligned}\mathbf{\Delta}(\mathbf{\Delta}(t)) &\equiv \mathbf{\Delta}(y \geq 0) \,\&\, y \geq 0 \Rightarrow \mathbf{\Delta}(\iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y) \neq 0) \\ &\equiv \top \,\&\, y \geq 0 \Rightarrow y \geq 0 \\ &\equiv y \geq 0 \Rightarrow y \geq 0.\end{aligned}$$

For another example, reconsider the formula $\neg(y = 0) \,\&\, \iota x_{\neg(y=0)}(y \cdot x = 1) = 5$, which we abbreviated as $y \neq 0 \,\&\, \frac{1}{y} = 5$. We then have

$$\begin{aligned}\mathbf{\Delta}\left(\neg(y = 0) \,\&\, \frac{1}{y} = 5\right) &\equiv \mathbf{\Delta}(\neg(y = 0)) \,\&\, \left((\neg(y = 0)) \Rightarrow \mathbf{\Delta}\left(\frac{1}{y} = 5\right)\right) \\ &\equiv \top \,\&\, ((\neg(y = 0)) \Rightarrow \neg(y = 0)) \\ &\equiv (\neg(y = 0)) \Rightarrow \neg(y = 0).\end{aligned}$$

Substitution is defined as in the Hermes calculus, extended with the case $[t/x]\,\iota y_\psi(\varphi)$ as follows:

- If $x \equiv y$ or if x is not a free variable of φ, then $[t/x]\,\iota y_\psi(\varphi) \equiv \iota y_\psi(\varphi)$ when x is not a free variable of ψ, or $[t/x]\,\iota y_\psi(\varphi) \equiv \iota y_{\mathbf{\Delta}(t)\&[t/x]\psi}(\varphi)$ when x is a free variable of ψ.
- If $x \not\equiv y$ and x is a free variable of φ and y is not a free variable of t, then $[t/x]\,\iota y_\psi(\varphi) \equiv \iota y_{\mathbf{\Delta}(t)\&[t/x]\psi}([t/x]\,\varphi)$.
- If $x \not\equiv y$ and x is a free variable of φ and y is a free variable of t, then $[t/x]\,\iota y_\psi(\varphi)$ is undefined; we say that the substitution would **capture** the free variable y of t.

We see that for the definiens, the substitution behaves similarly to the case $[t/x]\forall y(\alpha)$. Note that the 'naive' definition $[t/x]\,\iota y_\psi(\varphi) \equiv \iota y_{[t/x]\psi}([t/x]\,\varphi)$ in the cases where we defined it as $\iota y_{\mathbf{\Delta}(t)\&[t/x]\psi}(\varphi)$ would get us into trouble later on (in particular, lemma 19 would not hold).

Remark that the substitution $[t/x]\,\iota y_\psi(\varphi)$ in some cases uses $[t/x]\,\varphi$ and/or $[t/x]\,\psi$; if in those cases one or both substitutions would be undefined, the substitution $[t/x]\,\iota y_\psi(\varphi)$ is also undefined.

The **complexity** of a formula α or term t, abbreviated $\mathrm{cpl}(\alpha)$ or $\mathrm{cpl}(t)$, is defined as the number of $\&$, $\neg$, $\forall$, ι, $=$, predicate and function symbols in α or t. Note that we define complexity also for terms since they can contain formulae too; for example, $\mathrm{cpl}(\iota x_\psi(\varphi)) = 1 + \mathrm{cpl}(\psi) + \mathrm{cpl}(\varphi)$.

Terms of complexity 0 are variable symbols; there are no formulae of complexity 0.
Terms of complexity 1 are of the form $f(x_1, x_2, \ldots, x_n)$; formulae of complexity 1 are of the form $p(x_1, x_2, \ldots, x_n)$ or $x_1 = x_2$.
Terms of complexity 2 are of the form $f_1(x_1, \ldots, x_{m-2}, f_2(y_1, \ldots, y_k), x_m, x_{m+1}, \ldots, x_n)$; formulae of complexity 2 are of the form $p(x_1, \ldots, x_{m-2}, f(y_1, \ldots, y_k), x_m, \ldots, x_n)$, $\neg p(x_1, \ldots)$ or $\forall x(p(x_1, \ldots, x_n))$ where $p(x_1, \ldots, x_n)$ may be replaced with $x_1 = x_2$.
We remark that ι-terms only start occurring in terms from complexity 3 onwards (these are of the form $\iota x_{p_1(x_1,\ldots,x_n)}(p_2(y_1, \ldots, y_m))$ where one or both of the $p_i(\ldots)$ may be replaced by $x_1 = x_2$ or $y_1 = y_2$); and in formulae starting at complexity 4.

3.2.1 Properties of substitution

We extended the notion of substitution to the case $[t/x]\iota y_\psi(\varphi)$; we will now investigate how this affects the properties of substitution.

In this section, α will stand for a formula of the PITFOL calculus, t, u, t_1, t_2 and τ for terms of the PITFOL calculus and w, x, y and z for variable symbols.

Property 1 *If x is not a free variable of α or τ then x is also not a free variable of $\mathbf{\Delta}(\alpha)$ resp. $\mathbf{\Delta}(\tau)$.*

This is easy to prove using structural induction on α and τ.

Another way of stating this property is $FV(\mathbf{\Delta}(\alpha)) \subseteq FV(\alpha)$ and likewise for τ.

Property 2 *The substitutions $[x/x]\alpha$ and $[x/x]\tau$ are always defined; $[x/x]\alpha \equiv \alpha$ and $[x/x]\tau \equiv \tau$.*

Property 3 *If the substitution $[x/y]\alpha$ is defined and $\mathbf{\Delta}(\alpha) \not\equiv \top$ then $\mathbf{\Delta}([x/y]\alpha) \equiv [x/y]\mathbf{\Delta}(\alpha)$. If the substitution $[x/y]\tau$ is defined and $\mathbf{\Delta}(\alpha) \not\equiv \top$ then $\mathbf{\Delta}([x/y]\tau) \equiv [x/y]\mathbf{\Delta}(\tau)$.*

For convenience, we define $[t/x]\top \equiv \top$, so the previous property can be stated more succinctly as "If the substitution $[x/y]\alpha$ is defined then $\mathbf{\Delta}([x/y]\alpha) \equiv [x/y]\mathbf{\Delta}(\alpha)$".

Property 4 *If the substitution $[t/x]\alpha$ is defined and x is a free variable of α, then $FV([t/x]\alpha) = FV(t) \cup (FV(\alpha) \setminus \{x\})$.*

If the substitution $[t/x]\tau$ is defined and x is a free variable of τ, then $FV([t/x]\tau) = FV(t) \cup (FV(\tau) \setminus \{x\})$.

From this property, we also get

- If x is not a free variable of t, then x is also not a free variable of $[t/x]\alpha$ resp. $[t/x]\tau$ if the substitution is defined.
- If y is not a free variable of t and α resp. τ, then y is also not a free variable of $[t/x]\alpha$ resp. $[t/x]\tau$ if the substitution is defined.

Property 5 *If x is not a free variable of α or τ, then $[t/x]\alpha \equiv \alpha$ resp. $[t/x]\tau \equiv \tau$ and this substitution is always defined.*

One proves properties 4 and 5 simultaneously by structural induction on α and τ.

Corollary 6 *If $x \not\equiv y$, then $[t/x][y/x]\alpha \equiv [y/x]\alpha$ and $[t/x][y/x]\tau \equiv [y/x]\tau$; all these substitutions are always defined.*

Proof.

First consider the case that x is not a free variable of α. Using the previous lemma, what we have to show simplifies to $\alpha \equiv \alpha$.

If x is free in α, then we apply property 4, yielding $FV([y/x]\alpha) = \{y\} \cup (FV(\alpha) \setminus \{x\})$. Hence $x \notin FV([y/x]\alpha)$. Applying property 5 concludes the proof. The case for $[t/x][y/x]\tau$ is analogous. □

Property 7 *If y is not a free variable of t, $x \not\equiv y$ and $x \not\equiv z$, then $[t/x][z/y]\alpha \equiv [z/y][t/x]\alpha$ if at least one of these substitutions is defined. Likewise for τ instead of α.*

Proof.

The proof is by structural induction on α and τ; the only interesting case is $\tau \equiv \iota w_\psi(\varphi)$.

- $y \equiv w$ and y is not a free variable of ψ. Then we have to prove that $[t/x]\tau \equiv [z/y][t/x]\tau$. Remark that in this case, y is not a free variable of τ; by property 4, y is also not a free variable of $[t/x]\tau$. Applying property 5 concludes this case.
- $y \equiv w$ and y is a free variable of ψ. Then we have to prove that $[t/x]\iota y_{[z/y]\psi}(\varphi) \equiv [z/y][t/x]\iota w_\psi(\varphi)$.
  - x is not a free variable of φ and ψ. Hence x is not a free variable of $[z/y]\psi$ and what remains to prove is $\iota y_{[z/y]\psi}(\varphi) \equiv [z/y]\iota y_\psi(\varphi)$, which is trivial.

  - x is not a free variable of φ and x is a free variable of ψ. We have to prove that $\iota y_{\boldsymbol{\Delta}(t)\&[t/x][z/y]\psi}(\varphi) \equiv [z/y]\,\iota y_{\boldsymbol{\Delta}(t)\&[t/x]\psi}(\varphi)$. The right hand side is identical to $\iota y_{\boldsymbol{\Delta}(t)\&[z/y][t/x]\psi}(\varphi)$. Induction on ψ concludes this case.
  - x is a free variable of φ. This case is the same as the previous one, except that the definiens is $[t/x]\,\varphi$ instead of φ.

- $y \not\equiv w$ and y is not a free variable of φ.
  - If y is not a free variable of ψ, then we have to prove that $[t/x]\,\iota w_{\psi}(\varphi) \equiv [z/y]\,[t/x]\,\iota w_{\psi}(\varphi)$. Note that y is not a free variable of $\iota w_{\psi}(\varphi)$ and hence also not a free variable of $[t/x]\,\iota w_{\psi}(\varphi)$, from which we this case easily follows.
  - If y is a free variable of ψ, then we have to prove that $[t/x]\,\iota w_{[z/y]\psi}(\varphi) \equiv [z/y]\,[t/x]\,\iota w_{\psi}(\varphi)$.
    - $x \equiv w$, or $x \not\equiv w$ and x is not a free variable of φ. Then we again have to show that $\iota w_{[z/y]\psi}(\varphi) \equiv [z/y]\,\iota w_{\psi}(\varphi)$ when x if not a free variable of ψ, or that $\iota w_{\boldsymbol{\Delta}(t)\&[t/x][z/y]\psi}(\varphi) \equiv [z/y]\,\iota y_{\boldsymbol{\Delta}(t)\&[t/x]\psi}(\varphi)$ when x is a free variable of ψ.
    - $x \not\equiv w$ and x is a free variable of φ. For this case, we have to prove that $\iota w_{\boldsymbol{\Delta}(t)\&[t/x][z/y]\psi}([t/x]\,\varphi) \equiv [z/y]\,\iota w_{\boldsymbol{\Delta}(t)\&[t/x]\psi}([t/x]\,\varphi)$; the proof is analogous.

- $y \not\equiv w$ and y is a free variable of φ. We have to prove that $[t/x]\,\iota w_{[z/y]\psi}([z/y]\,\varphi) \equiv [z/y]\,[t/x]\,\iota w_{\psi}([z/y]\,\varphi)$. The proof is analogous to the preceding cases.

□

Note that in general, $[t/x]\,[u/y]\,\tau \not\equiv [u/y]\,[t/x]\,\tau$, when $x \not\equiv y$, x is not a free variable of u and y is not a free variable of t, as was the case in the Hermes calculus. Indeed, a counterexample is given by

$$\begin{array}{rcl} [\iota x_{z=w}(x=x)/x]\,[\iota x_{z\neq w}(x=x)/y]\,\iota z_{x=y}(z=w) & \equiv & \iota z_{z=w\&\ldots}(z=w) \\ [\iota x_{z\neq w}(x=x)/y]\,[\iota x_{z=w}(x=x)/x]\,\iota z_{x=y}(z=w) & \equiv & \iota z_{z\neq w\&\ldots}(z=w) \end{array}$$

where w, x, y and z are mutually different variable symbols.

3.3 Semantics

In this section, we adapt the semantics from §2.2 to the PITFOL calculus.

Given an interpretation $\mathcal{I}$, we can associate to each term t of the PITFOL calculus an element of the domain ω or the status '**undefined**' (in the latter case, we write $\mathcal{I}(t) = \perp$), as follows:

- If ψ is valid in $\mathcal{I}$ and there is a unique $a \in \omega$ such that φ is valid in $\mathcal{I}^a_x$, then $\mathcal{I}(\iota x_\psi(\varphi))$ is the $a \in \omega$ such that φ is valid in $\mathcal{I}^a_x$.
  Else, $\mathcal{I}(\iota x_\psi(\varphi))$ is undefined.

- $\mathcal{I}(f(t_1, \ldots, t_n)) := \mathcal{I}(f)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))$, provided all $\mathcal{I}(t_i)$ are defined; else $\mathcal{I}(f(t_1, \ldots, t_n))$ is undefined.

Note that again, for function symbols f and predicate symbols p, $\mathcal{I}(f)$ and $\mathcal{I}(p)$ are supposed to be total functions resp. predicates. Also, the interpretation of a variable symbol is always an element of ω; it can never be undefined. Support for partially defined functions and predicates will be given in §5. Hence, for now, the only "source of undefinedness" are the ι-terms.

We attach to a formula α and an interpretation $\mathcal{I}$ the status valid, invalid or **undefined** as follows:

- If α is atomic, then we say that $t_1 = t_2$ is valid in $\mathcal{I}$ if both $\mathcal{I}(t_i)$ are defined and $\mathcal{I}(t_1) = \mathcal{I}(t_2)$. We say that $p(t_1, \ldots, t_n)$ is valid in $\mathcal{I}$ if all $\mathcal{I}(t_i)$ are defined and $\mathcal{I}(p)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))$ holds. However, if not all $\mathcal{I}(t_i)$ are defined, we say that α is undefined. In all other cases, α is invalid.

- $\neg\alpha$ is valid in $\mathcal{I}$ if α is invalid in $\mathcal{I}$; $\neg\alpha$ is invalid in $\mathcal{I}$ if α is valid in $\mathcal{I}$; $\neg\alpha$ is undefined if α is undefined.

- $\alpha \,\&\, \beta$ is valid in $\mathcal{I}$ if both α and β are valid in $\mathcal{I}$; $\alpha \,\&\, \beta$ is invalid when α is invalid, or when α is valid and β is invalid; else $\alpha \,\&\, \beta$ is undefined. (This is the **McCarthy conjunction** [McCarthy 1967].)

- $\forall x(\alpha)$ is valid in $\mathcal{I}$ if for each $a \in \omega$, α is valid in $\mathcal{I}^a_x$. If there exist one or more $a \in \omega$ such that α is undefined in $\mathcal{I}^a_x$, then we say that $\forall x(\alpha)$ is undefined. In all other cases, α is invalid.

When a formula is valid or invalid, we will call it **defined**.

Note that we use the term 'undefined' in two contexts: a substitution can be undefined in the sense that the *substitution* $[t/x]\alpha$ cannot take place, hence this is a syntactical notion; on the other hand, a term or a formula of the PITFOL calculus can be undefined in an *interpretation* $\mathcal{I}$, so this is a

semantical notion. It will be clear from the context which kind of 'undefined' we mean in the sequel.

This yields the following truth tables for $\neg\alpha$ and $\alpha \,\&\, \beta$:

α	$\neg\alpha$
T	F
F	T
U	U

$\alpha\backslash\beta$	T	F	U
T	T	F	U
F	F	F	F
U	U	U	U

where T, F and U stand for valid (true), invalid (false) and undefined respectively.

We can see that $\alpha \,\&\, \beta$ is defined if and only if α is defined and when α is true, β is defined. In other words, to prove that $\alpha \,\&\, \beta$ is defined, we have to prove that both

- α is defined
- β is defined, where we may assume that α is true ("β is defined in the context of α").

Yet another way to express this is that semantically, the & connective is evaluated from left to right with "**short circuit evaluation**": to evaluate $\alpha \,\&\, \beta$, we first check whether α is false. If this is the case, then we "short circuit" the evaluation here and declare $\alpha \,\&\, \beta$ to be false. If α is undefined, then we can short circuit again and declare $\alpha \,\&\, \beta$ to be undefined. Otherwise, we continue our left-to-right evaluation and consider β, knowing α to be true at this stage; the truth value of $\alpha \,\&\, \beta$ is then the truth value of β.

Using the definitions of $\vee$, $\Rightarrow$ and $\Leftrightarrow$ yields their truth tables:

$\alpha\backslash\beta$	T	F	U
T	T	T	T
F	T	F	U
U	U	U	U

$\alpha\backslash\beta$	T	F	U
T	T	F	U
F	T	T	T
U	U	U	U

$\alpha\backslash\beta$	T	F	U
T	T	F	U
F	F	T	U
U	U	U	U

Examining these tables, we conclude again that semantically, the evaluation is done left to right with short circuit. More specifically, we note that to prove that $\alpha \vee \beta$ is defined, we have to prove that α is defined, and that β is defined provided $\neg\alpha$ holds ("β is defined in the context of $\neg\alpha$"). Also, $\alpha \Rightarrow \beta$ is defined if and only if $\alpha \,\&\, \beta$ is defined, and $\alpha \Leftrightarrow \beta$ is defined when α and β are both defined.

We say that the **uniqueness condition** of a ι-term $\iota x_\psi(\varphi)$ holds when for any interpretation $\mathcal{I}$, we have:

- ψ is defined in $\mathcal{I}$.
- If ψ is valid in $\mathcal{I}$, then φ is valid in $\mathcal{I}^a_x$ for exactly one $a \in \omega$ and invalid in $\mathcal{I}^a_x$ for all other $a \in \omega$.

This will of course correspond to $\psi \models \exists! x(\varphi)$ once we have defined $\models$ in the PITFOL calculus.

We say that a formula α is a **consequence** of a list of formulae Γ if for any interpretation $\mathcal{I}$, all of the following conditions hold:

- the uniqueness conditions of all ι-terms in Γ and α hold.
- whenever all formulae of Γ are valid in $\mathcal{I}$, then also α is valid in $\mathcal{I}$.
- all formulae of Γ are **defined** in $\mathcal{I}$ (i.e., valid in $\mathcal{I}$ or invalid in $\mathcal{I}$).

We note this as $\Gamma \models_\iota \alpha$ and call the sequent $\Gamma \vdash_\iota \alpha$ a **sound sequent**.

The last condition 'forbids' us from considering undefined statements, just like in common mathematical practice, where one is not allowed to speak of e.g. $\frac{1}{0}$. For example, without the last condition, we would have that

$$\iota y_{x\neq 0}(x \cdot y = 1) \neq 0 \models_\iota x \neq 0 \qquad \text{(intuitively } \text{“}\frac{1}{x} \neq 0 \models_\iota x \neq 0\text{”)}$$

but because of the second condition, we are not allowed to talk about '$\frac{1}{x}$' if we cannot prove that it is always defined. On the other hand, we do have

$$x \neq 0 \models_\iota \iota y_{x\neq 0}(x \cdot y = 1) \neq 0 \qquad (\text{“}x \neq 0 \models_\iota \frac{1}{x} \neq 0\text{”})$$

because we do not require the consequent to be defined for all interpretations.

To simplify the treatment of contexts, a PITFOL sequent may also contain a **context** Σ. This makes the calculus easier to use; it is possible to develop a calculus for the PITFOL calculus that does not use contexts in its sequents, but that calculus turns out to be more cumbersome to use in practice (see §3.5.4).

Let $\Sigma \equiv \sigma_1, \sigma_2, \ldots$. We say that α is a **consequence** of $\Sigma; \Gamma$ if for any interpretation $\mathcal{I}$, the following holds:

- the uniqueness conditions of all ι-terms in Σ, Γ and α hold.
- whenever all formulae of Γ and Σ are valid in $\mathcal{I}$, then also α is valid in $\mathcal{I}$.
- whenever all formulae of Σ are valid in $\mathcal{I}$, then all formulae of Γ are defined in $\mathcal{I}$.

- σ_1 is defined in $\mathcal{I}$; when σ_1 is valid in $\mathcal{I}$, σ_2 is defined in $\mathcal{I}$; when σ_1 and σ_2 are valid in $\mathcal{I}$, σ_3 is defined in $\mathcal{I}$, and so on. This is equivalent to requiring that the formula $\sigma_1 \,\&\, \sigma_2 \,\&\, \ldots$ be defined in $\mathcal{I}$.

We note this as $\Sigma; \Gamma \models_\iota \alpha$.

3.3.1 Examples

The term $\iota x(x = y)$ has the same interpretation as the term y; indeed, the only x fulfilling the condition $x = y$ is y.

The interpretation of the term $\iota x_{\neg\forall x(x=x)}(x = y)$ is always 'undefined'. Likewise, $y = \iota x_{\neg\forall x(x=x)}(x = y)$ is a formula whose interpretation is always 'undefined'. Actually, any term of the form $\iota x_{\neg\forall x(x=x)}(\varphi)$ is always interpreted as 'undefined'.

One could wonder what a term of the form $\iota x_\psi(\varphi)$ means when x is a free variable of ψ, as in the term $\iota x_{x\neq z}(x = y)$. Following the definition of interpretation given above, the interpretation of this term is again y whenever the interpretation of x differs from that of z; else, it is undefined. In other words, this term has the same interpretation as $\iota w_{x\neq z}(w = y)$; we see that we can perform a change of variable name, and this change leaves the domain formula unchanged. We will formally derive this in corollary 43.

From this example, we observe that in a ι-term of the form $\iota x_\psi(\varphi)$, x binds in φ but not in ψ, which is reflected in the definition of $FV(\iota x_\psi(\varphi))$.

Note that in contrast to the short circuit evaluation of the conjunction, $\forall x(\alpha)$ is undefined as soon as $\mathcal{I}^a_x(\alpha)$ is undefined for one $a \in \omega$, even though there might be $b \in \omega$ for which $\mathcal{I}^a_x(\alpha)$ is invalid, which seems to run counter to the idea of universal quantification as a 'generalised conjunction'. However, in general, there is no natural order in which we can enumerate the elements of ω; if we were to start with a, the generalised conjunction would evaluate as undefined, whereas if we started with b, it would yield invalidity. Since as indicated, we wish to avoid to reason about potentially undefined terms, the only safe choice is to consider $\forall x(\alpha)$ as undefined.

3.4 Deduction rules

Now we present the rules that are analogue of the rules of the Hermes calculus. We present abbreviated names of the rules in a box at the top of the rule, to be able to concisely refer to them in the sequel.

Note that the order of the premises is significant; e.g., when we interchange both premises of the &-introduction rule, we get a different sequent, $\Sigma_2, \Sigma_1; \Delta, \Gamma \vdash_\iota \beta \,\&\, \alpha$ as a result:

$$\frac{\begin{array}{c}\Sigma_2; \Delta \vdash_\iota \beta \\ \Sigma_1; \Gamma \vdash_\iota \alpha\end{array}}{\Sigma_2, \Sigma_1; \Delta, \Gamma \vdash_\iota \beta \,\&\, \alpha}$$

Concerning the $\top$ symbol, we introduce the following conventions which will ascertain that the $\top$ symbol never will occur inside sequents, as we already announced:

- If $\top$ would occur inside the context or antecedent of a sequent, it is removed from the context or antecedent. For example, the sequent $\top, \alpha; \beta, \top \vdash_\iota \gamma$ is to be rewritten as $\alpha; \beta \vdash_\iota \gamma$.
- If $\top$ would occur as consequent, the whole sequent is simply dropped. For example, if a proof rule would require $\Sigma; \Gamma \vdash_\iota \top$ as premise, that premise doesn't need to be supplied.

We are now in a position to introduce the rules themselves.

Assumption introduction: $$\frac{\boxed{\text{ass}} \quad UC(\alpha)}{\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha)}{\Sigma; \alpha \vdash_\iota \alpha}$$

When $\boldsymbol{\Delta}(\alpha)$ is $\top$, the context Σ must be empty.

&-introduction: $$\frac{\boxed{\text{\&-intro}} \quad \begin{array}{c}\Sigma_1; \Gamma \vdash_\iota \alpha \\ \Sigma_2; \Delta \vdash_\iota \beta\end{array}}{\Sigma_1, \Sigma_2; \Gamma, \Delta \vdash_\iota \alpha \,\&\, \beta}$$

&-elimination: $$\frac{\boxed{\text{\&-elim}} \quad \Sigma; \Gamma \vdash_\iota \alpha \,\&\, \beta}{\Sigma; \Gamma \vdash_\iota \alpha} \qquad \frac{\boxed{\text{\&-elim}} \quad \Sigma; \Gamma \vdash_\iota \alpha \,\&\, \beta}{\Sigma; \Gamma \vdash_\iota \beta}$$

Removal: $$\frac{\boxed{\text{rem}} \quad \begin{array}{c}\Sigma_1; \Gamma, \alpha \vdash_\iota \beta \\ \Sigma_2; \Delta, \neg\alpha \vdash_\iota \beta\end{array}}{\Sigma_1, \Sigma_2; \Gamma, \Delta \vdash_\iota \beta}$$

Contradiction:
$$\frac{\begin{array}{c}\boxed{\text{contra}}\\ UC(\beta)\\ \Sigma_1;\Gamma\vdash_\iota \alpha\\ \Sigma_2;\Delta\vdash_\iota \neg\alpha\end{array}}{\Sigma_1,\Sigma_2;\Gamma,\Delta\vdash_\iota \beta}$$

∀-introduction:
$$\frac{\boxed{\forall\text{-intro}}\quad \Sigma;\Gamma\vdash_\iota \alpha}{\Sigma;\Gamma\vdash_\iota \forall x(\alpha)}$$
provided x is not free in Γ and Σ

∀-elimination:
$$\frac{\boxed{\forall\text{-elim}}\quad \Sigma;\Gamma\vdash_\iota \forall x(\alpha)}{\Sigma;\Gamma\vdash_\iota \alpha}$$

Substitution:
$$\frac{\begin{array}{c}\boxed{\text{subst}}\\ UC(t)\\ \Sigma;\Gamma\vdash_\iota \alpha\end{array}}{\boldsymbol{\Delta}(t)\,,[t/x]\Sigma;[t/x]\Gamma\vdash_\iota [t/x]\alpha}$$
where $\Gamma \equiv \gamma_1,\gamma_2,\ldots,\gamma_n$. Further, $[t/x]\Gamma$ is shorthand for $[t/x]\gamma_1,[t/x]\gamma_2,\ldots,[t/x]\gamma_n$. This rule can only be applied if all the substitutions are defined.

Equality rules:
$$\frac{\boxed{\text{eq}}\quad UC(t)}{\boldsymbol{\Delta}(t)\vdash_\iota t=t}\qquad \frac{\boxed{\text{eqSubst}}\quad \begin{array}{c}UC(t)\\ \Sigma;\Gamma\vdash_\iota \alpha\end{array}}{\Sigma,\boldsymbol{\Delta}(t)\,;\Gamma,x=t\vdash_\iota [t/x]\alpha}$$
The latter rule can only be applied if the substitution $[t/x]\alpha$ is defined.

We need some extra deduction rules:

ι-rule:
$$\frac{\boxed{\text{iota}}\quad \psi\vdash_\iota \exists!x(\varphi)}{\psi\vdash_\iota [\iota x_\psi(\varphi)/x]\widetilde{\varphi}}$$
where $\widetilde{\varphi}$ is obtained from φ by changing the names of all bound variables, such that they are different from the free variables of φ. This ensures that the substitution $[\widetilde{\iota x_\psi(\varphi)}/x]\widetilde{\varphi}$ is always defined. Formally, $\widetilde{x}\equiv x$; $\widetilde{f}(t_1,\ldots)\equiv f(\widetilde{t_1},\ldots)$; $\widetilde{\iota x_\Psi(\Phi)}\equiv \iota y_{\widetilde{\Psi}}([y/x]\widetilde{\Phi})$ where y is not a free variable of φ and $\widetilde{\Phi}$; $\widetilde{p}(t_1,\ldots)\equiv p(\widetilde{t_1},\ldots)$; $\widetilde{\alpha\,\&\,\beta}\equiv \widetilde{\alpha}\,\&\,\widetilde{\beta}$; $\widetilde{\neg\alpha}\equiv\neg\widetilde{\alpha}$ and $\widetilde{\forall x(\alpha)}\equiv\forall y([y/x]\widetilde{\alpha})$ where y is not a free variable of φ and $\widetilde{\alpha}$.

Note that to keep the notation simple, we do not indicate explicitly which variables are to be avoided in renaming—in this case, the free variables of φ. We call α and $\widetilde{\alpha}$ **alphabetic variant**s of each other.

UC-rule: $$\frac{\boxed{\text{UC}}\quad \Sigma;\Gamma \vdash_\iota \alpha}{\psi \vdash_\iota \exists!x(\varphi)}$$
where $\iota x_\psi(\varphi)$ is a ι-term occurring in Σ, Γ or α.

$\mathbf{\Delta}$-rules; only applicable when the consequent of the conclusion is not $\top$:

$$\frac{\boxed{\text{defAnt}}\quad \Sigma;\Gamma,\alpha \vdash_\iota \beta}{\Sigma;\vdash_\iota \mathbf{\Delta}(\alpha)} \qquad \frac{\boxed{\text{defCons}}\quad \Sigma;\Gamma \vdash_\iota \alpha}{\Sigma;\Gamma \vdash_\iota \mathbf{\Delta}(\alpha)}$$

Finally, we add these rules for manipulating contexts:

$$\frac{\boxed{\text{toCtxt}}\quad \Sigma;\sigma \,\&\, \Gamma,\Delta \vdash_\iota \alpha}{\Sigma,\sigma;\Gamma,\Delta \vdash_\iota \alpha} \qquad \frac{\boxed{\text{fromCtxt}}\quad \Sigma,\sigma;\Gamma \vdash_\iota \alpha}{\Sigma;\sigma \,\&\, \Gamma \vdash_\iota \alpha}$$

The notation $\sigma \,\&\, \Gamma$ stands for $\sigma \,\&\, \gamma_1, \sigma \,\&\, \gamma_2, \ldots, \sigma \,\&\, \gamma_n$; when Γ is empty it is shorthand for σ.

Note that in rules where the conclusion has Σ_1, Σ_2 as context, we could just as well have chosen Σ_2, Σ_1; we just had to fix an order to create our new deduction rules. Indeed: in the sequel (see the discussion of the WeakCtxtL rule), we will show that using the deduction rules, we are able do derive from $\Sigma_1;\Gamma \vdash_\iota \alpha$ and $\Sigma_2;\Delta \vdash_\iota \beta$ the sequent $\Sigma_2,\Sigma_1;\Gamma,\Delta \vdash_\iota \alpha \& \beta$, and the situation for the rem and contra rules is similar.

We call a sequent **pitfol derivable** if it can be obtained by applying a deduction rule of the PITFOL calculus with PITFOL derivable sequents as premises.

A **pitfol proof** is a finite list of PITFOL sequents $s_1, s_2, \ldots, s_n$ where each s_i is obtained by applying one of the deduction rules with as premises sequents obtained earlier in the proof; i.e., if s_j is used as a premise, then $j < i$.

Note that, as we remarked before, the order in which the premises are supplied to a deduction rule is significant, but we will not record the specific order in the proof, just like we do not record the exact deduction rule used in the proof.

Note that a proof of the Hermes calculus is automatically a PITFOL proof; in this view, our PITFOL calculus is an **extension** of the Hermes calculus (which will turn out to be a **conservative extension**; see §3.6.2). Indeed, for formulae of the Hermes calculus, $\mathbf{\Delta}(\alpha)$ is $\top$, and it is easy to see that the proof rules of the Hermes calculus coincide under those circumstances with

the proof rules of the PITFOL calculus. We believe that this property yields a logic close to common mathematical practice, where one reasons (or at least often claims to do so) in a two-valued setting, with the addition that one has to make sure that the 'definedness conditions' are fulfilled (e.g., that $x \neq 0$ when reasoning about $\frac{1}{x}$).

In the sequel, we will again define the symbols $\vee$, $\Rightarrow$, $\Leftrightarrow$ and $\exists$ in the same way as in the Hermes calculus.

3.5 Discussion of the rules

Before we will show that the calculus introduced above is consistent (§3.6), sound (§3.7) and complete (§3.9), we will discuss some peculiarities of the calculus.

3.5.1 Commutativity of the conjunction

We remarked that the conjunction is not commutative in the PITFOL calculus; however, examining deduction rules, this may not be readily apparent. The only rules concerning conjunction seem to be the &-introduction and &-elimination rules, which are unchanged from the original calculus. Indeed, using these rules, from $\Sigma; \Gamma \vdash_\iota \alpha \,\&\, \beta$, we can easily obtain $\Sigma; \Gamma \vdash_\iota \beta \,\&\, \alpha$.

However, if we consider the assumption introduction rule, we observe that it may be the case that we can derive $\Sigma; \alpha \,\&\, \beta \vdash_\iota \alpha \,\&\, \beta$, but not $\Sigma; \beta \,\&\, \alpha \vdash_\iota \beta \,\&\, \alpha$, because we can derive $\Sigma; \vdash_\iota \mathbf{\Delta}(\alpha \,\&\, \beta)$ but not $\Sigma; \vdash_\iota \mathbf{\Delta}(\beta \,\&\, \alpha)$.

For example, consider the example at the end of §3.1 where we set $\alpha \equiv y \neq 0$ and $\beta \equiv \frac{1}{y} = 5$; we can derive $\vdash_\iota \mathbf{\Delta}(\alpha \,\&\, \beta)$, which is

$$\vdash_\iota \underbrace{\mathbf{\Delta}(y \neq 0)}_{\equiv \top} \;\&\; y \neq 0 \Rightarrow \underbrace{\mathbf{\Delta}\left(\tfrac{1}{y} = 5\right)}_{\equiv y \neq 0},$$

i.e., $\vdash_\iota y \neq 0 \Rightarrow y \neq 0$, but we clearly cannot derive $\vdash_\iota \mathbf{\Delta}(\beta \,\&\, \alpha)$, which is $\vdash_\iota \mathbf{\Delta}\left(\frac{1}{y} = 5\right)$, i.e., $\vdash_\iota y \neq 0$.

Hence, we conclude that the non-commutativity of the conjunction manifests itself in the calculus because of the definition of $\mathbf{\Delta}$.

3.5.2 Definedness of generalisations

We defined $\mathbf{\Delta}(\forall x(\alpha))$ as $\forall x(\mathbf{\Delta}(\alpha))$. If one considers a generalisation as a 'generalised conjunction', this might seem surprising since one might expect

the noncummutativity of the definedness of a conjunction to surface here too. For example, suppose we have a finite domain $\omega = \{a_1, a_2, \ldots, a_n\}$ which can be described by terms of the logic, i.e., there exist terms $t_1, \ldots, t_n$ such that $\mathcal{I}t_i = a_i$ for $i = 1 \ldots n$. Then the generalised conjunction of α with respect to x is defined as

$$[t_1/x]\,\alpha \,\&\, [t_2/x]\,\alpha \,\&\, \cdots \,\&\, [t_n/x]\,\alpha.$$

In the classical two-valued setting, one expects this formula to be equivalent with $\forall x(\alpha)$; hence we call the $\forall$ quantifier a generalised conjunction.

In our three-valued setting, in general, the interpretation of

$$[t_1/x]\,\alpha \,\&\, \cdots \,\&\, [t_n/x]\,\alpha$$

depends on the order in which the t_i are chosen. Indeed, if for example $[t_1/x]\,\alpha$ is invalid and $[t_2/x]\,\alpha$ is undefined, then

$$\begin{array}{ll} [t_1/x]\,\alpha \,\&\, [t_2/x]\,\alpha \,\&\, \cdots \,\&\, [t_n/x]\,\alpha & \text{is invalid} \\ [t_2/x]\,\alpha \,\&\, [t_1/x]\,\alpha \,\&\, \cdots \,\&\, [t_n/x]\,\alpha & \text{is undefined} \end{array}$$

It seems natural to require that $\forall x(\alpha)$ to be considered valid only when all possible generalised conjunctions of α are valid. In particular, for each $i = 1 \ldots n$, we require that

$$[t_i/x]\,\alpha \,\&\, [t_1/x]\,\alpha \,\&\, \cdots \,\&\, [t_{i-1}/x]\,\alpha \,\&\, [t_{i+1}/x]\,\alpha \,\&\, \cdots \,\&\, [t_n/x]\,\alpha$$

should be valid, from which it easily follows that each $[t_i/x]\,\alpha$ should be valid. Hence we have no other choice than to define $\mathbf{\Delta}(\forall x(\alpha)) \equiv \forall x(\mathbf{\Delta}(\alpha))$.

3.5.3 $\mathbf{\Delta}$ and $\top$

One could wonder why we did not define $\mathbf{\Delta}$ to always yield a formula. In the cases where we used $\top$, at first sight we could just as well have used a validity τ, such as $\forall x(x = x)$ or $p \vee \neg p$ where p is a predicate constant.

The first drawback of avoiding the $\top$ symbol is that $\mathbf{\Delta}$ would yield very large and unwieldy formulae; for example, we would get

$$\begin{aligned} \mathbf{\Delta}(x = x \,\&\, y = y) &\equiv \mathbf{\Delta}(x = x) \,\&\, (x = x \Rightarrow \mathbf{\Delta}(y = y)) \\ &\equiv (\tau \,\&\, \tau) \,\&\, (x = x \Rightarrow (\tau \,\&\, \tau)) \end{aligned}$$

This complicates formal proofs considerably; for example, to apply the assumption rule on the formula $x = x \,\&\, y = y$, we need to derive first

$$\vdash_\iota (\tau \,\&\, \tau) \,\&\, (x = x \Rightarrow (\tau \,\&\, \tau))$$

which is not straightforward. To illustrate how much even simple proofs are affected, just deriving $\vdash_\iota x = x$ is not trivial, since the equality rule now yields the more complicated

$$\tau \,\&\, \tau \vdash_\iota (x = x)$$

In general, it seems necessary to add $\vdash_\iota \tau$ as an axiom to the calculus; for some specific forms of τ such as $\forall x(x = x)$ we can derive $\vdash_\iota \tau$ however:

$$\begin{array}{lr}
\forall x(x = x) \,\&\, \forall x(x = x) \vdash_\iota (x = x) & \text{eq} \\
\quad \vdash_\iota \forall x(\forall x(x = x) \,\&\, \forall x(x = x)) & \\
\qquad \&\, (\forall x(x = x) \Rightarrow \forall x(x = x) \,\&\, \forall x(x = x)) & \text{defAnt} \\
\quad \vdash_\iota \forall x(\forall x(x = x) \,\&\, \forall x(x = x)) & \&\text{-elim} \\
\quad \vdash_\iota \forall x(x = x) \,\&\, \forall x(x = x) & \forall\text{-elim} \\
\quad \vdash_\iota \forall x(x = x) & \&\text{-elim} \\
\quad \vdash_\iota x = x & \forall\text{-elim}
\end{array}$$

Moreover, if we drop the use of $\top$, a proof of the Hermes calculus using the assumption introduction rule, the substitution rule or the equality rules is not any more a proof of the PITFOL calculus and hence the PITFOL calculus would not be an extension of the original calculus.

Note that for theoretical purposes, it can be interesting to define a variant of $\boldsymbol{\Delta}$ along the lines sketched above, which is what we will do in §3.6 with the definition of $\mathcal{D}$. Indeed, for theoretical investigations where it is not important that the resulting formulae are small, the definition of $\mathcal{D}$ has as advantage that it has less different cases and we do not need to resort to the introduction of the symbol $\top$ and its simplification rules.

3.5.4 A note about contexts

It seems that we don't *have* to extend the notion of sequent with a context. In this variant of the calculus, some deduction rules look different:

$$\boxed{\text{subst}}\quad \frac{\begin{array}{r} UC(t) \\ \Gamma \vdash_\iota \alpha \end{array}}{\boldsymbol{\Delta}(t) \,\&\, [t/x]\Gamma \vdash_\iota [t/x]\alpha} \qquad \boxed{\text{eqSubst}}\quad \frac{\begin{array}{r} UC(t) \\ \Gamma \vdash_\iota \alpha \end{array}}{\Gamma, \boldsymbol{\Delta}(t) \,\&\, x = t \vdash_\iota [t/x]\alpha}$$

and instead of toCtxt and fromCtxt, we would have

$$\boxed{\text{assCtxt}}\quad \frac{\begin{array}{r} \vdash_\iota \boldsymbol{\Delta}(\gamma) \\ \gamma \vdash_\iota \boldsymbol{\Delta}(\alpha) \end{array}}{\gamma \,\&\, \alpha \vdash_\iota \alpha} \qquad \boxed{\text{remCtxt}}\quad \frac{\begin{array}{r} \Gamma, \gamma \,\&\, \alpha \vdash_\iota \beta \\ \Delta, \gamma \,\&\, \neg\alpha \vdash_\iota \beta \end{array}}{\Gamma, \Delta, \gamma \vdash_\iota \beta}$$

The advantage of this approach is that the notion of sequent (and hence of proof) is simpler, but it turns out to be much more cumbersome to actually prove things in this variant of the calculus. Most derived rules have a variant with and without 'context', e.g.

$$\frac{\Gamma, \neg\alpha \vdash_\iota \alpha}{\Gamma \vdash_\iota \alpha}\ \boxed{\text{SeAs}} \qquad \frac{\Gamma, \gamma \,\&\, \neg\alpha \vdash_\iota \alpha}{\Gamma, \gamma \vdash_\iota \alpha}\ \boxed{\text{SeAsCtxt}}$$

The clumsiness of the contextless calculus is illustrated further by the derivation of the Cut2 rule, where we are (at least not in an obvious way) unable to reuse the Cut rule, but have to invent a completely new proof:

$$\begin{array}{rr}
\Gamma \vdash_\iota \alpha & \text{prem} \\
\gamma \,\&\, \alpha \vdash_\iota \beta & \text{prem} \\
\vdash_\iota \boldsymbol{\Delta}(\gamma \,\&\, \alpha) & \text{defAnt} \\
\gamma \,\&\, \neg\alpha \vdash_\iota \gamma \,\&\, \neg\alpha & \text{ass} \\
\gamma \,\&\, \neg\alpha \vdash_\iota \neg\alpha & \&\text{-elim} \\
\Gamma, \gamma \,\&\, \neg\alpha \vdash_\iota \beta & \text{contra} \\
\Gamma, \gamma \vdash_\iota \beta & \text{remCtxt}
\end{array}$$

In contrast, compare the forthcoming proof of the Cut2 rule in the calculus with contexts, where we are indeed able to reuse the Cut rule.

One could also go to the other extreme and define sequents of the form

$$\sigma_1^1, \sigma_2^1, \ldots; \sigma_1^2, \sigma_2^2, \ldots; \ldots; \sigma_1^n, \sigma_2^n, \ldots \Gamma \vdash \alpha$$

where the order of the formulae in $\sigma_1^1, \sigma_2^1, \ldots$ is not important and double formulae are to be removed; likewise in $\sigma_1^2, \sigma_2^2, \ldots$, and so on. The idea here is that semantically, one evaluates first the sequents of the first list $\sigma_1^1, \sigma_2^1, \ldots$ in any random order, then those of the second list $\sigma_1^2, \sigma_2^2, \ldots, \ldots$ One would then need some rules specifying when a formula is allowed to move to the left or right of a semicolon:

$$\frac{\ldots; \Sigma_1; \sigma \,\&\, \Sigma_2, \Sigma_3; \ldots \vdash_\iota \alpha}{\ldots; \Sigma_1, \sigma; \Sigma_2, \Sigma_3; \ldots \vdash_\iota \alpha}\ \boxed{\text{LeftCtxt}} \qquad \frac{\ldots; \Sigma_1, \sigma; \Sigma_2; \ldots \vdash_\iota \alpha}{\ldots; \Sigma_1; \sigma \,\&\, \Sigma_2; \ldots \vdash_\iota \alpha}\ \boxed{\text{RightCtxt}}$$

It seems that all this extra complication in syntax would yield little gain in usability, so we opted not to pursue this option.

3.6 Equiconsistency proof

3.6.1 Translation of sequents

In this section, we will give a process to transform a sequent in the PITFOL calculus to one or more sequents in the Hermes calculus.

We define two metalogical operations on formulae α of the PITFOL calculus: the reduction $\mathcal{R}(\alpha)$ and the definedness $\mathcal{D}(\alpha)$. Both produce a formula of the Hermes calculus. Semantically, $\mathcal{R}(\alpha)$ will express that if α is defined, it is true (or equivalently, that α is undefined or true), and $\mathcal{D}(\alpha)$ will express that α is defined (we will actually prove this in lemma 20).

We define $\mathcal{R}$ by structural induction on the formula α:

- α is an atomic formula, i.e., $\alpha \equiv t_1 = t_2$ or $\alpha \equiv p(t_1, t_2, \ldots, t_n)$. We will only cover the latter case; the former is analogous.

  If there are no ι-term arguments of p, then we define $\mathcal{R}(\alpha) \equiv \alpha$.

  Else, we enumerate all occurrences of top-level ι-terms of α (i.e., those ι-terms that are themselves not contained into another ι-term) as $TLI(\alpha) \equiv \iota x_{1\psi_1}(\varphi_1)$, $\iota x_{2\psi_2}(\varphi_2)$, $\ldots$, $\iota x_{m\psi_m}(\varphi_m)$. We define $\mathcal{R}(\alpha)$ as

  $$\exists u_1 \exists u_2 \ldots \exists u_m \left(\mathcal{R}([{}^{u_1}\!/_{x_1}]\varphi_1) \,\&\, \mathcal{R}([{}^{u_2}\!/_{x_2}]\varphi_2) \,\&\, \ldots \,\&\, \mathcal{R}([{}^{u_m}\!/_{x_m}]\varphi_m) \,\&\, q\right)$$

  where q is the formula obtained from α by replacing all top-level ι-terms $\iota x_{i\psi_i}(\varphi_i)$ by their corresponding variable symbol u_i. We choose the u_i's such that they are all different from each other and such that u_i does not occur in α.
  Formally, we can construct q as follows. Choose the m different variable symbols $u_1, \ldots, u_m$ not occurring in α. We then define a metalogical operator Q which maps a term t to another term $Q(t)$, as follows:

  - $Q(x) \equiv x$
  - $Q(f(\tau_1, \ldots, \tau_k)) \equiv f(Q(\tau_1), \ldots, Q(\tau_k))$ with special cases $Q(f(\tau)) \equiv f(Q(\tau))$ and $Q(c) \equiv c$ where c is a constant (which we identified with nullary function symbols)
  - $Q(\iota x_{i\psi_i}(\varphi_i)) \equiv u_i$ where i is the index of this ι-term in the list op top-level ι-terms of the original formula α.

  Then we have $q \equiv p(Q(t_1), Q(t_2), \ldots, Q(t_n))$ if $n > 1$; $q \equiv p(Q(t_1))$ if $n = 1$ and $q \equiv \alpha$ if $n = 0$.

- $\mathcal{R}(\neg\alpha) \equiv \neg\mathcal{R}(\alpha)$

- $\mathcal{R}(\alpha \,\&\, \beta) \equiv \mathcal{R}(\alpha) \,\&\, \mathcal{R}(\beta)$
- $\mathcal{R}(\forall x(\alpha)) \equiv \forall x(\mathcal{R}(\alpha))$

Note that this is the same definition as in [Hilbert & Bernays 1968], except that there, $u_i \equiv x_i$ is chosen (which is incorrect when two or more x_i's are the same variable symbol) and we don't identify multiple occurrences of the same ι-term (each occurrence gets its own u_i).

One sees that the set of free variables of α is the same as that of $\mathcal{R}(\alpha)$.

We illustrate the definition of $\mathcal{R}$ on atomic formulae with the following example. Consider the formula $\alpha :\equiv \frac{1}{x} = \frac{1}{y}$, i.e., $\iota z_{x\neq 0}(x{\cdot}z = 1) = \iota z_{y\neq 0}(y{\cdot}z = 1)$. Then we have

$$\begin{aligned}\mathcal{R}(\iota z_{x\neq 0}(x \cdot z = 1) = \iota z_{y\neq 0}(y \cdot z = 1)) \\ &\equiv \exists u_1 \exists u_2(\mathcal{R}(x \cdot u_1 = 1) \,\&\, \mathcal{R}(y \cdot u_2 = 1) \,\&\, u_1 = u_2) \\ &\equiv \exists u_1 \exists u_2(x \cdot u_1 = 1 \,\&\, y \cdot u_2 = 1 \,\&\, u_1 = u_2)\end{aligned}$$

Semantically, if α is defined (i.e., if x and y are both nonzero), this expresses that u_1 is an inverse of x (and since we have the uniqueness condition, u_1 is *the* inverse of x), that u_2 is the inverse of y and that they are equal, i.e., α is valid.
If α is undefined, for example when $x = 0$, then $\mathcal{R}(\alpha)$ is equivalent with

$$\exists u_1 \exists u_2(0 = 1 \,\&\, y \cdot u_2 = 1 \,\&\, u_1 = u_2)$$

which is of course invalid.

Note that if we have the uniqueness conditions, $\mathcal{R}(\alpha)$ is equivalent with

$$\forall u_1 \forall u_2((x \cdot u_1 = 1 \,\&\, y \cdot u_2 = 1) \Rightarrow u_1 = u_2)$$

so we could have defined the reduction of an atomic formula as

$$\forall u_1 \forall u_2 \ldots \forall u_m\,((\mathcal{R}([u_1/x_1]\varphi_1) \,\&\, \mathcal{R}([u_2/x_2]\varphi_2) \,\&\, \ldots \,\&\, \mathcal{R}([u_m/x_m]\varphi_m)) \Rightarrow q)$$

See also lemma 12, where we show that given one form of the definition, we can derive the other.

Another example with nested iota terms is given by the formula $\frac{x}{\sqrt{y}} = 1$:

$$\begin{aligned}\mathcal{R}\big(\iota z_{y\geq 0 \& \iota w_{y\geq 0}(w\geq 0 \& w\cdot w = y)\neq 0}(z \cdot \iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y) = x) = 1\big) \\ &\equiv \exists u_1(\mathcal{R}(u_1 \cdot \iota w_{y\geq 0}(w \geq 0 \,\&\, w \cdot w = y) = x) \,\&\, u_1 = 1) \\ &\equiv \exists u_1(\exists u_2(\mathcal{R}(u_2 \geq 0 \,\&\, u_2 \cdot u_2 = y) \,\&\, u_1 \cdot u_2 = x) \,\&\, u_1 = 1) \\ &\equiv \exists u_1(\exists u_2(u_2 \geq 0 \,\&\, u_2 \cdot u_2 = y \,\&\, u_1 \cdot u_2 = x) \,\&\, u_1 = 1)\end{aligned}$$

Semantically, this expresses that $u_2 \equiv \sqrt{y}$, $u_1 = \frac{x}{u_2}$ and $u_1 = 1$.

We will prove shortly that the exact choice of the u's is not relevant: different choices yield equivalent reductions. Sometimes, we wish to avoid a certain choice of variable symbol for the u's, e.g., when the substitution $[t/x]\,\alpha$ is defined, the substitution $[t/x]\,\mathcal{R}(\alpha)$ is not necessarily defined because one or more of the u_i's might accidentally be a free variable of t. Hence given a set of variable symbols V, we define $\mathcal{R}_V(\alpha)$ in the same way as $\mathcal{R}(\alpha)$, but with the extra demand that all u's be different from all variables in V. Just like $\mathcal{R}$, $\mathcal{R}_V$ is defined inductively so we define $\mathcal{R}_V(\neg\alpha) \equiv \neg\mathcal{R}_V(\alpha)$ and so on. We call V the **exclusion set** of the reduction. If x is a variable symbol, then we will with a slight abuse of notation write $\mathcal{R}_x$ instead of $\mathcal{R}_{\{x\}}$ and $\mathcal{R}_{V,x}$ instead of $\mathcal{R}_{V\cup\{x\}}$.

Next, we define $\mathcal{D}$:

- $\mathcal{D}_V(p(t_1, t_2, \ldots, t_n)) \equiv \mathcal{R}_V(\psi_1)\,\&\,\mathcal{R}_V(\psi_2)\,\&\ldots\&\,\mathcal{R}_V(\psi_n)$ with the same notations as in the definition of $\mathcal{R}$. If there are no ι-terms as argument of p, then $\mathcal{D}_V(p(\ldots)) \equiv \forall x(x = x)$. We treat $\mathcal{D}_V(t_1 = t_2)$ analogously. Note that it would be natural to define $\mathcal{D}_V(p(t_1, t_2, \ldots, t_n)) \equiv \mathcal{R}_V(\psi_1)\&$ $\mathcal{D}_V(\psi_1)\,\&\ldots\&\,\mathcal{R}_V(\psi_n)\,\&\,\mathcal{D}_V(\psi_n)$ but in the sequel (see page 61) it will appear that we get $\mathcal{D}_V(\psi_i)$ from the uniqueness condition for $\iota x_{i\psi_i}(\varphi_i)$.
- $\mathcal{D}_V(\neg\alpha) \equiv \mathcal{D}_V(\alpha)$
- $\mathcal{D}_V(\alpha\ \&\ \beta) \equiv \mathcal{D}_V(\alpha)\ \&\ (\mathcal{R}_V(\alpha) \Rightarrow \mathcal{D}_V(\beta))$
- $\mathcal{D}_V(\forall x(\alpha)) \equiv \forall x(\mathcal{D}_V(\alpha))$

If V is the empty set, we write $\mathcal{D}$ instead of $\mathcal{D}_{\{\}}$; if V is a singleton $\{x\}$, we write $\mathcal{D}_x$ instead of $\mathcal{D}_{\{x\}}$.

When t is a term, we define $\mathcal{D}_V(t) \equiv \mathcal{D}_V(t = x)$.

Note that in contrast to $\mathbf{\Delta}$, the result of $\mathcal{D}$ can never be $\top$ and hence the resulting formulae will be longer. As we already indicated in §3.5.3, for theoretical purposes it is more interesting that there are less cases in the definition of $\mathcal{D}$.

Lemma 8 *For each formula α of the* PITFOL *calculus,*

1. *if the substitution $[y/x]\,\alpha$ is defined, then*

$$\mathcal{R}_V([y/x]\,\alpha) \dashv\vdash [y/x]\,\mathcal{R}_{V,y}(\alpha)$$

where V is a set of variable symbols.

2. *choosing different u's in the reduction of α yields equivalent reductions (and hence, $\mathcal{R}_V(\alpha) \dashv\vdash \mathcal{R}_W(\alpha)$ for any two sets of variable symbols V and W)*

Note that it is necessary to use $\mathcal{R}_{V,y}$ to ensure that the substitution $[y/x]\mathcal{R}_{V,y}(\alpha)$ is defined. As a counterexample, we take $\alpha \equiv p(\iota z(x = z))$. Then, the substitution

$$[y/x]\mathcal{R}(\alpha) \equiv [y/x]\exists u_1(x = u_1 \,\&\, p(u_1))$$

is not defined when we would choose $u_1 \equiv y$.

Proof.

We prove this by induction the complexity of α; we will at the same time prove that the exact choice of the u's in the reduction does not matter.

- If α has complexity zero, i.e., α is atomic and does not contain any ι-terms, then

  $$\mathcal{R}_V([y/x]\alpha) \equiv [y/x]\alpha \equiv [y/x]\mathcal{R}_{V,y}(\alpha)$$

  because the reduction of a formula without ι-terms is that formula itself.

- *1.* If α is atomic and contains ι-terms, i.e., $\alpha \equiv p(t_1, \ldots, t_n)$, where we treat the case $\alpha \equiv t_1 = t_2$ analogously, then we have to prove that the formula $\mathcal{R}_V([y/x]\alpha)$, or more explicitly,

$$\exists v_1 \exists v_2 \ldots \exists v_n \,(\mathcal{R}_V([v_1/x_1]\varphi_1') \,\&\, \mathcal{R}_V([v_2/x_2]\varphi_2') \,\&\, \ldots \,\&\, \mathcal{R}_V([v_m/x_m]\varphi_m') \,\&\, q'))$$

  is equivalent to the formula $[y/x]\mathcal{R}_{V,y}(\alpha)$, i.e., equivalent to

  $$[y/x]\exists u_1 \exists u_2 \ldots \exists u_n \,(\mathcal{R}_{V,y}([u_1/x_1]\varphi_1) \,\&\, \ldots \,\&\, \mathcal{R}_{V,y}([u_m/x_m]\varphi_m) \,\&\, q))$$

  with the usual notations: the v_i are all different from each other and do not occur in $[y/x]\alpha$ or V; the u_i are all different from each other, do not occur in α or V and are different from y; q' is the formula obtained from $[y/x]\alpha$ by replacing its top-level ι-terms by v's and q is obtained likewise from α by replacing its top-level ι-terms by u's; the top-level ι-terms of α are $TLI(\alpha) \equiv \iota x_{1\psi_1}(\varphi_1), \ldots, \iota x_{m\psi_m}(\varphi_m)$ and the top-level ι-terms of $[y/x]\alpha$ are $TLI([y/x]\alpha) \equiv \iota x_{1\psi_1'}(\varphi_1'), \ldots, \iota x_{m\psi_m'}(\varphi_m')$.

  It is easy to see that $[y/x]\iota x_{i\psi_i}(\varphi_i) \equiv \iota x_{i\psi_i'}(\varphi_i')$. Hence, φ_i' is the definiens of $[y/x]\iota x_{\psi_i}(\varphi_i)$, i.e.:

- If $x \equiv x_i$ or x is not a free variable of φ_i, then $[y/x]\iota x_{\psi_i}(\varphi_i) \equiv \iota x_{\psi_i}(\varphi_i)$ or $[y/x]\iota x_{\psi_i}(\varphi_i) \equiv \iota x_{\mathbf{\Delta}(y)\&[y/x]\psi_i}(\varphi_i) \equiv \iota x_{[y/x]\psi_i}(\varphi_i)$, hence $\varphi_i' \equiv \varphi_i$.
- If $x \not\equiv x_i$ and x is a free variable of φ_i, then $[y/x]\iota x_{i\psi_i}(\varphi_i) \equiv \iota x_{i\,\mathbf{\Delta}(y)\&[y/x]\psi_i}([y/x]\varphi_i) \equiv \iota x_{i[y/x]\psi_i}([y/x]\varphi_i)$. (We take the substitution to be defined, hence $x_i \not\equiv y$.) So in this case, $\varphi_i' \equiv [y/x]\varphi_i$.

We first establish that

$$\mathcal{R}_V([u_i/x_i]\varphi_i') \dashv\vdash [y/x]\mathcal{R}_{V,y}([u_i/x_i]\varphi_i)$$

or, using induction on φ_i,

$$\mathcal{R}_V([u_i/x_i]\varphi_i') \dashv\vdash \mathcal{R}_V([y/x][u_i/x_i]\varphi_i)$$

- If $x \equiv x_i$, then we have to prove that

  $$\mathcal{R}_V([u_i/x]\varphi_i) \dashv\vdash \mathcal{R}_V([y/x][u_i/x]\varphi_i)$$

  Since $u_i \not\equiv x$, this reduces using corollary 6 to

  $$\mathcal{R}_V([u_i/x]\varphi_i) \dashv\vdash \mathcal{R}_V([u_i/x]\varphi_i)$$

- If x is not a free variable of φ_i, then we have to prove that

  $$\mathcal{R}_V([u_i/x_i]\varphi_i) \dashv\vdash \mathcal{R}_V([y/x][u_i/x_i]\varphi_i)\,.$$

  Noting that u_i is different from x, we can apply property 4, yielding that x is not a free variable of $[u_i/x_i]\varphi_i$, so using property 5, $[y/x][u_i/x_i]\varphi_i \equiv [u_i/x_i]\varphi_i$.
- If x is a free variable of φ_i and $x \not\equiv x_i$, then we are left to show that

  $$\mathcal{R}_V([u_i/x_i][y/x]\varphi_i) \dashv\vdash \mathcal{R}_V([y/x][u_i/x_i]\varphi_i)$$

  Since we supposed the substitution to be defined, $x_i \not\equiv y$; we also have that u_i does not occur in α and hence $u_i \not\equiv x$. Hence property 7 is applicable, which shows that we may change the order of the substitutions without affecting the result.

Having established this, what remains to prove is

$$\exists v_1 \exists v_2 \ldots \exists v_m\,(\mathcal{R}_V([v_1/x_1]\varphi_1') \,\&\, \mathcal{R}_V([v_2/x_2]\varphi_2') \,\&\, \ldots \,\&\, \mathcal{R}_V([v_m/x_m]\varphi_m') \,\&\, q')$$

is equivalent to

$$\exists u_1 \exists u_2 \ldots \exists u_m \, (\mathcal{R}_V([u_1/x_1]\varphi_1') \,\&\, \ldots \,\&\, \mathcal{R}_V([u_m/x_m]\varphi_m') \,\&\, [y/x]q)$$

which follows from the next part, since this are just two reductions that only differ in the choice of the u_i.

2. We have to show that reductions with different choices of u are equivalent, i.e.,

$$\exists u_1 \exists u_2 \ldots \exists u_m \, (\mathcal{R}([u_1/x_1]\varphi_1) \,\&\, \mathcal{R}([u_2/x_2]\varphi_2) \,\&\, \ldots \,\&\, \mathcal{R}([u_m/x_m]\varphi_m) \,\&\, q) \tag{3.1}$$

is equivalent with

$$\exists u_1' \exists u_2' \ldots \exists u_m' \, (\mathcal{R}([u_1'/x_1]\varphi_1) \,\&\, \mathcal{R}([u_2'/x_2]\varphi_2) \,\&\, \ldots \,\&\, \mathcal{R}([u_m'/x_m]\varphi_m) \,\&\, q')$$

with the usual notations (this time, q and q' are both obtained from α by replacing its top-level ι-terms by u's, respectively by u''s). We may suppose that the u's and u''s are all different: if there would be common variables, then we can choose a set of u'''s which are different from the u's and the u'''s; the reduction with the u's then is equivalent with the reduction with the u'''s which is in turn equivalent to the one with the u''s.

Using the induction hypothesis, we have that $\mathcal{R}([u_i/x_i]\varphi_i) \dashv\vdash [u_i/x_i]\mathcal{R}_{u_i}(\varphi_i)$. Hence (3.1) is equivalent with

$$\exists u_1 \exists u_2 \ldots \exists u_m \, ([u_1/x_1]\mathcal{R}_{u_1}(\varphi_1) \,\&\, \ldots \,\&\, [u_m/x_m]\mathcal{R}_{u_m}(\varphi_m) \,\&\, q)$$

We can now perform a change of variables, giving

$$\exists u_1' \ldots \exists u_m' \, ([u_1'/x_1]\mathcal{R}_{u_1}(\varphi_1) \,\&\, \ldots \,\&\, [u_m'/x_m]\mathcal{R}_{u_m}(\varphi_m) \,\&\, [u_1'/u_1]\cdots[u_m'/u_m]q)$$

and it is easy to see that $[u_1'/u_1][u_2'/u_2]\cdots[u_m'/u_m]q \equiv q'$. Now we only have to establish that $[u_i'/x_i]\mathcal{R}_{u_i}(\varphi_i) \dashv\vdash \mathcal{R}([u_i'/x_i]\varphi_i)$. Since by induction $\mathcal{R}_{u_i}(\varphi_i) \dashv\vdash \mathcal{R}_{u_i'}(\varphi_i)$, we have that

$$[u_i'/x_i]\mathcal{R}_{u_i}(\varphi_i) \dashv\vdash [u_i'/x_i]\mathcal{R}_{u_i'}(\varphi_i) \dashv\vdash \mathcal{R}([u_i'/x_i]\varphi_i)$$

- The cases where α is not atomic are straightforward.

□

We define the **translation** of a PITFOL sequent $\Sigma; \Gamma \vdash_\iota \alpha$ as the three sequents

$$\begin{cases} \vdash \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \ldots \,\&\, \sigma_n) \\ \mathcal{R}(\sigma_1), \mathcal{R}(\sigma_2), \ldots, \mathcal{R}(\sigma_n) \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\sigma_1), \ldots, \mathcal{R}(\sigma_n), \mathcal{R}(\gamma_1), \ldots, \mathcal{R}(\gamma_m) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

where $\Sigma \equiv \sigma_1, \sigma_2, \ldots, \sigma_n$ and $\Gamma \equiv \gamma_1, \gamma_2, \ldots, \gamma_m$

We will often write with a slight abuse of notation $\mathcal{R}(\Sigma)$ instead of $\mathcal{R}(\sigma_1), \mathcal{R}(\sigma_2), \ldots, \mathcal{R}(\sigma_n)$ and likewise $\mathcal{R}(\Gamma)$ instead of $\mathcal{R}(\gamma_1), \mathcal{R}(\gamma_2), \ldots$

In case Σ is empty, we define the translation as

$$\begin{cases} \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

If Γ is empty, we define the translation as

$$\begin{cases} \vdash \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \ldots \,\&\, \sigma_n) \\ \mathcal{R}(\Sigma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

When both Σ and Γ are empty, the translation of $\vdash_\iota \alpha$ is defined as the single sequent

$$\vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha)\,.$$

3.6.2 Translation of proofs

We define the translation of a PITFOL proof $s_1, s_2, \ldots, s_n$ as a list of sequents of the Hermes calculus which we obtain by replacing all the sequents of the given PITFOL proof by their translations.

We will show in the following sections that the translation of a PITFOL proof can be expanded to a proof (in the Hermes calculus). To do this, it is sufficient to prove for each deduction rule of the PITFOL calculus that when we replace the premises by their translations, we can derive in the Hermes calculus the sequent(s) of the translation of the conclusion.

Note that the translation of a PITFOL proof need not be a correct proof: often we need to add some 'glue sequents' in between. For example, consider the PITFOL proof

$$\begin{array}{ll} \vdash_\iota x = x & \text{eq} \\ \vdash_\iota y = y & \text{eq} \\ \vdash_\iota x = x \,\&\, y = y & \&\text{-intro} \end{array}$$

Its translation is

$$\vdash x = x \,\&\, \forall x(x = x)$$
$$\vdash y = y \,\&\, \forall x(x = x)$$
$$\vdash x = x \,\&\, y = y \,\&\, \forall x(x = x) \,\&\, x = x \Rightarrow \forall x(x = x)$$

which is not a valid proof but can be easily expanded into one (note that $\neg(x = x \,\&\, \neg\forall x(x = x))$ is an abbreviation for $x = x \Rightarrow \forall x(x = x)$):

$$\begin{array}{rll}
 & \vdash x = x & \text{eq} \\
 & \vdash \forall x(x = x) & \forall\text{-intro} \\
 & \boldsymbol{\vdash x = x \,\&\, \forall x(x = x)} & \&\text{-intro} \\
 & \vdash y = y & \text{eq} \\
 & \vdash y = y \,\&\, \forall x(x = x) & \&\text{-intro} \\
 & \vdash x = x \,\&\, y = y & \&\text{-intro} \\
 & \boldsymbol{\vdash x = x \,\&\, y = y \,\&\, \forall x(x = x)} & \&\text{-intro} \\
x = x \,\&\, \neg\forall x(x = x) & \vdash x = x \,\&\, \neg\forall x(x = x) & \text{ass} \\
x = x \,\&\, \neg\forall x(x = x) & \vdash \neg\forall x(x = x) & \&\text{-elim} \\
 & \vdash \forall x(x = x) & \forall\text{-intro} \\
x = x \,\&\, \neg\forall x(x = x) & \vdash \neg(x = x \,\&\, \neg\forall x(x = x)) & \text{contra} \\
\neg(x = x \,\&\, \neg\forall x(x = x)) & \vdash \neg(x = x \,\&\, \neg\forall x(x = x)) & \text{ass} \\
 & \vdash x = x \Rightarrow \forall x(x = x) & \text{rem} \\
 & \boldsymbol{\vdash x = x \,\&\, y = y \,\&\, \forall x(x = x)} & \\
 & \quad \boldsymbol{\&x = x \Rightarrow \forall x(x = x)} & \&\text{-intro}
\end{array}$$

Once we have shown that the translation of a PITFOL proof can be expanded to a proof, we can establish **equiconsistency** of the PITFOL calculus with the Hermes calculus as follows. Suppose that the PITFOL calculus would be inconsistent, i.e., every PITFOL sequent is derivable in the PITFOL calculus. In particular, the sequent

$$\vdash \neg(x = x)$$

would then be derivable in the PITFOL calculus. Translating the PITFOL proof of this sequent, we get a proof whose last sequent is

$$\vdash \mathcal{R}(\neg(x = x)) \,\&\, \mathcal{D}(\neg(x = x))$$

i.e.,

$$\vdash \neg(x = x) \,\&\, \forall x(x = x)$$

But then we would have a proof (in the Hermes calculus) of $\vdash \neg(x = x)$, and it is easy to see that this would imply that the Hermes calculus would be inconsistent.

Another corollary is that the PITFOL calculus is a **conservative extension** of the Hermes calculus. By this, we mean that when we have a PITFOL

proof of $\Gamma \vdash_\iota \alpha$, and Γ and α do not contain iota-terms, then there must exist a proof of $\Gamma \vdash \alpha$; in other words, in the PITFOL calculus, we cannot derive any "new" truths that were expressible in the original Hermes calculus. The proof is easy: the translations of $\Gamma \vdash_\iota \alpha$ are derivable, i.e.,

$$\begin{cases} \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

But since Γ and α do not contain iota-terms, $\mathcal{R}(\Gamma) \equiv \Gamma$ and $\mathcal{R}(\alpha) \equiv \alpha$. Hence, we have $\Gamma \vdash \alpha \,\&\, \mathcal{D}(\alpha)$, from which we easily obtain the desired sequent.

3.6.3 Lemmata

Remark that $\mathcal{R}(\alpha)$ and $\mathcal{D}(\alpha)$ never contain ι-terms. Further, it is easy to see that the reduction of a formula without ι-terms is the formula itself, and the definedness of a formula without ι-terms is a validity.

From these observations, we see that

- $\mathcal{R}(\mathcal{R}(\alpha)) \equiv \mathcal{R}(\alpha)$
- $\mathcal{D}(\mathcal{R}(\alpha))$ is a validity
- $\mathcal{R}(\mathcal{D}(\alpha)) \equiv \mathcal{D}(\alpha)$
- $\mathcal{D}(\mathcal{D}(\alpha))$ is a validity

Next, we establish that in $\mathcal{D}(\alpha_1 \,\&\, (\alpha_2 \,\&\, (\ldots \,\&\, (\alpha_{n-1} \,\&\, \alpha_n))))$, the order of association is not important, i.e., changing the association yields an equivalent formula:

$$\begin{aligned} \mathcal{D}(\alpha_1 \,\&\, (\alpha_2 \,\&\, \alpha_3)) &\equiv \mathcal{D}(\alpha_1) \,\&\, (\mathcal{R}(\alpha_1) \Rightarrow \mathcal{D}(\alpha_2 \,\&\, \alpha_3)) \\ &\equiv \mathcal{D}(\alpha_1) \,\&\, (\mathcal{R}(\alpha_1) \Rightarrow (\mathcal{D}(\alpha_2) \,\&\, (\mathcal{R}(\alpha_2) \Rightarrow \mathcal{D}(\alpha_3)))) \\ &\dashv\vdash \mathcal{D}(\alpha_1) \,\&\, (\mathcal{R}(\alpha_1) \Rightarrow \mathcal{D}(\alpha_2)) \,\&\, (\mathcal{R}(\alpha_1 \,\&\, \alpha_2) \Rightarrow \mathcal{D}(\alpha_3))) \\ &\equiv \mathcal{D}(\alpha_1 \,\&\, \alpha_2) \,\&\, (\mathcal{R}(\alpha_1 \,\&\, \alpha_2) \Rightarrow \mathcal{D}(\alpha_3)))) \\ &\equiv \mathcal{D}((\alpha_1 \,\&\, \alpha_2) \,\&\, \alpha_3) \end{aligned}$$

In general, one can prove that $\mathcal{D}(\alpha_1 \,\&\, \alpha_2 \,\&\, \ldots \,\&\, \alpha_n)$ is equivalent with

$$\begin{aligned} &\mathcal{D}(\alpha_1) \,\&\, (\mathcal{R}(\alpha_1) \Rightarrow \mathcal{D}(\alpha_2)) \\ &\quad \&\, (\mathcal{R}(\alpha_1 \,\&\, \alpha_2) \Rightarrow \mathcal{D}(\alpha_3)) \\ &\quad \&\, \ldots \\ &\quad \&\, (\mathcal{R}(\alpha_1 \,\&\, \alpha_2 \,\&\, \ldots \,\&\, \alpha_{n-1}) \Rightarrow \mathcal{D}(\alpha_n)) \end{aligned}$$

Using this result, we find that the translation of a sequent $\Sigma; \Gamma \vdash_\iota \alpha$ is equivalent with

$$\begin{cases} \vdash \mathcal{D}(\sigma_1) \\ \mathcal{R}(\sigma_1) \vdash \mathcal{D}(\sigma_2) \\ \mathcal{R}(\sigma_1), \mathcal{R}(\sigma_2) \vdash \mathcal{D}(\sigma_3) \\ \vdots \\ \mathcal{R}(\sigma_1), \mathcal{R}(\sigma_2), \ldots, \mathcal{R}(\sigma_{n-1}) \vdash \mathcal{D}(\sigma_n) \\ \mathcal{R}(\Sigma) \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\Sigma), \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

Finally, we investigate the definedness of $\alpha \vee \beta$, $\alpha \Rightarrow \beta$ and $\exists x(\alpha)$.

$$\begin{aligned} \mathcal{D}(\alpha \vee \beta) &\equiv \mathcal{D}(\neg(\neg\alpha \,\&\, \neg\beta)) \\ &\equiv \mathcal{D}(\neg\alpha \,\&\, \neg\beta) \\ &\equiv \mathcal{D}(\neg\alpha) \,\&\, (\mathcal{R}(\neg\alpha) \Rightarrow \mathcal{D}(\neg\beta)) \\ &\equiv \mathcal{D}(\alpha) \,\&\, ((\neg\mathcal{R}(\alpha)) \Rightarrow \mathcal{D}(\beta)) \end{aligned}$$

$$\begin{aligned} \mathcal{D}(\alpha \Rightarrow \beta) &\equiv \mathcal{D}(\neg(\alpha \,\&\, \neg\beta)) \\ &\equiv \mathcal{D}(\alpha \,\&\, \neg\beta) \\ &\equiv \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\neg\beta)) \\ &\equiv \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta)) \end{aligned}$$

$$\begin{aligned} \mathcal{D}(\alpha \Leftrightarrow \beta) &\equiv \mathcal{D}((\alpha \Rightarrow \beta) \,\&\, (\beta \Rightarrow \alpha)) \\ &\equiv \mathcal{D}(\alpha \Rightarrow \beta) \,\&\, (\mathcal{R}(\alpha \Rightarrow \beta)) \Rightarrow \mathcal{D}(\beta \Rightarrow \alpha)) \\ &\equiv \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta)) \,\&\, (\mathcal{R}(\alpha \Rightarrow \beta) \Rightarrow (\mathcal{D}(\beta) \,\&\, (\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\alpha)))) \\ &\dashv\vdash \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta)) \,\&\, (\mathcal{R}(\alpha \Rightarrow \beta) \Rightarrow \mathcal{D}(\beta)) \\ &\dashv\vdash \mathcal{D}(\alpha) \,\&\, \mathcal{D}(\beta) \end{aligned}$$

$$\mathcal{D}(\exists x(\alpha)) \equiv \mathcal{D}(\neg\forall x(\neg\alpha)) \equiv \mathcal{D}(\forall x(\neg\alpha)) \equiv \forall x(\mathcal{D}(\neg\alpha)) \equiv \forall x \mathcal{D}(\alpha)$$

Note that $\mathcal{D}(\alpha \Rightarrow \beta) \equiv \mathcal{D}(\alpha \,\&\, \beta)$ and $\mathcal{D}(\exists x(\alpha)) \equiv \mathcal{D}(\forall x(\alpha))$.

3.6.4 Translation of the proposition rules

The following lemma shows a connection between $\mathbf{\Delta}$ and $\mathcal{D}$.

Lemma 9 *If $\mathbf{\Delta}(\alpha)$ is not $\top$, then $\mathcal{R}(\mathbf{\Delta}(\alpha))$ is equivalent to $\mathcal{D}(\alpha)$. If $\mathbf{\Delta}(\alpha)$ is $\top$, then $\mathcal{D}(\alpha)$ is a validity.*

Proof.

We prove this by structural induction on α.

- If α is atomic, it is of the form $\alpha \equiv p(t_1, t_2, \ldots, t_n)$, where we treat the case $\alpha \equiv t_1 = t_2$ analogously.
  First suppose α contains some ι-terms. Then $\mathbf{\Delta}(\alpha) \equiv \psi_1 \& \psi_2 \& \ldots \& \psi_m$ with the usual notations, i.e., $\psi_1, \psi_2, \ldots, \psi_m$ are the domain formulae of the top-level iota terms of α. Hence $\mathcal{R}(\mathbf{\Delta}(\alpha))$ is

  $$\mathcal{R}(\psi_1 \,\&\, \psi_2 \,\&\, \ldots \,\&\, \psi_m) \equiv \mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) \,\&\, \ldots \,\&\, \mathcal{R}(\psi_m) \equiv \mathcal{D}(\alpha)$$

  If there are no ι-terms present, then $\mathbf{\Delta}(\alpha) \equiv \top$ and we have to show that $\mathcal{D}(\alpha)$ is a validity, which is the case since $\mathcal{D}(\alpha) \equiv \forall x(x = x)$.

- Suppose $\alpha \equiv \neg\beta$. If $\mathbf{\Delta}(\beta)$ is not $\top$, then by induction, we know that $\mathcal{R}(\mathbf{\Delta}(\beta))$ is equivalent to $\mathcal{D}(\beta)$. Hence

  $$\mathcal{R}(\mathbf{\Delta}(\alpha)) \equiv \mathcal{R}(\mathbf{\Delta}(\beta)) \dashv\vdash \mathcal{D}(\beta) \equiv \mathcal{D}(\alpha)$$

  If $\mathbf{\Delta}(\beta)$ is $\top$, then $\mathcal{D}(\beta)$ is a validity. Then $\mathcal{D}(\alpha)$ is a validity too, since $\mathcal{D}(\alpha) \equiv \mathcal{D}(\beta)$.

- For formulae of the form $\alpha \,\&\, \beta$, we have four cases.

  - If both $\mathbf{\Delta}(\alpha)$ and $\mathbf{\Delta}(\beta)$ are not $\top$, then $\mathcal{R}(\mathbf{\Delta}(\alpha))$ is equivalent to $\mathcal{D}(\alpha)$ and likewise for β. Then

    $$\begin{aligned}\mathcal{R}(\mathbf{\Delta}(\alpha \,\&\, \beta)) &\equiv \mathcal{R}(\mathbf{\Delta}(\alpha) \,\&\, (\alpha \Rightarrow \mathbf{\Delta}(\beta))) \\ &\equiv \mathcal{R}(\mathbf{\Delta}(\alpha)) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{R}(\mathbf{\Delta}(\beta))) \\ &\dashv\vdash \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta)) \\ &\equiv \mathcal{D}(\alpha \,\&\, \beta)\end{aligned}$$

  - If $\mathbf{\Delta}(\alpha)$ is not $\top$ but $\mathbf{\Delta}(\beta)$ is, then $\mathcal{R}(\mathbf{\Delta}(\alpha))$ is equivalent to $\mathcal{D}(\alpha)$ and $\mathcal{D}(\beta)$ is a validity. Then

    $$\mathcal{R}(\mathbf{\Delta}(\alpha \,\&\, \beta)) \equiv \mathcal{R}(\mathbf{\Delta}(\alpha)) \dashv\vdash \mathcal{D}(\alpha) \dashv\vdash \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta))$$

    which is $\mathcal{D}(\alpha \,\&\, \beta)$.

– If $\mathbf{\Delta}(\alpha)$ is $\top$ and $\mathbf{\Delta}(\beta)$ is not, then $\mathcal{D}(\alpha)$ is a validity and $\mathcal{R}(\mathbf{\Delta}(\beta))$ is equivalent to $\mathcal{D}(\beta)$. Hence

$$\begin{aligned}\mathcal{R}(\mathbf{\Delta}(\alpha \,\&\, \beta)) &\equiv \mathcal{R}(\alpha \Rightarrow \mathbf{\Delta}(\beta)) \\ &\dashv\vdash \mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta) \\ &\dashv\vdash \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta))\end{aligned}$$

– If both $\mathbf{\Delta}(\alpha)$ and $\mathbf{\Delta}(\beta)$ are $\top$, then we have to show that $\mathcal{D}(\alpha \,\&\, \beta)$ is a validity. By induction, $\mathcal{D}(\alpha)$ and $\mathcal{D}(\beta)$ are validities, from which we get the desired result.

- Finally, we consider formulae of the form $\forall x(\alpha)$. If $\mathbf{\Delta}(\alpha)$ is not $\top$, then by induction we know that $\mathcal{R}(\mathbf{\Delta}(\alpha))$ is equivalent to $\mathcal{D}(\alpha)$. Now

$$\mathcal{R}(\mathbf{\Delta}(\forall x(\alpha))) \equiv \forall x(\mathcal{R}(\mathbf{\Delta}(\alpha)))$$

Using the induction hypothesis, we know that this is equivalent to $\forall x(\mathcal{D}(\alpha))$, which is $\mathcal{D}(\forall x(\alpha))$.
If $\mathbf{\Delta}(\alpha)$ is $\top$, we have to show that $\mathcal{D}(\forall x(\alpha))$ is a validity. By induction, $\mathcal{D}(\alpha)$ is a validity, from which we easily get the desired result.

□

Lemma 10 *If $\mathbf{\Delta}(\alpha)$ is not $\top$, and the translations of the uniqueness conditions of α are derived, then $\mathcal{D}(\mathbf{\Delta}(\alpha))$ is a validity. If $\mathbf{\Delta}(\alpha)$ is $\top$, then $\mathcal{D}(\alpha)$ is a validity.*

Proof.
We prove this by structural induction on α.

- If α is atomic, then either α does not contain ι-terms, and then $\mathbf{\Delta}(\alpha) \equiv \top$, and $\mathcal{D}(\alpha) \equiv \forall x(x = x)$, a validity, or else

$$\mathcal{D}(\mathbf{\Delta}(\alpha)) \equiv \mathcal{D}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m)$$

where as usual, we enumerated the top-level ι-terms of α as $TLI(\alpha) \equiv \iota x_{1\psi_1}(\varphi_1), \ldots, \iota x_{m\psi_m}(\varphi_m)$. From the translation of the uniqueness condition for $\iota x_{i\psi_i}(\varphi_i)$, we get $\vdash \mathcal{D}(\psi_i)$; hence we easily conclude $\vdash \mathcal{D}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m)$.

- The other cases are straightforward; only for formulae of the form $\alpha \& \beta$, the proof is somewhat more convoluted:

– If both $\mathbf{\Delta}(\alpha)$ and $\mathbf{\Delta}(\beta)$ are not $\top$, then both $\mathcal{D}(\mathbf{\Delta}(\alpha))$ and $\mathcal{D}(\mathbf{\Delta}(\beta))$ are validities and we have

$$\begin{aligned}\mathcal{D}(\mathbf{\Delta}(\alpha \,\&\, \beta)) &\equiv \mathcal{D}(\mathbf{\Delta}(\alpha) \,\&\, (\alpha \Rightarrow \mathbf{\Delta}(\beta))) \\ &\equiv \mathcal{D}(\mathbf{\Delta}(\alpha)) \,\&\, (\mathcal{R}(\mathbf{\Delta}(\alpha)) \Rightarrow \mathcal{D}(\alpha \Rightarrow \mathbf{\Delta}(\beta))) \\ &\equiv \mathcal{D}(\mathbf{\Delta}(\alpha)) \,\&\, (\mathcal{R}(\mathbf{\Delta}(\alpha)) \Rightarrow (\mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\mathbf{\Delta}(\beta))))) \\ &\dashv\vdash \mathcal{D}(\alpha) \Rightarrow \mathcal{D}(\alpha)\end{aligned}$$

using lemma 9, which clearly yields a validity.

– If $\mathbf{\Delta}(\alpha)$ is not $\top$ but $\mathbf{\Delta}(\beta)$ is, then $\mathcal{D}(\mathbf{\Delta}(\alpha \,\&\, \beta)) \equiv \mathcal{D}(\alpha)$, which is a validity using the induction hypothesis.

– If $\mathbf{\Delta}(\alpha)$ is $\top$ and $\mathbf{\Delta}(\beta)$ is not, then

$$\mathcal{D}(\mathbf{\Delta}(\alpha \,\&\, \beta)) \equiv \mathcal{D}(\alpha \Rightarrow \mathbf{\Delta}(\beta)) \equiv \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\mathbf{\Delta}(\beta)))$$

is a validity, since by induction, both $\mathcal{D}(\alpha)$ and $\mathcal{D}(\mathbf{\Delta}(\beta))$ are validities.

– If $\mathbf{\Delta}(\alpha)$ and $\mathbf{\Delta}(\beta)$ are both $\top$, then $\mathbf{\Delta}(\alpha \,\&\, \beta)$ is also $\top$ and we have to show that $\mathcal{D}(\alpha \,\&\, \beta)$ is a validity. Since by induction, both $\mathcal{D}(\alpha)$ and $\mathcal{D}(\beta)$ are validities,

$$\mathcal{D}(\alpha \,\&\, \beta) \equiv \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta))$$

is also a validity.

□

Structural rules

In the definition of PITFOL sequent, we indicated that one is free to add or remove duplicates of formulae in antecedent and context. This amounts to silently applying the following structural rules:

$$\boxed{\text{struct1}}\quad \frac{\Sigma; \Gamma_1, \alpha, \Gamma_2 \vdash_\iota \beta}{\Sigma; \Gamma_1, \alpha, \alpha, \Gamma_2 \vdash_\iota \beta} \qquad \boxed{\text{struct2}}\quad \frac{\Sigma; \Gamma_1, \alpha, \alpha, \Gamma_2 \vdash_\iota \beta}{\Sigma; \Gamma_1, \alpha, \Gamma_2 \vdash_\iota \beta} \qquad \boxed{\text{struct3}}\quad \frac{\Sigma; \Gamma_1, \alpha, \beta, \Gamma_2 \vdash_\iota \gamma}{\Sigma; \Gamma_1, \beta, \alpha, \Gamma_2 \vdash_\iota \gamma}$$

$$\boxed{\text{struct4}}\quad \frac{\Sigma_1, \sigma, \Sigma_2, \sigma, \Sigma_3; \Gamma \vdash_\iota \alpha}{\Sigma_1, \sigma, \Sigma_2, \Sigma_3; \Gamma \vdash_\iota \alpha} \qquad \boxed{\text{struct5}}\quad \frac{\Sigma_1, \sigma, \Sigma_2, \Sigma_3; \Gamma \vdash_\iota \alpha}{\Sigma_1, \sigma, \Sigma_2, \sigma, \Sigma_3; \Gamma \vdash_\iota \alpha}$$

where Σ, Σ_1, Σ_2, Σ_3, Γ, Γ_1 and Γ_2 are possibly empty lists of formulae and α, β, γ and σ are formulae. For all these rules, it is easy to see that one can derive the translations of the conclusion from the translation of the premise.

Assumption introduction

If $\mathbf{\Delta}(\alpha)$ is not $\top$, then the translation of the premise is

$$\begin{cases} \vdash_\iota \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \ldots \,\&\, \sigma_n) \\ \mathcal{R}(\Sigma) \vdash_\iota \mathcal{R}(\mathbf{\Delta}(\alpha)) \,\&\, \mathcal{D}(\mathbf{\Delta}(\alpha)) \end{cases}$$

From the lemmata above, we can conclude that the last sequent is equivalent to $\mathcal{R}(\Sigma) \vdash \mathcal{D}(\alpha)$. From this, we have to deduce the translation of the conclusion:

$$\begin{cases} \vdash_\iota \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \ldots \,\&\, \sigma_n) \\ \mathcal{R}(\Sigma) \vdash_\iota \mathcal{D}(\alpha) \\ \mathcal{R}(\Sigma)\,, \mathcal{R}(\alpha) \vdash_\iota \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

The first sequent we already have; the second we already have obtained, and the third can be easily deduced from it.

If $\mathbf{\Delta}(\alpha) \equiv \top$, then $\mathcal{D}(\alpha)$ is a validity and we have to derive

$$\begin{cases} \vdash_\iota \mathcal{D}(\alpha) \\ \mathcal{R}(\alpha) \vdash_\iota \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

which is easy.

Note that we could have used

$$\frac{\Sigma; \vdash_\iota \mathcal{D}(\alpha)}{\Sigma; \alpha \vdash_\iota \alpha}$$

as assumption introduction rule—in that case, the last sequent of the translation of the premise would have been

$$\mathcal{R}(\Sigma) \vdash \mathcal{R}(\mathcal{D}(\alpha)) \,\&\, \mathcal{D}(\mathcal{D}(\alpha))$$

and using §3.6.3, we can deduce $\mathcal{R}(\Sigma) \vdash \mathcal{D}(\alpha)$, which is the same sequent we got using the variant of the rule with $\Sigma; \vdash_\iota \mathbf{\Delta}(\alpha)$ as premise.

The only reason we introduced $\mathbf{\Delta}$ is that $\mathbf{\Delta}(\alpha)$ is usually a shorter formula than $\mathcal{D}(\alpha)$, since in the former we are still allowed to use ι-terms, while in the latter we must reduce them fully.

&-introduction

The translation of the premises gives us the sequents

$$\begin{cases} \vdash \mathcal{D}(\sigma_{1,1}) \\ \mathcal{R}(\sigma_{1,1}) \vdash \mathcal{D}(\sigma_{1,2}) \\ \vdots \\ \mathcal{R}(\Sigma_1) \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\Sigma_1), \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

and

$$\begin{cases} \vdash \mathcal{D}(\sigma_{2,1}) \\ \mathcal{R}(\sigma_{2,1}) \vdash \mathcal{D}(\sigma_{2,2}) \\ \vdots \\ \mathcal{R}(\Sigma_2) \vdash \mathcal{D}(\delta_1) \,\&\, \mathcal{D}(\delta_2) \,\&\, \ldots \,\&\, \mathcal{D}(\delta_l) \\ \mathcal{R}(\Sigma_2), \mathcal{R}(\Delta) \vdash \mathcal{R}(\beta) \,\&\, \mathcal{D}(\beta) \end{cases}$$

where we denote

$$\Sigma_1 \equiv \sigma_{1,1}, \sigma_{1,2}, \ldots, \sigma_{1,n_1} \qquad \Gamma \equiv \gamma_1, \gamma_2, \ldots, \gamma_m$$
$$\Sigma_2 \equiv \sigma_{2,1}, \sigma_{2,2}, \ldots, \sigma_{2,n_2} \qquad \Delta \equiv \delta_1, \delta_2, \ldots, \delta_l.$$

First, we have to derive the sequents

$$\vdash \mathcal{D}(\sigma_{1,1}) \tag{1}$$
$$\mathcal{R}(\sigma_{1,1}) \vdash \mathcal{D}(\sigma_{1,2})$$
$$\mathcal{R}(\sigma_{1,1}), \mathcal{R}(\sigma_{1,2}) \vdash \mathcal{D}(\sigma_{1,3})$$
$$\vdots$$
$$\mathcal{R}(\sigma_{1,1}), \mathcal{R}(\sigma_{1,2}), \ldots, \mathcal{R}(\sigma_{1,n_1-1}) \vdash \mathcal{D}(\sigma_{1,n_1}) \tag{2}$$
$$\mathcal{R}(\Sigma_1) \vdash \mathcal{D}(\sigma_{2,1}) \tag{3}$$
$$\mathcal{R}(\Sigma_1), \mathcal{R}(\sigma_{2,1}) \vdash \mathcal{D}(\sigma_{2,2})$$
$$\mathcal{R}(\Sigma_1), \mathcal{R}(\sigma_{2,1}), \mathcal{R}(\sigma_{2,1}) \vdash \mathcal{D}(\sigma_{2,3})$$
$$\vdots$$
$$\mathcal{R}(\Sigma_1), \mathcal{R}(\sigma_{2,1}), \ldots, \mathcal{R}(\sigma_{2,n_2-1}) \vdash \mathcal{D}(\sigma_{2,n_2}) \tag{4}$$

We already have the first group of sequents (1)–(2) from the translation of the first premise. For the second group (3)–(4), we easily obtain the

required sequent from the corresponding sequent of the translation of the second premise.

Next, we handle the one but last sequent of the translation of the conclusion:

$$\mathcal{R}(\Sigma_1), \mathcal{R}(\Sigma_2) \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \mathcal{D}(\gamma_m) \,\&\, \mathcal{D}(\delta_1) \,\&\, \mathcal{D}(\delta_2) \,\&\, \ldots \mathcal{D}(\delta_l)$$

This sequent is easy to derive by applying the &-introduction rule to the one but last sequents of the translation of both premises.

Finally, we have to deduce

$$\mathcal{R}(\Sigma_1), \mathcal{R}(\Sigma_2), \mathcal{R}(\Gamma), \mathcal{R}(\Delta) \vdash \mathcal{R}(\alpha \,\&\, \beta) \,\&\, \mathcal{D}(\alpha \,\&\, \beta)$$

i.e.,

$$\mathcal{R}(\Sigma_1), \mathcal{R}(\Sigma_2), \mathcal{R}(\Gamma), \mathcal{R}(\Delta) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{R}(\beta) \,\&\, \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta))$$

which is easily obtained from the last sequents of the translation of both premises.

For the other rules of this section, we will do the proofs for the case that the context Σ is empty. Expanding the proof to non-empty contexts is analogous to the proofs already given.

&-elimination

We will handle both &-elimination rules at once. Translating the premise, we have

$$\begin{cases} \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha \,\&\, \beta) \,\&\, \mathcal{D}(\alpha \,\&\, \beta) \end{cases}$$

This gives us immediately the first sequent of the translation of the conclusion. The second sequent is

$$\mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{R}(\beta) \,\&\, \mathcal{D}(\alpha) \,\&\, (\mathcal{R}(\alpha) \Rightarrow \mathcal{D}(\beta))$$

from which it is easy to deduce the second sequents of the translations we need:

$$\mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha)$$
$$\mathcal{R}(\Gamma) \vdash \mathcal{R}(\beta) \,\&\, \mathcal{D}(\beta)$$

Removal

Translation of the premises yields

$$\begin{cases} \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \,\&\, \mathcal{D}(\alpha) \\ \mathcal{R}(\Gamma), \mathcal{R}(\alpha) \vdash \mathcal{R}(\beta) \,\&\, \mathcal{D}(\beta) \\ \mathcal{R}(\Gamma), \mathcal{R}(\neg\alpha) \vdash \mathcal{R}(\beta) \,\&\, \mathcal{D}(\beta) \end{cases}$$

We only have three sequents, since the first sequents of the translations are identical (using the fact that $\mathcal{D}(\alpha) \equiv \mathcal{D}(\neg\alpha)$).

Using the property $\mathcal{R}(\neg\alpha) \equiv \neg\mathcal{R}(\alpha)$, we see that we can apply the removal rule to the remaining two sequents, yielding the required

$$\mathcal{R}(\Gamma) \vdash \mathcal{R}(\beta) \,\&\, \mathcal{D}(\beta)$$

Contradiction

The premises are in translation:

$$\begin{cases} \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

$$\begin{cases} \vdash \mathcal{D}(\delta_1) \,\&\, \mathcal{D}(\delta_2) \,\&\, \ldots \,\&\, \mathcal{D}(\delta_m) \\ \mathcal{R}(\Delta) \vdash \neg\mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

Using &-elimination twice, we get

$$\begin{aligned} \mathcal{R}(\Gamma) &\vdash \mathcal{R}(\alpha) \\ \mathcal{R}(\Delta) &\vdash \neg\mathcal{R}(\alpha) \end{aligned}$$

Applying the contradiction rule yields

$$\mathcal{R}(\Gamma), \mathcal{R}(\Delta) \vdash \mathcal{R}(\beta) \,\&\, \mathcal{D}(\beta)$$

which is the second sequent of the translation of the conclusion; obtaining the first sequent is trivial.

3.6.5 Translation of the predicate rules

∀-introduction

The translation of the premise is

$$\begin{cases} \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \ldots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

If we keep the first sequent and apply the ∀-introduction rule of the Hermes calculus, we get the translation of the conclusion.

∀-elimination

The proof is analogous to the ∀-introduction rule.

3.6.6 Translation of the other rules

Before we can handle these rules, we have to build up some theoretical machinery first. Basically, our approach is an adaptation of [Hilbert & Bernays 1968].

We start by proving a more general version of lemma 8.

Lemma 11 *If t is a term of the* PITFOL *calculus, V is the set of free variables of t and the substitution $[t/x]\alpha$ is defined, then*

$$\mathcal{R}([t/x]\alpha) \dashv\vdash \mathcal{R}([t/x]\mathcal{R}_V(\alpha))$$

Proof.

We prove this by induction on the complexity of α.

We first note that if x is not a free variable of α, the lemma is trivial. Hence we will suppose that x is a free variable of α.

- If α has complexity zero, i.e., α is atomic and does not contain any ι-terms, then, as we showed in §3.6.3, $\mathcal{R}_V(\alpha) \equiv \alpha$ and the lemma is trivially proved.

- When α is atomic and contains ι-terms, i.e., $\alpha \equiv p(\tau_1, \tau_2, \ldots, \tau_n)$, we proceed as follows (we treat the case $\alpha \equiv \tau_1 = \tau_2$ analogously).
  Enumerate the top-level ι-terms of α as

  $$TLI(\alpha) \equiv \iota x_{1\psi_1}(\varphi_1), \iota x_{2\psi_2}(\varphi_2), \ldots, \iota x_{m\psi_m}(\varphi_m).$$

  Enumerate the top-level ι-terms of t as

  $$TLI(t) \equiv \iota y_{1\zeta_1}(\chi_1), \iota y_{2\zeta_2}(\chi_2), \ldots, \iota y_{l\zeta_l}(\chi_l).$$

  Call k the number of top-level free occurrences in α of the variable symbol x (i.e., those occurrences not inside a $\forall x$ or inside a definiens or domain formula of a ι-term). Construct the terms t_1, t_2, …, t_k by replacing the top-level ι-terms of t with the variable symbols $v_1^i, v_2^i, \ldots, v_l^i$, that is,
  • t_1 is obtained from t by replacing its top-level ι-terms with $v_1^1, v_2^1, \ldots, v_l^1$
  • …

• t_k is obtained from t by replacing its top-level ι-terms with $v_1^k, v_2^k, \ldots, v_l^k$.
Finally, note the definiens of $[t/x]\iota x_{i\psi_i}(\varphi_i)$ as φ_i'.
The top-level ι-terms of $[t/x]\alpha$ originate from two sources:

- A top-level ι-term $\iota x_{i\psi_i}(\varphi_i)$ of α gives rise to a single ι-term $[t/x]\iota x_{i\psi_i}(\varphi_i)$
- A top-level variable symbol x gives rise to l top-level ι-terms, namely $TLI(t)$.

Then, $\mathcal{R}([t/x]\alpha)$ can be written as

$$\begin{aligned}\exists w_1\exists w_2\ldots\exists w_m\exists v_1^1\exists v_2^1\ldots\exists v_l^1\exists v_1^2\exists v_2^2\ldots\exists v_l^2\ldots\exists v_1^k\exists v_2^k\ldots\exists v_l^k\Big(&\\ \mathcal{R}([w_1/x_1]\varphi_1')\,\&\,\mathcal{R}([w_2/x_2]\varphi_2')\,\&\ldots\&\,\mathcal{R}([w_m/x_m]\varphi_m')&\\ \&\mathcal{R}([v_1^1/y_1]\chi_1)\,\&\,\mathcal{R}([v_2^1/y_2]\chi_2)\,\&\ldots\&\,\mathcal{R}([v_l^1/y_l]\chi_l)&\\ \&\ldots\mathcal{R}\big([v_1^k/y_1]\chi_1\big)\,\&\,\mathcal{R}\big([v_2^k/y_2]\chi_2\big)\,\&\ldots\&\,\mathcal{R}\big([v_l^k/y_l]\chi_l\big)\,\&\,q'\Big)&\end{aligned} \tag{3.2}$$

where q' is obtained from α by replacing its top-level ι-terms with w_1, w_2, …, w_m and its top-level occurrences of the variable symbol x with t_1, t_2, …, t_k.
As usual, the w's and v's are all different and do not occur in $[t/x]\alpha$, i.e., they do not occur in α and t.

Next, we investigate $\mathcal{R}_V(\alpha)$, which can be written as

$$\exists u_1\exists u_2\ldots\exists u_m(\mathcal{R}_V([u_1/x_1]\varphi_1)\,\&\,\mathcal{R}_V([u_2/x_2]\varphi_2)\,\&\ldots\&\,\mathcal{R}_V([u_m/x_m]\varphi_m)\,\&\,q)$$

where q is obtained from α by replacing its top-level ι-terms with u_1, u_2, …, u_m and the u's are all different, do not occur in α and are not free variables of t.
Hence, $\mathcal{R}([t/x]\mathcal{R}_V(\alpha))$ is

$$\begin{aligned}\exists u_1\exists u_2\ldots\exists u_m(\mathcal{R}([t/x]\mathcal{R}_V([u_1/x_1]\varphi_1))\,\&\ldots&\\ \&\,\mathcal{R}([t/x]\mathcal{R}_V([u_m/x_m]\varphi_m))\,\&\,\mathcal{R}([t/x]q))&\end{aligned}$$

Using induction on the formulae $[u_1/x_1]\varphi_1$, $[u_2/x_2]\varphi_2$, …, $[u_m/x_m]\varphi_m$, this is equivalent with

$$\exists u_1\exists u_2\ldots\exists u_m(\mathcal{R}([t/x][u_1/x_1]\varphi_1)\,\&\cdots\&\,\mathcal{R}([t/x][u_m/x_m]\varphi_m)\,\&\,\mathcal{R}([t/x]q))$$

We will now establish that

$$\mathcal{R}([t/x][u_i/x_i]\varphi_i)\dashv\vdash\mathcal{R}([u_i/x_i]\varphi_i')$$

– If $x \equiv x_i$, then $[t/x]\iota x_{\psi_i}(\varphi_i) \equiv \iota x_{\psi_i}(\varphi_i)$ or $[t/x]\iota x_{\psi_i}(\varphi_i) \equiv \iota x_{\mathbf{\Delta}(t)\&[t/x]\psi_i}(\varphi_i)$; hence $\varphi'_i \equiv \varphi_i$ and we have to prove that

$$\mathcal{R}([t/x][u_i/x]\varphi_i) \dashv\vdash \mathcal{R}([u_i/x]\varphi_i)$$

which is easy, since $[t/x][u_i/x]\varphi_i \equiv [u_i/x]\varphi_i$ because $u_i \not\equiv x$.

– If x is not a free variable of φ_i, then analogously, $\varphi'_i \equiv \varphi_i$. We must again prove that

$$\mathcal{R}([t/x][u_i/x_i]\varphi_i) \dashv\vdash \mathcal{R}([u_i/x_i]\varphi_i)$$

But since $u_i \not\equiv x$ and x is not a free variable of φ_i, we also have that x is not a free variable of $[u_i/x_i]\varphi_i$. Hence $[t/x][u_i/x_i]\varphi_i \equiv [u_i/x_i]\varphi_i$.

– If x is a free variable of φ_i and $x \not\equiv x_i$, then $[t/x]\iota x_{i\psi_i}(\varphi_i) \equiv \iota x_{i\,\mathbf{\Delta}(t)\&[t/x]\psi_i}([t/x]\varphi_i)$ (We take the substitution to be defined, hence x_i is not free in t.) So in this case, $\varphi'_i \equiv [t/x]\varphi_i$ and we have to prove that

$$\mathcal{R}([t/x][u_i/x_i]\varphi_i) \dashv\vdash \mathcal{R}([u_i/x_i][t/x]\varphi_i)$$

Since we supposed the substitution to be defined, x_i is not a free variable of t; we also have that u_i does not occur in α and hence $u_i \not\equiv x$. It is now easy to show that in that case, we may change the order of the substitutions without affecting the result.

Using this result, we conclude that $\mathcal{R}([t/x]\mathcal{R}_V(\alpha))$ is equivalent with

$$\exists u_1 \exists u_2 \ldots \exists u_m \big(\mathcal{R}([u_1/x_1]\varphi'_1) \,\&\, \mathcal{R}([u_2/x_2]\varphi'_2) \,\&\, \ldots \,\&\, \mathcal{R}([u_m/x_m]\varphi'_m) \,\&\, \mathcal{R}([t/x]q)\big)$$

We now investigate the structure of $\mathcal{R}([t/x]q)$. By construction, q does not contain any ι-terms; the only possible ι-terms in $[t/x]q$ are in copies of t (one for each free occurrence of x in q)—that is, if t contains any ι-terms at all. Hence, if we construct the terms t'_i by replacing the top-level ι-terms of t with the variable symbols $v'^i_1, v'^i_2, \ldots, v'^i_l$, we can write $\mathcal{R}([t/x]q)$ as

$$\begin{aligned}&\exists v'^1_1 \exists v'^1_2 \ldots \exists v'^1_l \exists v'^2_1 \exists v'^2_2 \ldots \exists v'^2_l \ldots \exists v'^k_1 \exists v'^k_2 \ldots \exists v'^k_l \Big(\\ &\quad \&\mathcal{R}\big(\big[v'^1_1/y_1\big]\chi_1\big) \,\&\, \mathcal{R}\big(\big[v'^1_2/y_2\big]\chi_2\big) \,\&\, \ldots \,\&\, \mathcal{R}\big(\big[v'^1_l/y_l\big]\chi_l\big) \\ &\quad \&\cdots\&\, \mathcal{R}\Big(\Big[v'^k_1/y_1\Big]\chi_1\Big) \,\&\, \mathcal{R}\Big(\Big[v'^k_2/y_2\Big]\chi_2\Big) \,\&\, \ldots \,\&\, \mathcal{R}\Big(\Big[v'^k_l/y_l\Big]\chi_l\Big) \,\&\, q''\Big)\end{aligned}$$

where we obtain q'' by replacing all free occurrences of x with $t'_1, t'_2, \ldots, t'_k$.

Combining this and moving the existential quantifiers $\exists v'^j_i$ to the front, we notice that $\mathcal{R}([t/x]\mathcal{R}(\alpha))$ is equivalent to a formula which has the same structure as (3.2) except that all u's and v''s are replaced with w's and v's.

But lemma 8 yields that reductions of the same formula ($[t/x]\alpha$ in this case) with different variables are equivalent.

- The cases where α is not atomic are straightforward.

□

If t does not contain any ι-terms, then this lemma simplifies to

$$\mathcal{R}([t/x]\alpha) \dashv\vdash [t/x]\mathcal{R}_V(\alpha)$$

For the sequel, we will need the following property of the Hermes calculus:

Lemma 12 *For each formula α and β,*

$$\exists!x(\alpha), \forall x(\alpha \Rightarrow \beta) \vdash \exists x(\alpha \,\&\, \beta)$$

Proof.

$$\begin{array}{rr}
\forall x\neg(\alpha \,\&\, \beta) \vdash \forall x\neg(\alpha \,\&\, \beta) & \text{ass} \\
\forall x\neg(\alpha \,\&\, \beta) \vdash \neg(\alpha \,\&\, \beta) & \forall\text{-elim} \\
\alpha \vdash \alpha & \text{ass} \\
\beta \vdash \beta & \text{ass} \\
\alpha, \beta \vdash \alpha \,\&\, \beta & \&\text{-intro} \\
\forall x\neg(\alpha \,\&\, \beta), \alpha, \beta \vdash \neg\alpha & \text{contra} \\
\forall x(\alpha \Rightarrow \beta) \vdash \forall x(\alpha \Rightarrow \beta) & \text{ass} \\
\forall x(\alpha \Rightarrow \beta) \vdash \neg(\alpha \,\&\, \neg\beta) & \forall\text{-elim} \\
\neg\beta \vdash \neg\beta & \text{ass} \\
\alpha, \neg\beta \vdash \alpha \,\&\, \neg\beta & \&\text{-intro} \\
\forall x(\alpha \Rightarrow \beta), \alpha, \neg\beta \vdash \neg\alpha & \text{contra} \\
\forall x\neg(\alpha \,\&\, \beta), \forall x(\alpha \Rightarrow \beta), \alpha \vdash \neg\alpha & \text{rem} \\
\neg\alpha \vdash \neg\alpha & \text{ass} \\
\forall x\neg(\alpha \,\&\, \beta), \forall x(\alpha \Rightarrow \beta) \vdash \neg\alpha & \text{rem} \\
\forall x\neg(\alpha \,\&\, \beta), \forall x(\alpha \Rightarrow \beta) \vdash \forall x(\neg\alpha) & \forall\text{-intro} \\
\exists!x(\alpha) \vdash \exists!x(\alpha) & \text{ass}
\end{array}$$

$$\begin{array}{rr}
\exists!x(\alpha) \vdash \neg\forall x(\neg\alpha) & \&\text{-elim} \\
\exists!x(\alpha), \forall x\neg(\alpha \,\&\, \beta), \forall x(\alpha \Rightarrow \beta) \vdash \neg\forall x\neg(\alpha \,\&\, \beta) & \text{contra} \\
\neg\forall x\neg(\alpha \,\&\, \beta) \vdash \neg\forall x\neg(\alpha \,\&\, \beta) & \text{ass} \\
\exists!x(\alpha), \forall x(\alpha \Rightarrow \beta) \vdash \neg\forall x\neg(\alpha \,\&\, \beta) & \text{rem}
\end{array}$$

□

Note that one can also prove $\exists!x(\alpha), \exists x(\alpha \,\&\, \beta) \vdash \forall x(\alpha \Rightarrow \beta)$.

Lemma 13 *From the translation of $\psi \vdash_\iota \exists!x(\varphi)$ we can deduce*

$$\mathcal{R}(\psi) \vdash (\mathcal{R}([u_1/x]\varphi) \,\&\, \mathcal{R}([u_2/x]\varphi)) \Leftrightarrow (\mathcal{R}([u_1/x]\varphi) \,\&\, u_1 = u_2)$$

if u_1 and u_2 do not occur free in φ.

Proof.

• First, we will prove the $\Rightarrow$ direction.

The translation of the uniqueness condition is

$$\begin{cases} & \vdash \ \mathcal{D}(\psi) \\ \mathcal{R}(\psi) & \vdash \ \mathcal{R}(\exists!x(\varphi)) \,\&\, \mathcal{D}(\exists!x(\varphi)) \end{cases}$$

from which we can deduce

$$\mathcal{R}(\psi) \vdash \forall x\forall y((\mathcal{R}(\varphi) \,\&\, \mathcal{R}([y/x]\varphi)) \Rightarrow x = y)$$

with y not occurring in φ. If x is not a free variable of φ, then the proof is trivial; hence suppose x is a free variable of φ.

Using $\mathcal{R}(\varphi) \dashv\vdash \mathcal{R}_{u_1}(\varphi)$ and $\mathcal{R}([y/x]\varphi) \dashv\vdash \mathcal{R}_{u_2}([y/x]\varphi)$, we get

$$\mathcal{R}(\psi) \vdash \forall x\forall y((\mathcal{R}_{u_1}(\varphi) \,\&\, \mathcal{R}_{u_2}([y/x]\varphi)) \Rightarrow x = y)$$

Suppose first that u_1 and u_2 differ from y. Then we have that u_1 and u_2 are not free variables of $(\mathcal{R}_{u_1}(\varphi) \,\&\, \mathcal{R}_{u_2}([y/x]\varphi)) \Rightarrow x = y$ and hence we can perform a rename of variables, yielding

$$\mathcal{R}(\psi) \vdash \forall u_1\forall u_2(([u_1/x][u_2/y]\mathcal{R}_{u_1}(\varphi) \,\&\, [u_1/x][u_2/y]\mathcal{R}_{u_2}([y/x]\varphi)) \Rightarrow u_1 = u_2)$$

Since y does not occur in φ, it does not occur free in $\mathcal{R}_{u_1}(\varphi)$, so $[u_2/y]\mathcal{R}_{u_1}(\varphi) \equiv \mathcal{R}_{u_1}(\varphi)$. By lemma 11, $[u_1/x]\mathcal{R}_{u_1}(\varphi) \dashv\vdash \mathcal{R}([u_1/x]\varphi)$. Analogously, we have $[u_1/x][u_2/y]\mathcal{R}_{u_2}([y/x]\varphi) \dashv\vdash \mathcal{R}([u_2/x]\varphi)$, yielding

$$\mathcal{R}(\psi) \vdash \forall u_1\forall u_2((\mathcal{R}([u_1/x]\varphi) \,\&\, \mathcal{R}([u_2/x]\varphi)) \Rightarrow u_1 = u_2)$$

from which we easily get the desired sequent.

The cases where $u_1 \equiv y$ or $u_2 \equiv y$ are similar.

- Next, we handle the $\Leftarrow$ direction. Using the assumption rule, we get

$$\mathcal{R}(\psi)\,, \mathcal{R}([u_1/x]\varphi) \,\&\, u_1 = u_2 \vdash \mathcal{R}([u_1/x]\varphi) \,\&\, u_1 = u_2$$

which is equivalent with

$$\mathcal{R}(\psi)\,, \mathcal{R}([u_1/x]\varphi) \,\&\, u_1 = u_2 \vdash \mathcal{R}_{u_2}([u_1/x]\varphi) \,\&\, u_1 = u_2$$

Using the equality rule, we can deduce

$$\mathcal{R}(\psi)\,, \mathcal{R}([u_1/x]\varphi) \,\&\, u_1 = u_2 \vdash [u_2/u_1]\mathcal{R}_{u_2}([u_1/x]\varphi)$$

Lemma 11 yields that the consequent is equivalent to $\mathcal{R}([u_2/u_1][u_1/x]\varphi)$ which is in turn equivalent to $\mathcal{R}([u_2/x]\varphi)$, since u_1 is not a free variable of φ. Now it is easy to get the desired sequent. □

Lemma 14 *If the translation of $\psi \vdash_\iota \exists!x(\varphi)$ is derivable, then*

$$\mathcal{R}(\psi) \vdash (\mathcal{R}(\varphi) \,\&\, \alpha) \Leftrightarrow (\mathcal{R}(\varphi) \,\&\, \exists x(\mathcal{R}(\varphi) \,\&\, \alpha))$$

Proof.

Choose a variable symbol y not occurring free in φ and α. Because y is not a free variable of $\mathcal{R}(\varphi)$, we have

$$\mathcal{R}(\psi) \vdash (\mathcal{R}(\varphi) \,\&\, \exists y([y/x]\mathcal{R}(\varphi) \,\&\, [y/x]\alpha)) \Leftrightarrow \exists y(\mathcal{R}(\varphi) \,\&\, [y/x]\mathcal{R}(\varphi) \,\&\, [y/x]\alpha)$$

Using the previous lemma, we get

$$\mathcal{R}(\psi) \vdash (\mathcal{R}(\varphi) \,\&\, \exists y([y/x]\mathcal{R}(\varphi) \,\&\, [y/x]\alpha)) \Leftrightarrow \exists y(\mathcal{R}(\varphi) \,\&\, x = y \,\&\, [y/x]\alpha)$$

from which

$$\mathcal{R}(\psi) \vdash (\mathcal{R}(\varphi) \,\&\, \exists y([y/x]\mathcal{R}(\varphi) \,\&\, [y/x]\alpha)) \Leftrightarrow (\mathcal{R}(\varphi) \,\&\, \alpha)$$

and finally by a change of variable name,

$$\mathcal{R}(\psi) \vdash (\mathcal{R}(\varphi) \,\&\, \exists x(\mathcal{R}(\varphi) \,\&\, \alpha)) \Leftrightarrow (\mathcal{R}(\varphi) \,\&\, \alpha).$$

□

The next lemma enables us to eliminate a iota-term $\iota x_\psi(\varphi)$ from a formula α. It is a key step in the equiconsistency proof. This is analogous to

[Hilbert & Bernays 1968], pp. 441–448: there, the theorem reads, translated to our notations,

$$\vdash (\forall x(\mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\alpha))) \Leftrightarrow \mathcal{R}([\iota x(\varphi)/x]\alpha)$$

from which we clearly can observe that our lemma is a generalisation of Hilbert and Bernays's. Moreover, we can prove the lemma in a way that is very close to the proof of Hilbert and Bernays:

Lemma 15 *If the translation of* $\psi \vdash_\iota \exists! x(\varphi)$ *is derivable, and the substitution* $[\iota x_\psi(\varphi)/x]\alpha$ *is defined, then*

$$\mathcal{R}(\psi) \vdash (\forall x(\mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\alpha))) \Leftrightarrow \mathcal{R}([\iota x_\psi(\varphi)/x]\alpha)$$

Proof.

We prove this by structural induction on α.

If x is not a free variable of α, then we have to prove that

$$\mathcal{R}(\psi) \vdash (\forall x(\mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\alpha))) \Leftrightarrow \mathcal{R}(\alpha)$$

which is equivalent with

$$\mathcal{R}(\psi) \vdash (\exists x(\mathcal{R}(\varphi)) \Rightarrow \mathcal{R}(\alpha)) \Leftrightarrow \mathcal{R}(\alpha)$$

This easy to show, using the uniqueness condition.

For the case where x is a free variable of α, we first treat the case where α is atomic, which we handle by induction on the nesting depth of the ι-terms of α.

• First, let us suppose that α does not contain any ι-terms. Hence, $\mathcal{R}(\alpha) \equiv \alpha$. We have to prove that

$$\mathcal{R}(\psi) \vdash (\forall x(\mathcal{R}(\varphi) \Rightarrow \alpha)) \Leftrightarrow \mathcal{R}([\iota x_\psi(\varphi)/x]\alpha)$$

Using lemma 12, this is equivalent to

$$\mathcal{R}(\psi) \vdash (\exists x(\mathcal{R}(\varphi) \,\&\, \alpha)) \Leftrightarrow \mathcal{R}([\iota x_\psi(\varphi)/x]\alpha)$$

Expanding the last reduction, we get

$$\mathcal{R}(\psi) \vdash (\exists x(\mathcal{R}(\varphi) \,\&\, \alpha)) \Leftrightarrow \exists u_1 \exists u_2 \ldots \exists u_k(\mathcal{R}([u_1/x]\varphi) \,\&\, \ldots \,\&\, \mathcal{R}([u_k/x]\varphi) \,\&\, q)$$

where q is the formula obtained by replacing all k top-level occurrences of x in α by u_1, u_2, ... We now use lemma 13 to transform this into

$$\mathcal{R}(\psi) \vdash (\exists x(\mathcal{R}(\varphi) \,\&\, \alpha)) \Leftrightarrow \exists u_1 \ldots \exists u_k(\mathcal{R}([u_1/x]\varphi) \,\&\, u_1 = u_2 \,\&\, \cdots \,\&\, u_1 = u_k \,\&\, q)$$

from which we can get

$$\mathcal{R}(\psi) \vdash (\exists x(\mathcal{R}(\varphi)\ \&\ \alpha)) \Leftrightarrow \exists u_1(\mathcal{R}([u_1/x]\varphi)\ \&\ [u_1/u_2][u_1/u_3]\cdots[u_1/u_k]q)$$

and finally, using $\mathcal{R}([u_1/x]\varphi) \dashv\vdash \mathcal{R}_{u_1}([u_1/x]\varphi)$ and lemma 11,

$$\mathcal{R}(\psi) \vdash (\exists x(\mathcal{R}(\varphi)\ \&\ \alpha)) \Leftrightarrow \exists x(\mathcal{R}(\varphi)\ \&\ [x/u_1][x/u_2]\cdots[x/u_k]q)$$

Noticing that $[x/u_1][x/u_2]\cdots[x/u_k]q \equiv \alpha$ concludes the proof of this case.

• If α contains ι-terms, i.e., $\alpha \equiv p(\tau_1, \ldots, \tau_n)$, where we treat the case $\alpha \equiv \tau_1 = \tau_2$ analogously, then we proceed as follows. We have

$$\begin{aligned}\mathcal{R}(\alpha) &\equiv \exists u_1 \exists u_2 \ldots \exists u_m(\mathcal{R}([u_1/x_1]\varphi_1)\ \&\ \ldots\ \&\ \mathcal{R}([u_m/x_m]\varphi_m)\ \&\ q)\\ \mathcal{R}([\iota x_\psi(\varphi)/x]\alpha) &\equiv \exists w_1 \exists w_2 \ldots \exists w_m \exists v_1 \exists v_2 \ldots \exists v_k\big(\\ &\qquad \mathcal{R}([v_1/x]\varphi)\ \&\ \ldots\ \&\ \mathcal{R}([v_k/x]\varphi)\\ &\qquad \&\ \mathcal{R}([w_1/x_1]\varphi'_1)\ \&\ \ldots\ \&\ \mathcal{R}([w_m/x_m]\varphi'_m)\ \&\ q'\big)\end{aligned}$$

where $TLI(\alpha) \equiv \iota x_{1\psi_1}(\varphi_1), \ldots, \iota x_{m\psi_m}(\varphi_m)$, we obtain q by replacing all top-level ι-terms of α by u_1, u_2, …, u_m, and q' by replacing all top-level free x (i.e., not inside a $\forall x$ quantifier or inside the domain formula of a ι-term) symbols in α by v_1, v_2, …, v_k and all top-level ι-terms by w_1, w_2, …, w_m and φ'_i is defined as the definiens of $[\iota x_\psi(\varphi)/x]\iota x_{i\psi_i}(\varphi_i)$. More explicitly, we have

- If $x \equiv x_i$ or x is not a free variable of φ_i, then $\varphi'_i \equiv \varphi_i$
- If $x \not\equiv x_i$ and x is a free variable of φ_i, then $\varphi'_i \equiv [\iota x_\psi(\varphi)/x]\varphi_i$. Note that we supposed the substitution $[\iota x_\psi(\varphi)/x]\alpha$ to be defined, hence the substitution $[\iota x_\psi(\varphi)/x]\iota x_{i\psi_i}(\varphi_i)$ is defined (and hence x_i is not a free variable of $\iota x_\psi(\varphi)$).

By proceeding similarly to the previous case, we find that under the condition $\mathcal{R}(\psi)$,

$$\begin{aligned}\mathcal{R}([\iota x_\psi(\varphi)/x]\alpha) \dashv\vdash \exists w_1 \exists w_2 \ldots \exists w_m \exists x\big(&\\ \mathcal{R}(\varphi)\ \&\ \mathcal{R}([w_1/x_1]\varphi'_1)\ \&\ \cdots\ \&\ \mathcal{R}([w_m/x_m]\varphi'_m)&\\ \&\ [x/v_1][x/v_2]\cdots[x/v_k]q'\big)&\end{aligned}$$

We have to prove that this is equivalent under the condition $\mathcal{R}(\psi)$ to $\forall x(\mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\alpha))$, or, using lemma 12, to $\exists x(\mathcal{R}(\varphi)\ \&\ \mathcal{R}(\alpha))$. We have

that

$$\begin{aligned}&\exists x(\mathcal{R}(\varphi)\,\&\,\mathcal{R}(\alpha))\\ &\quad\dashv\vdash \exists x(\mathcal{R}(\varphi)\,\&\,\exists u_1\ldots\exists u_m(\mathcal{R}([u_1/x_1]\varphi_1)\,\&\cdots\&\,\mathcal{R}([u_m/x_m]\varphi_m)\,\&\,q))\\ &\quad\dashv\vdash \exists x\exists u_1\ldots\exists u_m(\mathcal{R}(\varphi)\,\&\,\mathcal{R}([u_1/x_1]\varphi_1)\,\&\cdots\&\,\mathcal{R}([u_m/x_m]\varphi_m)\,\&\,q)\end{aligned}$$

We supposed here that the u's are not free variables of $\mathcal{R}(\varphi)$; if this would be the case, we would have to perform an extra change of variable names here.
Define

$$\begin{aligned}C &:= [u_{i_1}/x_{i_1}]\varphi_{i_1}\,\&\,[u_{i_2}/x_{i_2}]\varphi_{i_2}\,\&\cdots\&\,[u_{i_c}/x_{i_c}]\varphi_{i_c}\\ D &:= [u_{j_1}/x_{j_1}]\varphi_{j_1}\,\&\,[u_{j_2}/x_{j_2}]\varphi_{j_2}\,\&\cdots\&\,[u_{j_d}/x_{j_d}]\varphi_{j_d}\end{aligned}$$

where in C, all φ_i occur for which $\varphi_i' \equiv [\iota x_\psi(\varphi)/x_i]\varphi_i$ and in D, all φ_j for which $\varphi_j' \equiv \varphi_j$ occur. Then

$$\exists x(\mathcal{R}(\varphi)\,\&\,\mathcal{R}(\alpha)) \dashv\vdash \exists x\exists u_1\exists u_2\ldots\exists u_m(\mathcal{R}(\varphi)\,\&\,\mathcal{R}(C)\,\&\,\mathcal{R}(D)\,\&\,q)$$

Using the previous lemma, this is equivalent under $\mathcal{R}(\psi)$ to

$$\exists x\exists u_1\exists u_2\ldots\exists u_m(\mathcal{R}(\varphi)\,\&\,\exists x(\mathcal{R}(\varphi)\,\&\,\mathcal{R}(C))\,\&\,\mathcal{R}(D)\,\&\,q)$$

We now apply induction on C, yielding

$$\mathcal{R}(\psi)\vdash(\exists x(\mathcal{R}(\varphi)\,\&\,\mathcal{R}(C))\Leftrightarrow\mathcal{R}([\iota x_\psi(\varphi)/x]C)$$

so we have that $\exists x(\mathcal{R}(\varphi)\,\&\,\mathcal{R}(\alpha))$ is equivalent to

$$\exists x\exists u_1\exists u_2\ldots\exists u_m(\mathcal{R}(\varphi)\,\&\,\mathcal{R}([\iota x_\psi(\varphi)/x]C)\,\&\,\mathcal{R}(D)\,\&\,q)$$

i.e.,

$$\begin{aligned}\exists x\exists u_1\exists u_2\ldots\exists u_m\Big(&\mathcal{R}(\varphi)\\ &\&\,\mathcal{R}([\iota x_\psi(\varphi)/x][u_{i_1}/x_{i_1}]\varphi_{i_1}\,\&\cdots\&\,[\iota x_\psi(\varphi)/x][u_{i_c}/x_{i_c}]\varphi_{i_c})\\ &\&\,\mathcal{R}([u_{j_1}/x_{j_1}]\varphi_{j_1}\,\&\cdots\&\,[u_{j_d}/x_{j_d}]\varphi_{j_d})\,\&\,q\Big)\end{aligned}$$

Now u_{i_1} does not occur in α and hence differs from x, and as we remarked before, x_{i_1} is not a free variable of $\iota x_\psi(\varphi)$. Further, x_{i_1} differs from x. Hence the conditions for property 7 are fulfilled and we have $[\iota x_\psi(\varphi)/x][u_{i_1}/x_{i_1}]\varphi_{i_1} \equiv$

$[u_{i_1}/x_{i_1}][\iota x_\psi(\varphi)/x]\varphi_{i_1}$ and likewise for φ_{i_2} and so on. Hence we can transform our expression into

$$\exists u_1 \exists u_2 \ldots \exists u_m \exists x \Big(\mathcal{R}(\varphi) \,\&\, \mathcal{R}\big([u_{i_1}/x_{i_1}]\varphi'_{i_1} \,\&\, \cdots \,\&\, [u_{i_c}/x_{i_c}]\varphi'_{i_c}\big) \\ \&\, \mathcal{R}\big([u_{j_1}/x_{j_1}]\varphi'_{j_1} \,\&\, \cdots \,\&\, [u_{j_d}/x_{j_d}]\varphi'_{j_d}\big) \,\&\, q \Big)$$

But this is $\mathcal{R}([\iota x_\psi(\varphi)/x]\alpha)$ up to a choice of the u's, and we know that such reductions are equivalent.

If α is not atomic, then we can apply induction on the structure of α. □

Note that we need the antecedent $\mathcal{R}(\psi)$. Indeed, otherwise setting $\alpha \equiv x = x$ and $\iota x_\psi(\varphi) \equiv \iota x_{y\neq 0}(x \cdot y = 1)$, we would obtain

$$\vdash (\forall x(x \cdot y = 1 \Rightarrow x = x)) \Leftrightarrow \exists u \exists v(u \cdot y = 1 \,\&\, v \cdot y = 1 \,\&\, u = v)$$

Since $x = x$ is a validity, so is $x \cdot y = 1 \Rightarrow x = x$, so the sequent obtained is equivalent with

$$\vdash \exists u \exists v(u \cdot y = 1 \,\&\, v \cdot y = 1 \,\&\, u = v)$$

from which we easily could derive the unsound sequent

$$\vdash \exists u \exists v(u \cdot 0 = 1 \,\&\, v \cdot 0 = 1 \,\&\, u = v).$$

Lemma 16 *Let t be a term of the* PITFOL *calculus and α a formula of the* PITFOL *calculus. If the translations of the uniqueness conditions for t hold and the substitution $[t/x]\alpha$ is defined, then*

$$\mathcal{D}(t)\,, \forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}([t/x]\alpha)$$

Proof.

- If t does not contain any ι-terms, then $\mathcal{D}(t)$ is a validity and we have to prove
$$\forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}([t/x]\alpha)$$
We have $\forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}(\alpha)$ and because of lemma 8, this is equivalent with $\forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}_V(\alpha)$ where V is the set of free variables of α. Applying the substitution rule yields $\forall x(\mathcal{R}(\alpha)) \vdash [t/x]\mathcal{R}_V(\alpha)$ Using lemma 11 we can transform this into $\forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}([t/x]\alpha)$.

- If $t \equiv \iota x_\psi(\varphi)$, then we use lemma 15 to get

$$\mathcal{D}(t) \vdash (\forall x(\mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\alpha))) \Leftrightarrow \mathcal{R}([t/x]\alpha)$$

and hence

$$\mathcal{D}(t)\,, \forall x(\mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\alpha)) \vdash \mathcal{R}([t/x]\alpha)$$

One easily shows that $\forall x(\mathcal{R}(\alpha)) \vdash \forall x(\mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\alpha))$ and using this result, this case is easily proved.

- The case $t \equiv \iota y_\psi(\varphi)$ with $y \not\equiv x$ is handled as follows. Choose a variable symbol z different from x and which does not occur in α and t. Applying the previous case to the formula $[y/x][z/y]\alpha$, we get

$$\mathcal{D}(t)\,, \forall y(\mathcal{R}([y/x][z/y]\alpha)) \vdash \mathcal{R}([t/y][y/x][z/y]\alpha)$$

which is equivalent to

$$\mathcal{D}(t)\,, \forall x(\mathcal{R}([z/y]\alpha)) \vdash \mathcal{R}_y([t/x][z/y]\alpha)$$

from which we can derive

$$[y/z]\mathcal{D}(t)\,, [y/z]\forall x(\mathcal{R}([z/y]\alpha)) \vdash [y/z]\mathcal{R}_y([t/x][z/y]\alpha)$$

Since z does not occur in t, it is not a free variable of $\mathcal{D}(t)$ and $[y/z]\mathcal{D}(t) \equiv \mathcal{D}(t)$.
Next, $[y/z]\forall x(\mathcal{R}([z/y]\alpha)) \equiv \forall x([y/z]\mathcal{R}([z/y]\alpha))$ and the latter formula is, using lemma 11, equivalent with $\forall x([y/z][z/y]\mathcal{R}_z(\alpha))$ which is identical to $\forall x(\mathcal{R}_z(\alpha))$. From lemma 8 we learn that this is equivalent with $\forall x(\mathcal{R}(\alpha))$.
Finally, $[y/z]\mathcal{R}_y([t/x][z/y]\alpha)$ is equivalent with $\mathcal{R}([y/z][t/x][z/y]\alpha)$. Because z is not a free variable of t, we can swap the order of the substitutions; the last formula is identical to $\mathcal{R}([t/x][y/z][z/y]\alpha)$ which in turn is identical to $\mathcal{R}([t/x]\alpha)$.
Combining these three observations yields the desired sequent.

- If $t \equiv f(t_1, t_2, \ldots, t_n)$ then we enumerate the top-level ι-terms of t as

$$TLI(t) \equiv \iota x_{1\psi_1}(\varphi_1), \iota x_{2\psi_2}(\varphi_2), \ldots, \iota x_{m\psi_m}(\varphi_m)$$

and define g as the term obtained by replacing all top-level ι-terms in t with the variable symbols $u_1, u_2, \ldots, u_m$, which we choose all different and not occurring in t or α. Applying the first case of our proof yields

$$\forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}([g/x]\alpha)$$

from which we easily get

$$\forall x(\mathcal{R}(\alpha)) \vdash \forall u_1(\mathcal{R}([g/x]\,\alpha))$$

Applying the previous case, we get

$$\mathcal{R}(\psi_1)\,, \forall u_1(\mathcal{R}([g/x]\,\alpha)) \vdash \mathcal{R}\big(\big[\iota x_{1\psi_1}(\varphi_1)/u_1\big]\,[g/x]\,\alpha\big)$$

Combining these sequents yields

$$\mathcal{R}(\psi_1)\,, \forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}\big(\big[\iota x_{1\psi_1}(\varphi_1)/u_1\big]\,[g/x]\,\alpha\big)$$

Continuing in this way, we get

$$\mathcal{R}(\psi_1)\,, \ldots, \mathcal{R}(\psi_m)\,, \forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}\big(\big[\iota x_{m\psi_m}(\varphi_m)/u_m\big] \ldots \big[\iota x_{1\psi_1}(\varphi_1)/u_1\big]\,[g/x]\,\alpha\big)$$

i.e.,

$$\mathcal{D}(t)\,, \forall x(\mathcal{R}(\alpha)) \vdash \mathcal{R}([t/x]\,\alpha)$$

□

Note that we need $\mathcal{D}(t)$ to be present in the antecedent. As a counterexample, consider $\alpha \equiv x = x$ and $t \equiv \iota x_{y\neq 0}(x \cdot y = 1)$. Then we would get $\forall x(x = x) \vdash \exists u \exists v(u \cdot y = 1 \,\&\, v \cdot y = 1 \,\&\, u = v)$ from which we easily could derive the unsound sequent $\vdash \exists u \exists v(u \cdot 0 = 1 \,\&\, v \cdot 0 = 1 \,\&\, u = v)$.

Lemma 17 *Let t be a term of the* PITFOL *calculus and α a formula of the* PITFOL *calculus. If the substitution $[t/x]\,\alpha$ is defined, then*

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\,\mathcal{D}_V(\alpha)) \vdash \mathcal{D}([t/x]\,\alpha)$$

where V is the set of free variables of α.

Proof.

If x is not a free variable of α, then the proof is trivial, so we will suppose that x is a free variable of α.

- If α is atomic, then call $TLI(\alpha) \equiv \iota x_{1\psi_1}(\varphi_1), \iota x_{2\psi_2}(\varphi_2), \ldots, \iota x_{m\psi_m}(\varphi_m)$ its top-level ι-terms. Then,

$$\mathcal{D}_V(\alpha) \equiv \mathcal{R}_V(\psi_1) \,\&\, \mathcal{R}_V(\psi_2) \,\&\, \cdots \,\&\, \mathcal{R}_V(\psi_m)$$

If we denote the top-level ι-terms of $[t/x]\alpha$ as $TLI([t/x]\alpha) \equiv \iota x'_{1\psi'_1}(\varphi'_1), \iota x'_{2\psi'_2}(\varphi'_2), \ldots, \iota x'_{M\psi'_M}(\varphi'_M)$ then analogously, we have

$$\mathcal{D}([t/x]\alpha) \equiv \mathcal{R}(\psi'_1) \,\&\, \mathcal{R}(\psi'_2) \,\&\, \cdots \,\&\, \mathcal{R}(\psi'_M)$$

Denoting the domain formula of $[t/x]\iota x_{i\psi_i}(\varphi_i)$ as ψ''_i, we see that the ψ''s contain all the ψ'''s and the rest of them originate from the top-level ι-terms of t.
Summarizing, we have that

$$\mathcal{R}(\psi'_1)\,\&\,\mathcal{R}(\psi'_2)\,\&\cdots\&\,\mathcal{R}(\psi'_M) \dashv\vdash \mathcal{R}(\psi''_1)\,\&\,\mathcal{R}(\psi''_2)\,\&\ldots\&\,\mathcal{R}(\psi''_m)\,\&\,\mathcal{D}(t)$$

We will now establish that either $\psi''_i \equiv [t/x]\psi_i$ or $\psi''_i \equiv \mathbf{\Delta}(t) \,\&\, [t/x]\psi_i$.

- If $x \equiv x_i$ or x is not a free variable of φ_i, then $\psi''_i \equiv \psi_i$ when x is not a free variable of ψ_i and hence $\psi''_i \equiv [t/x]\psi_i$, or $\psi''_i \equiv \mathbf{\Delta}(t)\,\&\,[t/x]\psi_i$ when x is a free variable of ψ_i.
- Else, $\psi''_i \equiv \mathbf{\Delta}(t) \,\&\, [t/x]\psi_i$.

Hence, $\mathcal{R}(\psi''_i)$ is equivalent to $\mathcal{R}([t/x]\psi_i)$ or $\mathcal{D}(t) \,\&\, \mathcal{R}([t/x]\psi_i)$.
So what we have to prove amounts to

$$\begin{aligned}\mathcal{D}(t)\,, \mathcal{R}([t/x](\mathcal{R}_V(\psi_1) \,\&\, \cdots \,\&\, \mathcal{R}_V(\psi_m))) \\ \vdash \mathcal{R}([t/x]\psi_1) \,\&\, \cdots \,\&\, \mathcal{R}([t/x]\psi_m) \,\&\, \mathcal{D}(t)\end{aligned}$$

for which it will be sufficient to derive

$$\mathcal{R}([t/x](\mathcal{R}_V(\psi_i)) \vdash \mathcal{R}([t/x]\psi_i)$$

which we immediately obtain from lemma 11.

- If $\alpha \equiv \neg\beta$ then we must prove

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\mathcal{D}_V(\neg\beta)) \vdash \mathcal{D}([t/x]\neg\beta)$$

Because of the definition of $\mathcal{D}$, this is identical to the induction hypothesis

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\mathcal{D}_V(\beta)) \vdash \mathcal{D}([t/x]\beta)$$

- If $\alpha \equiv \beta \,\&\, \gamma$, then we have to show that

$$\begin{aligned}\mathcal{D}(t)\,, \mathcal{R}([t/x]\mathcal{D}_V(\beta) \,\&\, [t/x](\mathcal{R}_V(\beta) \Rightarrow \mathcal{D}_V(\gamma))) \\ \vdash \mathcal{D}([t/x]\beta) \,\&\, (\mathcal{R}([t/x]\beta) \Rightarrow \mathcal{D}([t/x]\gamma))\end{aligned}$$

Induction on β yields $\mathcal{D}(t)\,, \mathcal{R}([t/x]\mathcal{D}_V(\beta)) \vdash \mathcal{D}([t/x]\beta)$, so we still have to derive

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\mathcal{R}_V(\beta)) \Rightarrow \mathcal{R}([t/x]\mathcal{D}_V(\gamma))) \vdash \mathcal{R}([t/x]\beta) \Rightarrow \mathcal{D}([t/x]\gamma)\,.$$

Invoking lemma 11, this becomes

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\beta) \Rightarrow \mathcal{R}([t/x]\mathcal{D}_V(\gamma))) \vdash \mathcal{R}([t/x]\beta) \Rightarrow \mathcal{D}([t/x]\gamma)$$

which is easy to prove using induction on γ.

- If $\alpha \equiv \forall y(\beta)$, then, using the induction hypothesis, we can derive

$$\forall y(\mathcal{D}(t)), \forall y(\mathcal{R}([t/x]\mathcal{D}_V(\beta))) \vdash \forall y(\mathcal{D}([t/x]\beta))$$

Since the substitution $[t/x]\alpha$ is derived, either x is not a free variable of α, and then the lemma is trivial, or y is not a free variable of t. In the latter case, we have $\mathcal{D}(t) \vdash \forall y(\mathcal{D}(t))$, and we can derive

$$\mathcal{D}(t)\,, \forall y(\mathcal{R}([t/x]\mathcal{D}_V(\beta))) \vdash \forall y(\mathcal{D}([t/x]\beta))$$

Noticing that $\forall y(\mathcal{R}([t/x]\mathcal{D}_V(\beta))) \equiv \mathcal{R}([t/x]\mathcal{D}_V(\forall y(\alpha)))$ and $\forall y(\mathcal{D}([t/x]\beta)) \equiv \mathcal{D}([t/x]\forall y(\beta))$ proves this case.

□

Corollary 18 *Let t be a term of the* PITFOL *calculus and α a formula of the* PITFOL *calculus. If the translations of the uniqueness conditions for t hold and the substitution $[t/x]\alpha$ is defined, then*

$$\mathcal{D}(t)\,, \forall x(\mathcal{D}(\alpha)) \vdash \mathcal{D}([t/x]\alpha)$$

Proof.

Applying lemma 16 on the formula $\mathcal{D}_V(\alpha)$ gives

$$\mathcal{D}(t)\,, \forall x(\mathcal{R}(\mathcal{D}_V(\alpha))) \vdash \mathcal{R}([t/x]\mathcal{D}_V(\alpha))$$

In combination with

$$\forall x(\mathcal{R}(\mathcal{D}(\alpha))) \vdash \forall x(\mathcal{R}(\mathcal{D}_V(\alpha)))$$

this yields

$$\mathcal{D}(t)\,, \forall x(\mathcal{R}(\mathcal{D}(\alpha))) \vdash \mathcal{R}([t/x]\mathcal{D}_V(\alpha))$$

Applying the previous lemma yields

$$\mathcal{D}(t)\,, \forall x(\mathcal{R}(\mathcal{D}(\alpha))) \vdash \mathcal{D}([t/x]\alpha)$$

which is the desired sequent, using the observations in §3.6.3. □

Lemma 19 *If the translation of* $\psi \vdash_\iota \exists!y(\varphi)$ *is derivable and the translations of the uniqueness conditions of* t *are given, then so is the translation of the uniqueness condition of* $[t/x]\iota y_\psi(\varphi)$*, if the substitution is defined.*

Proof.

The translation of the uniqueness condition we have at our disposal is

$$\begin{cases} \vdash \mathcal{D}(\psi) \\ \mathcal{R}(\psi) \vdash \mathcal{R}(\exists!y(\varphi)) \,\&\, \mathcal{D}(\exists!y(\varphi)) \end{cases}$$

Denote the result of the substitution, $[t/x]\iota y_\psi(\varphi)$, as $\iota y_{\psi'}(\varphi')$. We have to derive

$$\begin{cases} \vdash \mathcal{D}(\psi') \\ \mathcal{R}(\psi') \vdash \mathcal{R}(\exists!y(\varphi')) \,\&\, \mathcal{D}(\exists!y(\varphi')) \end{cases}$$

• First, we show how $\vdash \mathcal{D}(\psi')$ can be derived from $\vdash \mathcal{D}(\psi)$. If $\psi \equiv \psi'$ then we have nothing to do; otherwise we have to derive $\vdash \mathcal{D}(\mathbf{\Delta}(t) \,\&\, [t/x]\psi)$, i.e.,

$$\vdash \mathcal{D}(\mathbf{\Delta}(t)) \,\&\, (\mathcal{R}(\mathbf{\Delta}(t)) \Rightarrow \mathcal{D}([t/x]\psi))$$

Using lemmas 10 and 9, this simplifies to

$$\vdash \mathcal{D}(t) \Rightarrow \mathcal{D}([t/x]\psi))$$

which we get easily from $\vdash \mathcal{D}(\psi)$ using corollary 18.

• Next, we prove that from $\mathcal{R}(\psi) \vdash \mathcal{R}(\exists!y(\varphi))$, we can obtain $\mathcal{R}(\psi') \vdash \mathcal{R}(\exists!y(\varphi'))$.

If $x \not\equiv y$ and x is a free variable of φ, then we have to obtain

$$\mathcal{D}(t)\&\mathcal{R}([t/x]\psi) \vdash \exists y(\mathcal{R}([t/x]\varphi))\&\forall y\forall v((\mathcal{R}([t/x]\varphi)\&\mathcal{R}([v/y][t/x]\varphi)) \Rightarrow y = v)$$

where v does not occur in $[t/x]\varphi$, and we already have

$$\mathcal{R}(\psi) \vdash \exists y(\mathcal{R}(\varphi)) \,\&\, \forall y\forall z((\mathcal{R}(\varphi) \,\&\, \mathcal{R}([z/y]\varphi)) \Rightarrow y = z)$$

where z does not occur in φ.

We can assume that $v \not\equiv x$; if $v \equiv x$ then one can prove easily that replacing v by another variable also not occurring in $[t/x]\varphi$ yields an equivalent sequent.

Using the property that $\mathcal{R}(\alpha)$ is equivalent with $\mathcal{R}_v(\alpha)$, we get after renaming z to v:

$$\mathcal{R}(\psi) \vdash \exists y(\mathcal{R}(\varphi)) \,\&\, \forall y\forall v(([v/z]\mathcal{R}_v(\varphi) \,\&\, [v/z]\mathcal{R}_v([z/y]\varphi)) \Rightarrow y = v)$$

Since z does not occur in φ, it is not a free variable of $\mathcal{R}(\varphi)$ and hence $[v/z]\mathcal{R}_v(\varphi) \equiv \mathcal{R}(\varphi)$.
Using lemma 11, we also get that $[v/z]\mathcal{R}_v([v/y]\varphi) \dashv\vdash \mathcal{R}([v/z][z/y]\varphi)$; since z does not occur in φ, the latter sequent can be written as $\mathcal{R}([v/y]\varphi)$.
These two observations yield

$$\mathcal{R}(\psi) \vdash \exists y(\mathcal{R}(\varphi)) \,\&\, \forall y \forall v((\mathcal{R}(\varphi) \,\&\, \mathcal{R}([v/y]\varphi)) \Rightarrow y = v)$$

from which we easily derive

$$\vdash \forall x(\mathcal{R}_V(\psi) \Rightarrow \exists y(\mathcal{R}_V(\varphi)) \,\&\, \forall y \forall v((\mathcal{R}_V(\varphi) \,\&\, \mathcal{R}_V([v/y]\varphi)) \Rightarrow y = v))$$

where V is the set of free variables of t. Using lemma 16, we easily obtain

$$\begin{aligned}\mathcal{D}(t) \vdash \mathcal{R}([t/x]\mathcal{R}_V(\psi)) \Rightarrow \Big(&\mathcal{R}([t/x]\exists y(\mathcal{R}_V(\varphi))) \,\&\\ &\mathcal{R}([t/x]\forall y \forall v((\mathcal{R}_V(\varphi) \,\&\, \mathcal{R}_V([v/y]\varphi)) \Rightarrow y = v))\Big)\end{aligned}$$

Applying lemma 11 yields

$$\begin{aligned}\mathcal{D}(t) \vdash \mathcal{R}([t/x]\psi) \Rightarrow \Big(&\exists y(\mathcal{R}([t/x]\varphi)) \&\\ &\forall y \forall v((\mathcal{R}([t/x]\varphi) \,\&\, \mathcal{R}([t/x][v/y]\varphi)) \Rightarrow y = v)\Big)\end{aligned}$$

Since y is not a free variable of t (because the substitution is defined) and $x \not\equiv v$, we have that $[t/x][v/y]\varphi \equiv [v/y][t/x]\varphi$. Now we can easily derive the desired sequent.

If $x \equiv y$ or x is not a free variable of φ, then $\varphi' \equiv \varphi$. If also x is not a free variable of ψ, then $\psi' = \psi$ and we have nothing to do; so suppose $\psi' = \boldsymbol{\Delta}(t) \,\&\, [t/x]\psi$. We rewrite the sequent at our disposal as

$$\vdash \mathcal{R}(\psi \Rightarrow \exists!y(\varphi))$$

Using lemma 16, we easily get

$$\mathcal{D}(t) \vdash \mathcal{R}([t/x]\psi \Rightarrow [t/x]\exists!y(\varphi))$$

If $x \equiv y$, then $[t/x]\exists!y(\varphi) \equiv \exists!y(\varphi)$; the same holds if x is not a free variable of φ:

$$\begin{aligned}[t/x]\exists!y(\varphi) &\equiv [t/x]\exists y(\mathcal{R}(\varphi)) \,\&\, [t/x]\forall y \forall z((\mathcal{R}(\varphi) \,\&\, \mathcal{R}([z/y]\varphi)) \Rightarrow y = z)\\ &\equiv \exists y(\mathcal{R}(\varphi)) \,\&\, \forall y \forall z((\mathcal{R}(\varphi) \,\&\, \mathcal{R}([z/y]\varphi)) \Rightarrow y = z)\end{aligned}$$

at least if z is not a free variable of φ; in that case, we would have to perform a variable renaming as the first step (before applying lemma 16).

Hence we easily can transform our sequent into

$$\mathcal{D}(t)\,,\mathcal{R}([t/x]\psi) \vdash \mathcal{R}(\exists!y(\varphi))$$

which is by lemma 9 equivalent with

$$\mathcal{R}(\mathbf{\Delta}(t)\ \&\ [t/x]\psi) \vdash \mathcal{R}(\exists!y(\varphi))$$

• Finally, we prove that from $\mathcal{R}(\psi) \vdash \mathcal{R}(\exists!y(\varphi))\ \&\ \mathcal{D}(\exists!y(\varphi))$, we can obtain $\mathcal{R}(\psi') \vdash \mathcal{D}(\exists!y(\varphi'))$.

If $x \not\equiv y$ and x is a free variable of φ, then we have to get $\mathcal{R}(\mathbf{\Delta}(t)\ \&\ [t/x]\psi) \vdash \mathcal{D}(\exists!y([t/x]\varphi))$, i.e., we have to obtain

$$\begin{array}{l}\mathcal{D}(t)\,,\mathcal{R}([t/x]\psi) \vdash \mathcal{D}(\exists y([t/x]\varphi))\ \&\\ \quad(\exists y(\mathcal{R}([t/x]\varphi)) \Rightarrow (\forall y\forall v(\mathcal{D}([t/x]\varphi\ \&\ [v/y]\,[t/x]\varphi \Rightarrow y = v))))\end{array}$$

where v does not occur in $[t/x]\varphi$. From the previous part of the proof, we have $\mathcal{D}(t)\,,\mathcal{R}([t/x]\psi) \vdash \exists y(\mathcal{R}([t/x]\varphi))$, so it is sufficient to derive

$$\mathcal{D}(t)\,,\mathcal{R}([t/x]\psi) \vdash \mathcal{D}(\exists y([t/x]\varphi))\ \&\ \forall y\forall v(\mathcal{D}([t/x]\varphi\ \&\ [v/y]\,[t/x]\varphi \Rightarrow y = v)))$$

From the given sequent, $\mathcal{R}(\psi) \vdash \mathcal{R}(\exists!y(\varphi))\ \&\ \mathcal{D}(\exists!y(\varphi))$, we get

$$\mathcal{R}(\psi) \vdash \exists y(\mathcal{R}(\varphi))\&\mathcal{D}(\exists y(\varphi))\&(\exists y(\mathcal{R}(\varphi)) \Rightarrow (\forall y\forall z(\mathcal{D}(\varphi\ \&\ [z/y]\varphi \Rightarrow y = z))))$$

where z does not occur in φ. From which in turn,

$$\mathcal{R}(\psi) \vdash \mathcal{D}(\exists y(\varphi))\ \&\ \forall y\forall z(\mathcal{D}(\varphi\ \&\ [z/y]\varphi)\ \&\ (\mathcal{R}(\varphi\ \&\ [z/y]\varphi) \Rightarrow \mathcal{D}(y = z))))$$

Using the fact that $\mathcal{D}(y = z)$ is a validity, we get

$$\mathcal{R}(\psi) \vdash \mathcal{D}(\exists y(\varphi))\ \&\ \forall y\forall z(\mathcal{D}(\varphi\ \&\ [z/y]\varphi)))$$

which is equivalent with

$$\mathcal{R}(\psi) \vdash \forall y(\mathcal{D}(\varphi))\ \&\ \forall y\forall z(\mathcal{D}(\varphi)\ \&\ (\mathcal{R}(\varphi) \Rightarrow \mathcal{D}_v([z/y]\varphi))))$$

We rename z to v:

$$\mathcal{R}(\psi) \vdash \forall y(\mathcal{D}(\varphi))\ \&\ \forall y\forall v([v/z]\mathcal{D}(\varphi)\ \&\ ([v/z]\mathcal{R}(\varphi) \Rightarrow [v/z]\mathcal{D}_v([z/y]\varphi))))$$

Since z does not occur in φ, we can conclude that $[v/z]\mathcal{R}(\varphi) \equiv \mathcal{R}(\varphi)$ and $[v/z]\mathcal{D}(\varphi) \equiv \mathcal{D}(\varphi)$.
Lemma 17 yields $[v/z]\mathcal{D}_v([z/y]\varphi) \vdash \mathcal{D}([v/z]\,[z/y]\varphi)$ and the last sequent is

$\mathcal{D}([v/y]\varphi)$ since z does not occur in φ.
So we can transform our sequent into

$$\mathcal{R}(\psi) \vdash \forall y(\mathcal{D}(\varphi)) \,\&\, \forall y \forall v(\mathcal{D}(\varphi) \,\&\, (\mathcal{R}(\varphi) \Rightarrow \mathcal{D}([v/y]\varphi))))$$

which we can simplify to

$$\mathcal{R}(\psi) \vdash \forall y \forall v(\mathcal{D}(\varphi) \,\&\, (\mathcal{R}(\varphi) \Rightarrow \mathcal{D}([v/y]\varphi))))$$

Finally, we transform the sequent into

$$\vdash \mathcal{R}(\psi \Rightarrow \forall y \forall v(\mathcal{D}_V(\varphi) \,\&\, (\varphi \Rightarrow \mathcal{D}_V([v/y]\varphi)))))$$

Combining this with an application of lemma 16 yields

$$\mathcal{D}(t) \vdash \mathcal{R}([t/x]\psi \Rightarrow \forall y \forall v([t/x]\mathcal{D}_V(\varphi) \,\&\, ([t/x]\varphi \Rightarrow [t/x]\mathcal{D}_V([v/y]\varphi)))))$$

Using lemma 17, we obtain

$$\mathcal{D}(t) \vdash \mathcal{R}([t/x]\psi) \Rightarrow \forall y \forall v(\mathcal{D}([t/x]\varphi) \,\&\, (\mathcal{R}([t/x]\varphi) \Rightarrow \mathcal{D}([t/x][v/y]\varphi))))$$

For the same reasons of the previous part of the proof, we can change the order of the substitutions. We then easily obtain the sequent we had to derive.

If $x \equiv y$ or x is not a free variable of φ, then again $\varphi' \equiv \varphi$. If also x is not a free variable of ψ, then $\psi' = \psi$ and we have nothing to do; so suppose $\psi' = \mathbf{\Delta}(t) \,\&\, [t/x]\psi$.

Reasoning analogously as in the previous case, we find that we have to obtain

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\psi) \vdash \mathcal{D}(\exists y(\varphi)) \,\&\, \forall y \forall v(\mathcal{D}(\varphi \,\&\, [v/y]\varphi \Rightarrow y = v)))$$

and that we can derive from the sequent we have at our disposal the sequent

$$\mathcal{D}(t) \vdash \mathcal{R}([t/x]\psi \Rightarrow [t/x]\forall y \forall v(\mathcal{D}_V(\varphi) \,\&\, (\varphi \Rightarrow \mathcal{D}_V([v/y]\varphi)))))$$

If $x \equiv y$, then we see immediately that this is equal to

$$\mathcal{D}(t) \vdash \mathcal{R}([t/x]\psi \Rightarrow \forall y \forall v(\mathcal{D}_V(\varphi) \,\&\, (\varphi \Rightarrow \mathcal{D}_V([v/y]\varphi)))))$$

and if x is not a free variable of φ, the same holds (provided we chose v such that it is not a free variable of t, which always is possible using a variable renaming). From this sequent, it is not difficult to derive the required sequent. □

Substitution rule

The translation of the premise is

$$\begin{cases} \vdash \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n) \\ \mathcal{R}(\Sigma) \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \cdots \,\&\, \mathcal{D}(\gamma_m) \\ \mathcal{R}(\Sigma)\,, \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha) \,\&\, \mathcal{D}(\alpha) \end{cases}$$

and from these sequents, we have to prove that

$$\begin{cases} \vdash \mathcal{D}(\mathbf{\Delta}(t) \,\&\, [t/x]\sigma_1 \,\&\, [t/x]\sigma_2 \,\&\, \cdots \,\&\, [t/x]\sigma_n) \\ \mathcal{R}(\mathbf{\Delta}(t))\,, \mathcal{R}([t/x]\Sigma) \vdash \mathcal{D}([t/x]\gamma_1) \,\&\, \mathcal{D}([t/x]\gamma_2) \,\&\, \cdots \,\&\, \mathcal{D}([t/x]\gamma_m) \\ \mathcal{R}(\mathbf{\Delta}(t))\,, \mathcal{R}([t/x]\Sigma)\,, \mathcal{R}([t/x]\Gamma) \vdash \mathcal{R}([t/x]\alpha) \,\&\, \mathcal{D}([t/x]\alpha) \end{cases}$$

• We start with deriving the first sequent of the translation of the conclusion, which is equivalent to the two sequents

$$\begin{aligned} \vdash \mathcal{D}(\mathbf{\Delta}(t)) \\ \mathcal{R}(\mathbf{\Delta}(t)) \vdash \mathcal{D}([t/x]\sigma_1 \,\&\, [t/x]\sigma_2 \,\&\, \cdots \,\&\, [t/x]\sigma_n) \end{aligned}$$

Remembering that $\mathbf{\Delta}(t) \equiv \mathbf{\Delta}(t = x)$, lemma 10 yields the first sequent and we can apply lemma 9 to simplify the second sequent to

$$\mathcal{D}(t) \vdash \mathcal{D}([t/x]\sigma_1 \,\&\, [t/x]\sigma_2 \,\&\, \cdots \,\&\, [t/x]\sigma_n)$$

From corollary 18 we learn that

$$\mathcal{D}(t)\,, \forall x(\mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n)) \vdash \mathcal{D}([t/x]\,(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n))$$

From the first sequent of the translation of the premise, we can derive $\vdash \forall x(\mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n))$, which we can use to simplify the former sequent to

$$\mathcal{D}(t) \vdash \mathcal{D}([t/x]\,(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n))$$

which is the desired sequent.

• To derive the second sequent of the translation of the conclusion, we modify the second sequent of the translation of the premise into

$$\vdash \mathcal{R}((\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n) \Rightarrow (\mathcal{D}_V(\gamma_1) \,\&\, \mathcal{D}_V(\gamma_2) \,\&\, \cdots \,\&\, \mathcal{D}_V(\gamma_m)))$$

where V is the set of free variables of t. From this sequent, we can deduce

$$\vdash \forall x(\mathcal{R}((\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n) \Rightarrow (\mathcal{D}_V(\gamma_1) \,\&\, \mathcal{D}_V(\gamma_2) \,\&\, \cdots \,\&\, \mathcal{D}_V(\gamma_m))))$$

Combining this result with an application of lemma 16, we get

$$\mathcal{D}(t) \vdash \mathcal{R}(([t/x]\sigma_1 \,\&\, \cdots \,\&\, [t/x]\sigma_n) \Rightarrow ([t/x]\mathcal{D}_V(\gamma_1) \,\&\, \cdots \,\&\, [t/x]\mathcal{D}_V(\gamma_2)))$$

which is equivalent with

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\sigma_1 \,\&\, \cdots \,\&\, [t/x]\sigma_n) \vdash \mathcal{R}([t/x]\mathcal{D}_V(\gamma_1)) \,\&\, \cdots \,\&\, \mathcal{R}([t/x]\mathcal{D}_V(\gamma_m))$$

Using lemma 17, we can deduce

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\sigma_1 \,\&\, \cdots \,\&\, [t/x]\sigma_n) \vdash \mathcal{D}([t/x]\gamma_1) \,\&\, \mathcal{D}([t/x]\gamma_2) \,\&\, \cdots \,\&\, \mathcal{D}([t/x]\gamma_m)$$

Applying lemma 9, we get the desired sequent.

• Finally, we derive the last sequent of the translation of the conclusion. From the last sequent of the translation of the premise, using similar manipulations as above, we can obtain

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\Sigma)\,, \mathcal{R}([t/x]\Gamma) \vdash \mathcal{R}([t/x]\alpha) \,\&\, \mathcal{R}([t/x]\mathcal{D}_V(\alpha))$$

Lemma 17 transforms this into

$$\mathcal{D}(t)\,, \mathcal{R}([t/x]\Sigma)\,, \mathcal{R}([t/x]\Gamma) \vdash \mathcal{R}([t/x]\alpha) \,\&\, \mathcal{D}([t/x]\alpha)$$

from which the desired sequent easily follows.

First equality rule

We have to prove that

$$\begin{cases} \vdash \mathcal{D}(\boldsymbol{\Delta}(t)) \\ \mathcal{R}(\boldsymbol{\Delta}(t)) \vdash \mathcal{R}(t = t) \,\&\, \mathcal{D}(t = t) \end{cases}$$

If t does not contain any ι-terms, then this reduces to the single sequent $\vdash \mathcal{R}(t = t) \,\&\, \mathcal{D}(t = t)$ which is equal to $\vdash t = t \,\&\, \forall x(x = x)$. This is trivial to deduce.

If t does contain ι-terms, call its top-level ι-terms $TLI(t) \equiv \iota x_{1\psi_1}(\varphi_1)$, $\iota x_{2\psi_2}(\varphi_2)$, …, $\iota x_{m\psi_m}(\varphi_m)$. We then have to prove that

$$\begin{cases} \vdash \mathcal{D}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m) \\ \mathcal{R}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m) \vdash \mathcal{R}(t = t) \,\&\, \mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) \,\&\, \cdots \,\&\, \mathcal{R}(\psi_m) \end{cases}$$

The first sequent is easy to deduce, since we have $\vdash \mathcal{D}(\psi_i)$ from the uniqueness conditions.

To deduce the second sequent, it is sufficient to deduce $\mathcal{R}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m) \vdash \mathcal{R}(t = t)$, which is

$$\begin{aligned}\mathcal{R}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m)&\\ \vdash \exists u_1 \ldots \exists u_m \exists v_1 \ldots \exists v_m \big(&\mathcal{R}([u_1/x]\varphi_1) \,\&\, \cdots \,\&\, \mathcal{R}([u_m/x]\varphi_m)\\ &\,\&\, \mathcal{R}([v_1/x]\varphi_1) \,\&\, \cdots \,\&\, \mathcal{R}([v_m/x]\varphi_m) \,\&\, q\big)\end{aligned}$$

where q is obtained from $t = t$ by replacing the top-level ι-terms in the left hand side of the equals sign by u_1, u_2, …, u_m and in the right hand side by v_1, v_2, …, v_m.

Applying lemma 13, this simplifies to

$$\begin{aligned}\mathcal{R}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m)&\\ \vdash \exists u_1 \ldots \exists u_m \exists v_1 \ldots \exists v_m \big(&\mathcal{R}([u_1/x]\varphi_1) \,\&\, \cdots \,\&\, \mathcal{R}([u_m/x]\varphi_m)\\ &\,\&\, u_1 = v_1 \,\&\, \cdots \,\&\, u_m = v_m \,\&\, q\big)\end{aligned}$$

from which we can deduce

$$\begin{aligned}\mathcal{R}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m)&\\ \vdash \exists u_1 \exists u_2 \ldots \exists u_m \big(&\mathcal{R}([u_1/x]\varphi_1) \,\&\, \ldots \,\&\, [u_1/v_1][u_2/v_2]\cdots[u_m/v_m]q\big)\end{aligned}$$

Now we notice that $[u_1/v_1][u_2/v_2]\cdots[u_m/v_m]q$ is of the form $t' = t'$ for some t'; hence the last sequent is equivalent with

$$\mathcal{R}(\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m) \vdash \exists u_1 \exists u_2 \ldots \exists u_m \big(\mathcal{R}([u_1/x]\varphi_1) \,\&\, \cdots \,\&\, \mathcal{R}([u_m/x]\varphi_m)\big)$$

which we can deduce easily from the uniqueness conditions.

Second equality rule

• First, we prove the rule for terms of the form $t \equiv y$. The rule then reduces to

$$\frac{\Sigma; \Gamma \vdash_\iota \alpha}{\Sigma; \Gamma, x = y \vdash_\iota [y/x]\alpha}$$

The first sequents of the translation of the premise and the conclusion are the same:

$$\vdash \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n)$$

For the second sequent, the premise yields

$$\mathcal{R}(\Sigma) \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \cdots \,\&\, \mathcal{D}(\gamma_m)$$

and we have to deduce

$$\mathcal{R}(\Sigma) \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \cdots \,\&\, \mathcal{D}(\gamma_m) \,\&\, \mathcal{D}(x = y)$$

Since $\mathcal{D}(x = y) \equiv \forall x(x = x)$, a validity, this is trivial.

The translation of the last sequent of the premise is equivalent with

$$\mathcal{R}(\Sigma)\,, \mathcal{R}(\Gamma) \vdash \mathcal{R}_y(\alpha) \,\&\, \mathcal{D}_y(\alpha)$$

Applying the equality rule of the Hermes calculus yields

$$\mathcal{R}(\Sigma)\,, \mathcal{R}(\Gamma)\,, x = y \vdash [y/x]\mathcal{R}_y(\alpha) \,\&\, [y/x]\mathcal{D}_y(\alpha)$$

Using lemmata 8 and 17 yields the translation of the last sequent of the conclusion.

• We already derived the translation of $\Sigma; \Gamma, x = y \vdash_\iota [y/x]\alpha$. If we choose y such that y does not occur in Γ and α, we can now apply the substitution rule of the PITFOL calculus to get the sequent

$$\mathbf{\Delta}(t)\,, \Sigma; \Gamma, x = t \vdash_\iota [t/y][y/x]\alpha$$

and because y does not occur in α, we have that $[t/y][y/x]\alpha \equiv [t/x]\alpha$.
This is the desired sequent up to the order of the context, which needs to be $\Sigma, \mathbf{\Delta}(t)$. To make this change of order, we only need to deduce

$$\vdash \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n \,\&\, \mathbf{\Delta}(t))$$

i.e., $\vdash \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n) \,\&\, (\mathcal{R}(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n) \Rightarrow \mathcal{D}(\mathbf{\Delta}(t)))$ But we already have $\vdash \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n)$: it is the first sequent of the translation of the premise. Moreover, $\mathcal{D}(\mathbf{\Delta}(t))$ is a validity according to lemma 10.

ι-rule

First, we note that the substitution $[\iota x_\psi(\varphi)/x]\widetilde{\varphi}$ is always defined.

Next, we remark that one easily proves that $\mathcal{R}(\alpha) \dashv\vdash \mathcal{R}(\widetilde{\alpha})$ and $\mathcal{D}(\alpha) \dashv\vdash \mathcal{D}(\widetilde{\alpha})$ using induction on the complexity of α.

We have to prove that

$$\begin{cases} \vdash \mathcal{D}(\psi) \\ \mathcal{R}(\psi) \vdash \mathcal{R}([\iota x_\psi(\varphi)/x]\widetilde{\varphi}) \,\&\, \mathcal{D}([\iota x_\psi(\varphi)/x]\widetilde{\varphi}) \end{cases}$$

We get the first sequent from the translation of the uniqueness condition for $\iota x_\psi(\varphi)$.

Using lemma 15, the second sequent can be rewritten as

$$\mathcal{R}(\psi) \vdash \forall x(\mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\widetilde{\varphi})) \,\&\, \mathcal{D}([\iota x_\psi(\varphi)/_x]\,\widetilde{\varphi})$$

Since from our remark above, $\vdash \mathcal{R}(\varphi) \Rightarrow \mathcal{R}(\widetilde{\varphi})$, this sequent is equivalent with

$$\mathcal{R}(\psi) \vdash \mathcal{D}([\iota x_\psi(\varphi)/_x]\,\widetilde{\varphi})$$

From the translation of the uniqueness condition for $\iota x_\psi(\varphi)$, we easily get $\mathcal{R}(\psi) \vdash \mathcal{D}(\exists x(\varphi))$, i.e., $\mathcal{R}(\psi) \vdash \forall x(\mathcal{D}(\varphi))$. Using our remark, this is equivalent with $\mathcal{R}(\psi) \vdash \forall x(\mathcal{D}(\widetilde{\varphi}))$. Combining this with corollary 18, we get $\mathcal{R}(\psi) \vdash \mathcal{D}([\iota x_\psi(\varphi)/_x]\,\widetilde{\varphi})$.

Note that it is necessary to rename the bound variables in case φ contains a variable symbol both bound and free. For example, the substitution

$$[\iota x(x = y \,\&\, \exists y(x = y))/_x](x = y \,\&\, \exists y(x = y))$$

is not defined, but

$$[\iota x(x = y \,\&\, \exists y(x = y))/_x](x = y \,\&\, \exists y'(x = y'))$$

poses no problems. In case φ does not contain a variable symbol both bound and free, it is not strictly necessary to rename the bound variables, but we chose to do so to keep the formulation of the rule more uniform.

UC rule

For all the other rules, it was sufficient to only consider the premises of the rule. The UC rule is an exception: here, we have to look at the whole proof leading to the premise $\Sigma; \Gamma \vdash_\iota \alpha$.

We will prove that we can derive the translation of $\psi \vdash_\iota \exists! x(\varphi)$ by induction on the length of the proof of $\Sigma; \Gamma \vdash_\iota \alpha$, where $\iota x_\psi(\varphi)$ is a ι-term occurring in Σ, Γ or α.

If the proof has length 1, it must consist of a single application of ass, eq without premises.

For the ass rule, $UC(\alpha)$ has to be empty (i.e., α must not contain any ι-terms), and Σ has to be empty. The conclusion is thus $\alpha \vdash_\iota \alpha$ and there are no ι-terms present.

For the eq rule we can use a similar argument.

For the induction step, suppose we have a proof of length $n > 1$ of $\Sigma; \Gamma \vdash_\iota \alpha$. By induction, we can assume that we can derive the translation of the uniqueness condition of any ι-term in the first $n-1$ sequents of the proof,

and more in particular, of the premises of the rule used to obtain the last formula of the proof. We will show that using these uniqueness conditions, we can derive the translation of $\psi \vdash_\iota \exists! x(\varphi)$. We consider the rule used to obtain the last sequent of the proof.

- For the ass rule, the only new formula in the last sequent of the proof is α, and this rule requires $UC(\alpha)$ as a premise. Analogous for contra and eq.
- The &-elim, rem, contra, $\forall$-intro, $\forall$-elim, defAnt, defCons, toCtxt and fromCtxt rules do not introduce any new ι-terms in the last sequent of the proof.
- For the subst and eqSubst rule, we have to show that given the translation of the uniqueness condition of a ι-term $\iota y_\Psi(\Phi)$ occurring in the last premise of the rule, and the translation of $UC(t)$, we can derive the translation of the uniqueness condition of $[t/x]\,\iota y_\Psi(\Phi)$, which is precisely what we proved in lemma 19.
- For the iota rule, we have to show that we can derive the translations of $UC(\psi)$ and $UC([\iota x_\psi(\varphi)/x]\,\widetilde{\varphi})$. By induction, we can derive the translations of $UC(\psi)$ and $UC(\varphi)$. It is not difficult to show that from the latter, we can derive the translations of $UC(\widetilde{\varphi})$. Lemma 19 then yields the translations of $UC([\iota x_\psi(\varphi)/x]\,\widetilde{\varphi})$.
- For the UC rule itself, by induction, we can derive the translations of $UC(\iota x_\psi(\varphi))$.

3.6.7 Δ-rules

defAnt

We have to show that

$$\begin{cases} \vdash \mathcal{D}(\sigma_1 \,\&\, \sigma_2 \,\&\, \ldots \sigma_n) \\ \mathcal{R}(\Sigma) \vdash \mathcal{R}(\boldsymbol{\Delta}(\alpha)) \,\&\, \mathcal{D}(\boldsymbol{\Delta}(\alpha)) \end{cases}$$

The first sequent is identical to the first sequent of the translation of the premise.
The second sequent of the translation of the premise is

$$\mathcal{R}(\Sigma) \vdash \mathcal{D}(\gamma_1) \,\&\, \mathcal{D}(\gamma_2) \,\&\, \cdots \,\&\, \mathcal{D}(\gamma_m) \,\&\, \mathcal{D}(\alpha)$$

Using again lemmas 9 and 10, we quickly get the second sequent of the translation of the conclusion.

defCons

The first and second sequents of the translation of the conclusion are identical to those of the premise, so we only have to handle the third sequent of the translations.
The premise gives us

$$\mathcal{R}(\Sigma)\,, \mathcal{R}(\Gamma) \vdash \mathcal{R}(\alpha)\,\&\,\mathcal{D}(\alpha)$$

and we have to prove

$$\mathcal{R}(\Sigma)\,, \mathcal{R}(\Gamma) \vdash \mathcal{R}(\mathbf{\Delta}(\alpha))\,\&\,\mathcal{D}(\mathbf{\Delta}(\alpha))$$

which is not difficult, again using lemmas 9 and 10.

3.6.8 Contextual rules

These rules are proved similarly.

3.7 Soundness

We first prove a connection between the semantics and the operations $\mathcal{R}$ and $\mathcal{D}$.

Lemma 20 *Given a formula α of the* PITFOL *calculus and an interpretation $\mathcal{I}$. Suppose that the uniqueness conditions of α are valid in $\mathcal{I}$. Then*

$$\begin{aligned}
\alpha \text{ is valid in } \mathcal{I} &\Leftrightarrow \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha) \text{ is valid in } \mathcal{I} \text{ (in the Hermes calculus)}\\
\alpha \text{ is invalid in } \mathcal{I} &\Leftrightarrow \mathcal{D}(\alpha)\,\&\,\neg\mathcal{R}(\alpha) \text{ is valid in } \mathcal{I} \text{ (in the Hermes calculus)}\\
\alpha \text{ is undefined in } \mathcal{I} &\Leftrightarrow \neg\mathcal{D}(\alpha) \text{ is valid in } \mathcal{I} \text{ (in the Hermes calculus)}
\end{aligned}$$

Proof.
We prove this by induction on the complexity of α.

- If α is atomic and does not contain any ι-terms, then $\mathcal{R}(\alpha) \equiv \alpha$ and $\mathcal{D}(\alpha)$ is a validity. Also note that in this case, the interpretation of α as a formula of the PITFOL calculus is the same as its interpretation as a formula of the Hermes calculus. Finally, it is clear from the definition of an interpretation of a formula that α can never be undefined. From these observations, this case is easily proved.

- Suppose α is atomic and contains ι-terms. Hence $\alpha \equiv p(t_1, \ldots, t_n)$, where we treat the case $\alpha \equiv t_1 = t_2$ analogously.

– Suppose α is valid in $\mathcal{I}$. Call its top-level ι-terms $TLI(\alpha) \equiv \iota x_{1\psi_1}(\varphi_1), \iota x_{2\psi_2}(\varphi_2), \ldots, \iota x_{m\psi_m}(\varphi_m)$ We then have to prove, with the usual notation for q, that

$$\begin{aligned}&\mathcal{R}(\psi_1) \,\&\, \ldots \,\&\, \mathcal{R}(\psi_m)\\ &\qquad \&\, \exists u_1 \ldots \exists u_m \,(\mathcal{R}([u_1/x_1]\varphi_1) \,\&\, \cdots \,\&\, \mathcal{R}([u_m/x_m]\varphi_m) \,\&\, q)\end{aligned}$$

is valid in $\mathcal{I}$.

It is easy to see that all $\iota x_{i\psi_i}(\varphi_i)$ must be defined in $\mathcal{I}$, so ψ_i must be valid in $\mathcal{I}$ and by induction, $\mathcal{D}(\psi_i) \,\&\, \mathcal{R}(\psi_i)$ must be valid too.

Since the uniqueness conditions are valid in $\mathcal{I}$, for each top-level ι-term $\iota x_{i\psi_i}(\varphi_i)$, there exists only one a_i such that φ_i is valid in $\mathcal{I}^{a_i}_{x_i}$, from which it easily follows that $[u_i/x_i]\varphi_i$ is valid in $\mathcal{I}^{a_1 a_2 \ldots a_m}_{u_1 u_2 \ldots u_m}$. Hence, by induction, $\mathcal{I}^{a_1 a_2 \ldots a_m}_{u_1 u_2 \ldots u_m}[u_i/x_i]\mathcal{R}(\varphi_i)$ is valid too.

It is easy to see that $\mathcal{I}(\alpha) \equiv \mathcal{I}^{\mathcal{I}\iota x_{1\psi_1}(\varphi_1) \ldots \mathcal{I}\iota x_{m\psi_m}(\varphi_m)}_{u_1 \quad\ldots\quad u_m} q$. Because of the definition of interpretation of ι-terms, we see that $\mathcal{I}(\iota x_{i\psi_i}(\varphi_i)) = a_i$, so $\mathcal{I}^{a_1 a_2 \cdots a_m}_{u_1 u_2 \cdots u_m}(q)$ also holds, hence

$$\exists u_1 \ldots \exists u_m \,(\mathcal{R}([u_1/x_1]\varphi_1) \,\&\, \mathcal{R}([u_2/x_2]\varphi_2) \,\&\, \cdots \,\&\, \mathcal{R}([u_m/x_m]\varphi_m) \,\&\, q)$$

is valid too in $\mathcal{I}$.

– If α is invalid in $\mathcal{I}$, then we have to prove that

$$\begin{aligned}&\mathcal{R}(\psi_1) \,\&\, \ldots \,\&\, \mathcal{R}(\psi_m)\\ &\qquad \&\, \neg\exists u_1 \ldots \exists u_m \,(\mathcal{R}([u_1/x_1]\varphi_1) \,\&\, \cdots \,\&\, \mathcal{R}([u_m/x_m]\varphi_m) \,\&\, q)\end{aligned}$$

is valid in $\mathcal{I}$. Analogously as above, the $\mathcal{R}(\psi_i)$ are all valid in $\mathcal{I}$. Since the uniqueness conditions hold, we have that the interpretation of $\neg\exists u_1 \ldots \exists u_m \,(\mathcal{R}([u_1/x_1]\varphi_1) \,\&\, \cdots \,\&\, \mathcal{R}([u_m/x_m]\varphi_m) \,\&\, q)$ is the same as that of

$$\exists u_1 \ldots \exists u_m \,(\mathcal{R}([u_1/x_1]\varphi_1) \,\&\, \cdots \,\&\, \mathcal{R}([u_m/x_m]\varphi_m) \,\&\, \neg q)\,,$$

which we can show to be valid analogously as above.

– If α is undefined in $\mathcal{I}$, then at least one $\mathcal{I}(t_i)$ is undefined, hence at least one $\mathcal{I}(\iota x_{i\psi_i}(\varphi_i))$ is undefined, which means that either ψ_i is invalid in $\mathcal{I}$ (and hence, by induction, $\mathcal{D}(\alpha) \,\&\, \neg\mathcal{R}(\psi_i)$ is valid in $\mathcal{I}$) or there are no or multiple a such that φ_i is valid in $\mathcal{I}^a_x$. The latter case is impossible since we supposed that the uniqueness condition for $\mathcal{I}(\iota x_{i\psi_i}(\varphi_i))$ is valid.

We also have to prove the converse: if $\mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha)$ is valid in $\mathcal{I}$, then α must be valid in $\mathcal{I}$. This follows easily from the fact that exactly one of $\mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha)$, $\mathcal{D}(\alpha)\,\&\,\neg\mathcal{R}(\alpha)$ and $\neg\mathcal{D}(\alpha)$ must be valid. Indeed, if $\mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha)$ is valid, then $\mathcal{D}(\alpha)\,\&\,\neg\mathcal{R}(\alpha)$ cannot be valid and hence α cannot be invalid; also, $\neg\mathcal{D}(\alpha)$ cannot be valid and hence α cannot be undefined. The only remaining possibility is that α is valid.
The other two cases follow analogously.

- The cases where α is not atomic are proved easily.

□

Theorem 21 (Soundness of the pitfol calculus) *If* $\Gamma \vdash_\iota \alpha$, *then also* $\Gamma \models_\iota \alpha$.

Proof.

We know that if we have a PITFOL proof of $\Gamma \vdash_\iota \alpha$, we can translate this into a proof in the Hermes calculus, i.e., we obtain a proof of the sequents

$$\begin{cases} \vdash \mathcal{D}(\gamma_1)\,\&\,\mathcal{D}(\gamma_2)\,\&\,\ldots\,\&\,\mathcal{D}(\gamma_m) \\ \mathcal{R}(\gamma_1)\,,\mathcal{R}(\gamma_2)\,,\ldots,\mathcal{R}(\gamma_m) \vdash \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha) \end{cases}$$

We have to prove $\Gamma \models_\iota \alpha$, which means that

- For any interpretation $\mathcal{I}$, if γ_1, γ_2, …, γ_m are valid in $\mathcal{I}$, then α is valid too in $\mathcal{I}$. Using the previous lemma, this is equivalent with proving in the Hermes calculus that if $\mathcal{D}(\gamma_1)\,\&\,\mathcal{R}(\gamma_1)\,\&\,\cdots\,\&\,\mathcal{D}(\gamma_m)\,\&\,\mathcal{R}(\gamma_m)$ is valid, then $\mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha)$ is valid, i.e.,

  $$\mathcal{D}(\gamma_1)\,,\mathcal{R}(\gamma_1)\,,\ldots,\mathcal{D}(\gamma_m)\,,\mathcal{R}(\gamma_m) \models \mathcal{R}(\alpha)\,\&\,\mathcal{D}(\alpha)$$

  Because of the soundness of the Hermes calculus, this amounts to

  $$\mathcal{D}(\gamma_1)\,,\mathcal{R}(\gamma_1)\,,\ldots,\mathcal{D}(\gamma_m)\,,\mathcal{R}(\gamma_m) \vdash \mathcal{R}(\alpha)\,\&\,\mathcal{D}(\alpha)$$

  which follows easily from the second sequent of the translation given.

- There exist no interpretations in which a γ_i is undefined. Hence, in each interpretation, γ_i must be valid or invalid, which we can express using the previous lemma as $\models (\mathcal{D}(\gamma_i)\,\&\,\mathcal{R}(\gamma_i)) \vee (\mathcal{D}(\gamma_i)\,\&\,\neg\mathcal{R}(\gamma_i))$ which is equivalent with $\models \mathcal{D}(\gamma_i)$. Again, because of the completeness of the Hermes calculus, we have to prove that $\vdash \mathcal{D}(\gamma_i)$, which follows easily from the first sequent of the translation.

□

3.8 Derived rules

In this section, we will develop some derived rules. Most of them are adaptations of the derived rules in [Hermes 1973].

We start with the self-assertion rule:

$$\begin{array}{rr} \boxed{\text{SeAs}} & \\ \Sigma;\Gamma,\neg\alpha \vdash_\iota \alpha & \text{prem} \\ \Sigma;\vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{defAnt} \\ \Sigma;\alpha \vdash_\iota \alpha & \text{ass} \\ \Sigma;\Gamma \vdash_\iota \alpha & \text{rem} \end{array}$$

We note the name of the rule, SeAs, in a box at the top. For each sequent, we supply a justification in the rightmost column. The first line(s) are the premise(s) of the rule, if any (hence the justification 'prem' in the first line) and the last line is its conclusion.

Note that, strictly speaking, the assumption introduction rule needs $UC(\alpha)$ as premise(s), which we have not explicitly derived. However, the formula α occurs in the premise and the UC rule immediately yields all uniqueness conditions required. Hence, we will silently ignore such $UC(\ldots)$ premises in the future.

Finally, note that strictly speaking, we need to supply another proof when $\boldsymbol{\Delta}(\alpha) \equiv \top$, since in that case we cannot apply defAnt. In most cases, it is easy to supply an alternative derivation for these cases, so we will not explicitly derive them. In the SeAs rule, it suffices to drop the defAnt line from the proof:

$$\begin{array}{rr} \boxed{\text{SeAs}} & \\ \Sigma;\Gamma,\neg\alpha \vdash_\iota \alpha & \text{prem} \\ \alpha \vdash_\iota \alpha & \text{ass} \\ \Sigma;\Gamma \vdash_\iota \alpha & \text{rem} \end{array}$$

As stated before, by convention, if a premise of a rule would have $\top$ as consequent, that premise may be dropped. For example, the NN1 rule (which will follow shortly) has $\Sigma;\vdash_\iota \boldsymbol{\Delta}(\alpha)$ as its single premise; if we want to apply this rule to a formula α for which $\boldsymbol{\Delta}(\alpha)$ is $\top$, we can immediately apply the rule without any premises.

From now on, we are allowed to use a new rule

$$\frac{\begin{array}{r}\boxed{\text{SeAs}}\\ \Sigma;\Gamma,\neg\alpha \vdash_\iota \alpha\end{array}}{\Sigma;\Gamma \vdash_\iota \alpha}$$

in our proofs. It is easy to see that this does not change the power of the logic, since one can simply replace each use of this new rule with its derivation given above.

A similar rule is the self-denial rule:

$$\boxed{\text{SeDe}}$$

$$\begin{array}{rr}
\Sigma; \Gamma, \alpha \vdash_\iota \neg\alpha & \text{prem} \\
\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{defAnt} \\
\Sigma; \neg\alpha \vdash_\iota \neg\alpha & \text{ass} \\
\Sigma; \Gamma \vdash_\iota \neg\alpha & \text{rem}
\end{array}$$

A variant of the assumption rule when $\boldsymbol{\Delta}(\alpha) \equiv \top$, where we set $\Gamma \equiv \gamma_1, \gamma_2, \ldots, \gamma_n$:

$$\boxed{\text{Ass2}}$$

$$\begin{array}{rr}
\Sigma; \Gamma \vdash_\iota \beta & \text{prem} \\
\vdash_\iota x = x & \text{eq} \\
\Sigma; \Gamma \vdash_\iota \beta \,\&\, x = x & \&\text{-intro} \\
\Sigma; \Gamma \vdash_\iota x = x & \&\text{-elim}
\end{array}$$

When $\boldsymbol{\Delta}(\gamma_1) \not\equiv \top$

$$\begin{array}{rr}
\Sigma; \vdash_\iota \boldsymbol{\Delta}(\gamma_1) & \text{defAnt} \\
\Sigma; \neg\gamma_1 \vdash_\iota \neg\gamma_1 & \text{ass} \\
\Sigma; \neg\gamma_1 \vdash_\iota x = x \,\&\, \neg\gamma_1 & \&\text{-intro} \\
\Sigma; \neg\gamma_1 \vdash_\iota x = x & \&\text{-elim}
\end{array}$$

When $\boldsymbol{\Delta}(\gamma_1) \equiv \top$

$$\begin{array}{rr}
\neg\gamma_1 \vdash_\iota \neg\gamma_1 & \text{ass} \\
\neg\gamma_1 \vdash_\iota x = x \,\&\, \neg\gamma_1 & \&\text{-intro} \\
\neg\gamma_1 \vdash_\iota x = x & \&\text{-elim}
\end{array}$$

$$\begin{array}{rr}
\Sigma; \gamma_2, \ldots, \gamma_n \vdash_\iota x = x & \text{rem} \\
\vdots & \\
\Sigma; \gamma_n \vdash_\iota x = x & \text{rem}
\end{array}$$

When $\boldsymbol{\Delta}(\gamma_n) \not\equiv \top$

$$\begin{array}{rr}
\Sigma; \vdash_\iota \boldsymbol{\Delta}(\gamma_n) & \text{defAnt} \\
\Sigma; \neg\gamma_n \vdash_\iota \neg\gamma_n & \text{ass} \\
\Sigma; \neg\gamma_n \vdash_\iota x = x \,\&\, \neg\gamma_n & \&\text{-intro} \\
\Sigma; \neg\gamma_n \vdash_\iota x = x & \&\text{-elim}
\end{array}$$

When $\boldsymbol{\Delta}(\gamma_n) \equiv \top$

$$\begin{array}{rr}
\neg\gamma_n \vdash_\iota \neg\gamma_n & \text{ass} \\
\neg\gamma_n \vdash_\iota x = x \,\&\, \neg\gamma_n & \&\text{-intro} \\
\neg\gamma_n \vdash_\iota x = x & \&\text{-elim}
\end{array}$$

$$\begin{array}{rr}
\Sigma; \vdash_\iota x = x & \text{rem} \\
\alpha \vdash_\iota \alpha & \text{ass} \\
\Sigma; \alpha \vdash_\iota x = x \,\&\, \alpha & \&\text{-intro} \\
\Sigma; \alpha \vdash_\iota \alpha & \&\text{-elim}
\end{array}$$

When Γ is empty, we can use

Ass2	
$\Sigma; \vdash_\iota \beta$	prem
$\alpha \vdash_\iota \alpha$	ass
$\Sigma; \alpha \vdash_\iota \alpha \,\&\, \beta$	&-intro
$\Sigma; \alpha \vdash_\iota \alpha$	&-elim

We see that in case $\mathbf{\Delta}(\alpha)$ is $\top$, we can add a context Σ to the resulting sequent using the Ass2 rule, just as in the assumption rule when $\mathbf{\Delta}(\alpha)$ is not $\top$. However, one is not entirely free in the choice of the context Σ: it has to have been already used before as a context. Indeed, if we were allowed to choose Σ at will, we would be able to obtain unsound sequents such as $\iota y_{x\neq 0}(x \cdot y = 1) \neq 0; \alpha \vdash_\iota \alpha$.

Next, we prove two weakening rules for contexts:

WeakCtxtL	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \Delta \vdash_\iota \beta$	prem
$\Sigma_1; x = x \vdash_\iota x = x$	Ass2
$\vdash_\iota x = x$	eq
$\neg(x = x) \vdash_\iota \neg(x = x)$	ass
$\neg(x = x) \vdash_\iota x = x$	contra
$\Sigma_1; \vdash_\iota x = x$	rem
$\Sigma_1, \Sigma_2; \Delta \vdash_\iota x = x \,\&\, \beta$	&-intro
$\Sigma_1, \Sigma_2; \Delta \vdash_\iota \beta$	&-elim

WeakCtxtR	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \Delta \vdash_\iota \beta$	prem
$\Sigma_1; x = x \vdash_\iota x = x$	Ass2
$\vdash_\iota x = x$	eq
$\neg(x = x) \vdash_\iota \neg(x = x)$	ass
$\neg(x = x) \vdash_\iota x = x$	contra
$\Sigma_1; \vdash_\iota x = x$	rem
$\Sigma_2, \Sigma_1; \Delta \vdash_\iota \beta \,\&\, x = x$	&-intro
$\Sigma_2, \Sigma_1; \Delta \vdash_\iota \beta$	&-elim

If there is a $\gamma \in \Gamma$ for which $\mathbf{\Delta}(\gamma) \not\equiv \top$ it is possible to use a shorter derivation:

WeakCtxtL	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \Delta \vdash_\iota \beta$	prem
$\Sigma_1; \vdash_\iota \mathbf{\Delta}(\gamma)$	defAnt
$\Sigma_1, \Sigma_2; \Delta \vdash_\iota \mathbf{\Delta}(\gamma) \,\&\, \beta$	&-intro
$\Sigma_1, \Sigma_2; \Delta \vdash_\iota \beta$	&-elim

WeakCtxtR	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \Delta \vdash_\iota \beta$	prem
$\Sigma_1; \vdash_\iota \mathbf{\Delta}(\gamma)$	defAnt
$\Sigma_2, \Sigma_1; \Delta \vdash_\iota \beta \,\&\, \mathbf{\Delta}(\gamma)$	&-intro
$\Sigma_2, \Sigma_1; \Delta \vdash_\iota \beta$	&-elim

We already indicated that the choice of Σ_1, Σ_2 as context of a conclusion of a rule was arbitrary in the sense that we could just as well have built a corresponding rule with Σ_2, Σ_1 as context instead. For example, with the &-intro rule, we can transform $\Sigma_1, \Sigma_2; \Gamma, \Delta \vdash_\iota \alpha \& \beta$ into $\Sigma_2, \Sigma_1; \Gamma, \Delta \vdash_\iota \alpha \& \beta$ using the WeakCtxtL rule together with the sequent $\Sigma_1; \Gamma \vdash_\iota \alpha$. (Note that in the resulting sequent, we would have $\Sigma_2, \Sigma_1, \Sigma_2$ as context, but since we agreed that we are allowed to only keep the first occurrence of formulae that occur more than once in contexts, this reduces to Σ_2, Σ_1.)

Double negation rules:

NN1	
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	prem
$\Sigma; \alpha \vdash_\iota \alpha$	ass
$\Sigma; \neg\alpha \vdash_\iota \neg\alpha$	ass
$\Sigma; \alpha, \neg\alpha \vdash_\iota \neg\neg\alpha$	contra
$\Sigma; \alpha \vdash_\iota \neg\neg\alpha$	SeDe

NN2	
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	prem
$\Sigma; \neg\alpha \vdash_\iota \neg\alpha$	ass
$\Sigma; \neg\neg\alpha \vdash_\iota \neg\neg\alpha$	ass
$\Sigma; \neg\alpha, \neg\neg\alpha \vdash_\iota \alpha$	contra
$\Sigma; \neg\neg\alpha \vdash_\iota \alpha$	SeAs

If $\boldsymbol{\Delta}(\alpha) \equiv \top$ and Σ is not empty, we need $\Sigma; \vdash_\iota \beta$ as a premise instead:

NN1	
$\Sigma; \vdash_\iota \beta$	prem
$\alpha \vdash_\iota \alpha$	ass
$\neg\alpha \vdash_\iota \neg\alpha$	ass
$\alpha, \neg\alpha \vdash_\iota \neg\neg\alpha$	contra
$\alpha \vdash_\iota \neg\neg\alpha$	SeDe
$\Sigma; \alpha \vdash_\iota \neg\neg\alpha \,\&\, \beta$	&-intro
$\Sigma; \alpha \vdash_\iota \neg\neg\alpha$	&-elim

NN2	
$\Sigma; \vdash_\iota \beta$	prem
$\neg\alpha \vdash_\iota \neg\alpha$	ass
$\neg\neg\alpha \vdash_\iota \neg\neg\alpha$	ass
$\neg\alpha, \neg\neg\alpha \vdash_\iota \alpha$	contra
$\neg\neg\alpha \vdash_\iota \alpha$	SeAs
$\Sigma; \neg\neg\alpha \vdash_\iota \alpha \,\&\, \beta$	&-intro
$\Sigma; \neg\neg\alpha \vdash_\iota \alpha$	&-elim

DefCtxt	
$\Sigma, \sigma; \Gamma \vdash_\iota \alpha$	prem
$\Sigma; \sigma \,\&\, \Gamma \vdash_\iota \alpha$	fromCtxt
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\sigma \,\&\, \gamma_1)$	fromCtxt (*)
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\sigma)$	&-elim

(*) If Γ is empty, then we immediately get the next line of the proof.

If $\boldsymbol{\Delta}(\alpha)$ is not $\top$:

Cut	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \Delta, \alpha \vdash_\iota \beta$	prem
$\Sigma_2; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	defAnt
$\Sigma_2; \neg\alpha \vdash_\iota \neg\alpha$	ass
$\Sigma_1, \Sigma_2; \Gamma, \neg\alpha \vdash_\iota \beta$	contra
$\Sigma_2, \Sigma_1; \Gamma, \Delta \vdash_\iota \beta$	rem
$\Sigma_1, \Sigma_2; \Gamma, \Delta \vdash_\iota \beta$	WeakCtxtL

If $\boldsymbol{\Delta}(\alpha) \equiv \top$:

Cut	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \Delta, \alpha \vdash_\iota \beta$	prem
$\neg\alpha \vdash_\iota \neg\alpha$	ass
$\Sigma_1; \Gamma, \neg\alpha \vdash_\iota \beta$	contra
$\Sigma_2, \Sigma_1; \Gamma, \Delta \vdash_\iota \beta$	rem
$\Sigma_1, \Sigma_2; \Gamma, \Delta \vdash_\iota \beta$	WeakCtxtL

Variant of fromCtxt:
If $\Gamma \equiv \gamma_1, \gamma_2, \ldots, \gamma_n$ is not empty:

FromCtxt2	
$\Sigma_1; \Gamma, \Delta \vdash_\iota \alpha$	prem
$\Sigma_2, \sigma; \Gamma \vdash_\iota \beta$	prem
$\Sigma_2; \sigma \,\&\, \Gamma \vdash_\iota \beta$	fromCtxt

$$\dfrac{\dfrac{\text{When } \mathbf{\Delta}(\sigma) \not\equiv \top}{\Sigma_2; \vdash_\iota \mathbf{\Delta}(\sigma \,\&\, \gamma_1)}\ \text{defAnt}}{\Sigma_2; \vdash_\iota \mathbf{\Delta}(\sigma)}\ \&\text{-elim}}{\Sigma_2; \sigma \vdash_\iota \sigma}\ \text{ass} \qquad \dfrac{\text{When } \mathbf{\Delta}(\sigma) \equiv \top}{\Sigma_2; \sigma \vdash_\iota \sigma}\ \text{Ass2}$$

$$\dfrac{\dfrac{\text{When } \mathbf{\Delta}(\gamma_1) \not\equiv \top}{\Sigma_1; \vdash_\iota \mathbf{\Delta}(\gamma_1)}\ \text{defAnt}}{\Sigma_1; \gamma_1 \vdash_\iota \gamma_1}\ \text{ass} \qquad \dfrac{\text{When } \mathbf{\Delta}(\gamma_1) \equiv \top}{\Sigma_1; \gamma_1 \vdash_\iota \gamma_1}\ \text{Ass2}$$

$$\dfrac{\dfrac{\Sigma_2, \Sigma_1; \gamma_1, \sigma \vdash_\iota \sigma \,\&\, \gamma_1}{\Sigma_2, \Sigma_1; \sigma, \gamma_1, \sigma \,\&\, \gamma_2, \sigma \,\&\, \gamma_3, \ldots \vdash_\iota \beta}\ \text{Cut}}{}\ \&\text{-intro}$$

$$\dfrac{\dfrac{\text{When } \mathbf{\Delta}(\gamma_2) \not\equiv \top}{\Sigma_1; \vdash_\iota \mathbf{\Delta}(\gamma_2)}\ \text{defAnt}}{\Sigma_1; \gamma_2 \vdash_\iota \gamma_2}\ \text{ass} \qquad \dfrac{\text{When } \mathbf{\Delta}(\gamma_2) \equiv \top}{\Sigma_1; \gamma_2 \vdash_\iota \gamma_2}\ \text{Ass2}$$

$$\dfrac{\dfrac{\Sigma_2, \Sigma_1; \gamma_2, \sigma \vdash_\iota \sigma \,\&\, \gamma_2}{\Sigma_2, \Sigma_1; \sigma, \gamma_1, \gamma_2, \sigma \,\&\, \gamma_3, \ldots \vdash_\iota \beta}\ \text{Cut}}{}\ \&\text{-intro}$$

$$\dfrac{\dfrac{\vdots}{\Sigma_2, \Sigma_1; \sigma, \Gamma \vdash_\iota \beta}\ \text{Cut}}{\Sigma_1, \Sigma_2; \sigma, \Gamma \vdash_\iota \beta}\ \text{WeakCtxtL}$$

If Γ is empty then we have to use this variant:

$$\dfrac{\dfrac{\boxed{\text{FromCtxt2}}}{\Sigma_1; \Delta \vdash_\iota \alpha}\ \text{prem}}{\Sigma_2, \sigma; \vdash_\iota \beta}\ \text{prem}}{\Sigma_2; \sigma \vdash_\iota \beta}\ \text{fromCtxt}}{\Sigma_1, \Sigma_2; \sigma \vdash_\iota \beta}\ \text{WeakCtxtL}$$

Semantically, we can interpret this rule as follows. Without the first premise, we can only derive

$$\Sigma_2; \sigma \,\&\, \Gamma \vdash_\iota \beta$$

If we want to get rid of $\sigma \,\&\, \Gamma$ and have σ, Γ instead, we need the first premise, which asserts that Γ is defined in the context of Σ_1. We already have that σ is defined in the context of Σ_2; the FromCtxt2 rule allows us to derive the expected

$$\Sigma_1, \Sigma_2; \sigma, \Gamma \vdash_\iota \beta$$

Repeated application of FromCtxt2 allows us to retrieve a list of formulae Σ from the context:

$$\boxed{\text{FromCtxt2*}}$$

$$\begin{array}{rl}
\Sigma_1; \Sigma, \Gamma, \Delta \vdash_\iota \alpha & \text{prem} \\
\Sigma_2, \Sigma; \Gamma \vdash_\iota \beta & \text{prem} \\
\Sigma_1, \Sigma_2, \sigma_1, \sigma_2, \ldots, \sigma_{n-1}; \sigma_n, \Gamma \vdash_\iota \beta & \text{FromCtxt2} \\
\Sigma_1, \Sigma_2, \sigma_1, \sigma_2, \ldots, \sigma_{n-2}; \sigma_{n-1}, \sigma_n, \Gamma \vdash_\iota \beta & \text{FromCtxt2} \\
\vdots & \\
\Sigma_1, \Sigma_2; \Sigma, \Gamma \vdash_\iota \beta & \text{FromCtxt2}
\end{array}$$

We illustrate a contextual version of a derived rule. In contrast with the approach with contextless sequents, we can actually reuse the proof of the cut rule. Other rules can be given contextual versions in an analogous manner.

$$\boxed{\text{Cut2}}$$

$$\begin{array}{rl}
\Sigma_1; \Gamma \vdash_\iota \alpha & \text{prem} \\
\Sigma_2; \Delta, \gamma \,\&\, \alpha \vdash_\iota \beta & \text{prem} \\
\Sigma_2, \gamma; \Delta, \alpha \vdash_\iota \beta & \text{toCtxt} \\
\Sigma_1, \Sigma_2, \gamma; \Gamma, \Delta \vdash_\iota \beta & \text{Cut} \\
\Sigma_1, \Sigma_2; \Gamma, \Delta, \gamma \,\&\, \alpha \vdash_\iota \alpha \,\&\, \beta & \text{\&-intro} \\
\Sigma_1, \Sigma_2; \gamma, \Gamma, \Delta \vdash_\iota \beta & \text{FromCtxt2}
\end{array}$$

Another variant:

$$\boxed{\text{Cut3}}$$

$$\begin{array}{rl}
\Sigma_1; \Gamma \vdash_\iota \alpha & \text{prem} \\
\Sigma_2, \mathbf{\Delta}(\alpha)\,; \alpha \vdash_\iota \beta & \text{prem} \\
\Sigma_1; \Gamma \vdash_\iota \mathbf{\Delta}(\alpha) & \text{defCons} \\
\Sigma_1; \Gamma \vdash_\iota \mathbf{\Delta}(\alpha) \,\&\, \alpha & \text{\&-intro} \\
\Sigma_2; \mathbf{\Delta}(\alpha) \,\&\, \alpha \vdash_\iota \beta & \text{fromCtxt} \\
\Sigma_1, \Sigma_2; \Gamma \vdash_\iota \beta & \text{Cut}
\end{array}$$

Contraposition rules:

$$\boxed{\text{CoPo1}}$$

$$\begin{array}{rl}
\Sigma_1; \Gamma, \alpha \vdash_\iota \beta & \text{prem} \\
\Sigma_2; \vdash_\iota \mathbf{\Delta}(\beta) & \text{prem} \\
\Sigma_1; \vdash_\iota \mathbf{\Delta}(\alpha) & \text{defAnt} \\
\Sigma_2; \neg\beta \vdash_\iota \neg\beta & \text{ass} \\
\Sigma_1, \Sigma_2; \Gamma, \alpha, \neg\beta \vdash_\iota \neg\alpha & \text{contra} \\
\Sigma_1, \Sigma_2; \Gamma, \neg\beta \vdash_\iota \neg\alpha & \text{SeDe}
\end{array}$$

$$\boxed{\text{CoPo2}}$$

$$\begin{array}{rl}
\Sigma_1; \Gamma, \alpha \vdash_\iota \neg\beta & \text{prem} \\
\Sigma_2; \vdash_\iota \mathbf{\Delta}(\beta) & \text{prem} \\
\Sigma_1; \vdash_\iota \mathbf{\Delta}(\alpha) & \text{defAnt} \\
\Sigma_2; \beta \vdash_\iota \beta & \text{ass} \\
\Sigma_1, \Sigma_2; \Gamma, \alpha, \beta \vdash_\iota \neg\alpha & \text{contra} \\
\Sigma_1, \Sigma_2; \Gamma, \beta \vdash_\iota \neg\alpha & \text{SeDe}
\end{array}$$

CoPo3	
$\Sigma_1; \Gamma, \neg\alpha \vdash_\iota \beta$	prem
$\Sigma_2; \vdash_\iota \boldsymbol{\Delta}(\beta)$	prem
$\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	defAnt
$\Sigma_2; \neg\beta \vdash_\iota \neg\beta$	ass
$\Sigma_1, \Sigma_2; \Gamma, \neg\alpha, \neg\beta \vdash_\iota \alpha$	contra
$\Sigma_1, \Sigma_2; \Gamma, \neg\beta \vdash_\iota \alpha$	SeAs

CoPo4	
$\Sigma_1; \Gamma, \neg\alpha \vdash_\iota \neg\beta$	prem
$\Sigma_2; \vdash_\iota \boldsymbol{\Delta}(\beta)$	prem
$\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	defAnt
$\Sigma_2; \beta \vdash_\iota \beta$	ass
$\Sigma_1, \Sigma_2; \Gamma, \neg\alpha, \beta \vdash_\iota \alpha$	contra
$\Sigma_1, \Sigma_2; \Gamma, \beta \vdash_\iota \alpha$	SeAs

Ex contradictione quodlibet:

XQ1	
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	prem
$\Sigma; \alpha \vdash_\iota \alpha$	ass
$\Sigma; \neg\alpha \vdash_\iota \neg\alpha$	ass
$\Sigma; \alpha, \neg\alpha \vdash_\iota \beta$	contra

Weakening rules:

Weak*	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \Delta \vdash_\iota \beta$	prem
$\Sigma_1, \Sigma_2; \Gamma, \Delta \vdash_\iota \alpha \,\&\, \beta$	&-intro
$\Sigma_1, \Sigma_2; \Gamma, \Delta \vdash_\iota \alpha$	&-elim

Weak	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \vdash_\iota \boldsymbol{\Delta}(\beta)$	prem
$\Sigma_2; \beta \vdash_\iota \beta$	ass
$\Sigma_1, \Sigma_2; \Gamma, \beta \vdash_\iota \alpha$	Weak*

An example of a contextual version of derived rules with one premise:

SeAsCtxt	
$\Sigma; \Gamma, \gamma \,\&\, \neg\alpha \vdash_\iota \alpha$	prem
$\Sigma, \gamma; \Gamma, \neg\alpha \vdash_\iota \alpha$	toCtxt
$\Sigma, \gamma; \Gamma \vdash_\iota \alpha$	SeAs
$\Sigma; \gamma, \Gamma \vdash_\iota \alpha$	FromCtxt2

SeDeCtxt	
$\Sigma; \Gamma, \gamma \,\&\, \alpha \vdash_\iota \neg\alpha$	prem
$\Sigma, \gamma; \Gamma, \alpha \vdash_\iota \neg\alpha$	toCtxt
$\Sigma, \gamma; \Gamma \vdash_\iota \neg\alpha$	SeDe
$\Sigma; \gamma, \Gamma \vdash_\iota \neg\alpha$	FromCtxt2

Deduction rules:

DdRu1	
$\Sigma; \Gamma \vdash_\iota \alpha \Rightarrow \beta$	prem
$\Sigma, \Gamma; \vdash_\iota \alpha \Rightarrow \beta$	toCtxt(*)
$\Sigma, \Gamma; \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, \beta)$	defCons
$\Sigma, \Gamma; \alpha \,\&\, \neg\beta \vdash_\iota \alpha \,\&\, \neg\beta$	ass
$\Sigma, \Gamma; \alpha \,\&\, \neg\beta \vdash_\iota \beta$	contra
$\Sigma, \Gamma; \alpha \vdash_\iota \beta$	SeAsCtxt

DdRu2	
$\Sigma; \Gamma, \alpha \vdash_\iota \beta$	prem
$\Sigma, \Gamma; \alpha \vdash_\iota \beta$	toCtxt(*)
$\Sigma, \Gamma, \alpha; \vdash_\iota \beta$	toCtxt
$\Sigma, \Gamma, \alpha; \vdash_\iota \mathbf{\Delta}(\beta)$	defCons
$\Sigma, \Gamma, \alpha; \neg\beta \vdash_\iota \neg\beta$	ass
$\Sigma, \Gamma, \alpha; \neg\beta \vdash_\iota \alpha \Rightarrow \beta$	contra
$\Sigma, \Gamma; \alpha \,\&\, \neg\beta \vdash_\iota \alpha \Rightarrow \beta$	fromCtxt
$\Sigma, \Gamma; \vdash_\iota \alpha \Rightarrow \beta$	SeDe
$\Sigma; \Gamma \vdash_\iota \alpha \Rightarrow \beta$	FromCtxt2*

where 'toCtxt(*)' indicates a repeated application of the toCtxt rule.

Note that in DdRu1, Γ is pushed into the context. Semantically, this is necessary because from the premise, we can't conclude that α is defined when all Σ are valid, but we only know α to be defined when both Σ and Γ are defined.
If we have also $\Sigma_2; \vdash_\iota \mathbf{\Delta}(\alpha)$, we can get Γ out of the context again:

$\Sigma; \Gamma \vdash_\iota \alpha \Rightarrow \beta$	prem
$\Sigma, \Gamma; \alpha \vdash_\iota \beta$	DdRu1
$\Sigma_2; \alpha \vdash_\iota \alpha$	ass
$\Sigma, \Sigma_2; \Gamma, \alpha \vdash_\iota \alpha \Rightarrow \beta \,\&\, \alpha$	&-intro
$\Sigma, \Sigma_2; \Gamma, \alpha \vdash_\iota \beta$	FromCtxt2*

Definedness of definedness:

Theorem 22 (Ddef rule) *For each term t of the* PITFOL *calculus, given $UC(t)$, if $\mathbf{\Delta}(\mathbf{\Delta}(t))$ is not $\top$, then we can derive $\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t))$.*

For each formula α of the PITFOL *calculus, the analogous theorem holds.*

Proof.

We prove this by induction on the complexity of α and t.

- If t is a variable symbol, then $\mathbf{\Delta}(t) \equiv \top$ and hence $\mathbf{\Delta}(\mathbf{\Delta}(t)) \equiv \top$, so we have nothing to prove.
- If $t \equiv f(t_1, t_2, \ldots, t_n)$ or $t \equiv t_1 = t_2$, then we derive

$$\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_1) \,\&\, (\mathbf{\Delta}(t_2) \,\&\, (\cdots \,\&\, (\mathbf{\Delta}(t_{n-1}) \,\&\, \mathbf{\Delta}(t_n))))$$

  i.e.,

$$\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_1)) \,\&\, (\mathbf{\Delta}(t_1) \Rightarrow (\mathbf{\Delta}(\mathbf{\Delta}(t_2)) \,\&\, \mathbf{\Delta}(t_2) \Rightarrow (\cdots \Rightarrow (\mathbf{\Delta}(\mathbf{\Delta}(t_{n-1})) \,\&\, (\mathbf{\Delta}(t_{n-1}) \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(t_n)))))))),$$

as follows:

$$\begin{array}{rl}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_n)) & \text{induction} \\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_{n-1})) & \text{induction} \\
\mathbf{\Delta}(t_{n-1}) \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_n)) & \text{Weak} \\
\vdash_\iota \mathbf{\Delta}(t_{n-1}) \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(t_n)) & \text{DdRu2} \\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_{n-1})) \,\&\, (\mathbf{\Delta}(t_{n-1}) \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(t_n))) & \text{\&-intro} \\
\vdots & \\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_2)) \,\&\, (\mathbf{\Delta}(t_2) \Rightarrow (\mathbf{\Delta}(\mathbf{\Delta}(t_3)) \,\&\, \mathbf{\Delta}(t_3) \Rightarrow (\dots)) & \text{\&-intro} \\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_1)) & \text{induction} \\
\mathbf{\Delta}(t_1) \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(t_2)) \,\&\, (\mathbf{\Delta}(t_2) \Rightarrow (\mathbf{\Delta}(\mathbf{\Delta}(t_3)) \,\&\, \mathbf{\Delta}(t_3) \Rightarrow (\dots)) & \text{Weak} \\
\vdash_\iota \mathbf{\Delta}(t_1) \Rightarrow (\mathbf{\Delta}(\mathbf{\Delta}(t_2)) \,\&\, (\mathbf{\Delta}(t_2) \Rightarrow (\dots)) & \text{DdRu2} \\
\vdash_\iota \mathbf{\Delta}(t_1) \,\&\, (\mathbf{\Delta}(t_1) \Rightarrow (\mathbf{\Delta}(\mathbf{\Delta}(t_2)) \,\&\, (\mathbf{\Delta}(t_2) \Rightarrow (\dots))) & \text{\&-intro}
\end{array}$$

- If $t \equiv \iota x_\psi(\varphi)$ then we have

$$\begin{array}{rl}
\psi \vdash_\iota \exists! x(\varphi) & \text{prem} \\
\vdash_\iota \mathbf{\Delta}(\psi) & \text{defAnt}
\end{array}$$

- If $\alpha \equiv p(t_1, t_2, \dots, t_n)$ then we proceed as in the case $t \equiv f(t_1, t_2, \dots, t_n)$.

- If $\alpha \equiv \neg\beta$, then by induction, we have $\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\beta))$, which is the required sequent.

- If $\alpha \equiv \beta \,\&\, \gamma$, then we have to derive $\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\beta) \,\&\, (\beta \Rightarrow \mathbf{\Delta}(\gamma)))$, i.e. $\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\beta)) \,\&\, (\mathbf{\Delta}(\beta) \Rightarrow \mathbf{\Delta}(\beta \Rightarrow \mathbf{\Delta}(\gamma)))$:

$$\begin{array}{rl}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\beta)) & \text{induction} \\
\mathbf{\Delta}(\beta) \vdash_\iota \mathbf{\Delta}(\beta) & \text{ass} \\
\mathbf{\Delta}(\beta)\,; \vdash_\iota \mathbf{\Delta}(\beta) & \text{toCtxt} \\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\gamma)) & \text{induction} \\
\mathbf{\Delta}(\beta)\,; \beta \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\gamma)) & \text{Weak} \\
\mathbf{\Delta}(\beta)\,; \vdash_\iota \beta \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(\gamma)) & \text{DdRu2} \\
\mathbf{\Delta}(\beta)\,; \vdash_\iota \mathbf{\Delta}(\beta) \,\&\, (\beta \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(\gamma))) & \text{\&-intro} \\
\mathbf{\Delta}(\beta) \vdash_\iota \mathbf{\Delta}(\beta \Rightarrow \mathbf{\Delta}(\gamma)) & \text{fromCtxt} \\
\vdash_\iota \mathbf{\Delta}(\beta) \Rightarrow \mathbf{\Delta}(\beta \Rightarrow \mathbf{\Delta}(\gamma)) & \text{DdRu2} \\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\beta)) \,\&\, (\mathbf{\Delta}(\beta) \Rightarrow \mathbf{\Delta}(\beta \Rightarrow \mathbf{\Delta}(\gamma))) & \text{\&-intro}
\end{array}$$

- If $\alpha \equiv \forall x(\beta)$, then induction yields $\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\beta))$; applying the $\forall$-intro rule yields the desired sequent.

□

Decomposition and unification of antecedent:

AnDc	
$\Sigma_1; \Gamma, \alpha \,\&\, \beta \vdash_\iota \gamma$	prem
$\Sigma_2; \vdash_\iota \boldsymbol{\Delta}(\beta)$	prem
$\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, \beta)$	defAnt
$\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	&-elim
$\Sigma_1; \alpha \vdash_\iota \alpha$	ass
$\Sigma_2; \beta \vdash_\iota \beta$	ass
$\Sigma_1, \Sigma_2; \alpha, \beta \vdash_\iota \alpha \,\&\, \beta$	&-intro
$\Sigma_1, \Sigma_2; \Gamma, \alpha, \beta \vdash_\iota \gamma$	Cut

AnU	
$\Sigma; \Gamma, \alpha, \beta \vdash_\iota \gamma$	prem
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	defAnt
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\beta)$	defAnt
$\Sigma; \alpha \vdash_\iota \boldsymbol{\Delta}(\beta)$	Weak
$\Sigma; \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta)$	DdRu2
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, \beta)$	&-intro
$\Sigma; \alpha \,\&\, \beta \vdash_\iota \alpha \,\&\, \beta$	ass
$\Sigma; \alpha \,\&\, \beta \vdash_\iota \alpha$	&-elim
$\Sigma; \Gamma, \alpha \,\&\, \beta, \beta \vdash_\iota \gamma$	Cut
$\Sigma; \alpha \,\&\, \beta \vdash_\iota \beta$	&-elim
$\Sigma; \Gamma, \alpha \,\&\, \beta \vdash_\iota \gamma$	Cut

Modus Ponens:

MP	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \Delta \vdash_\iota \alpha \Rightarrow \beta$	prem
$\Sigma_2, \Delta; \alpha \vdash_\iota \beta$	DdRu1
$\Sigma_1, \Sigma_2, \Delta; \Gamma \vdash_\iota \beta$	Cut
$\Sigma_1, \Sigma_2; \Gamma, \Delta \vdash_\iota \beta$	FromCtxt2*

Contextual version of assumption rule:

AssCtxt	
$UC(\alpha)$	
$\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha))$	Ddef
$\boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha)$	ass
$\boldsymbol{\Delta}(\alpha)\,; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	toCtxt
$\boldsymbol{\Delta}(\alpha)\,; \alpha \vdash_\iota \alpha$	ass

Introduction of the implication:

$$\begin{array}{rr}
\boxed{\Rightarrow\text{-intro}} & \\
UC(\alpha), UC(\beta) & \\
\mathbf{\Delta}(\beta \,\&\, \alpha)\,;\beta \,\&\, \alpha \vdash_\iota \beta \,\&\, \alpha & \text{AssCtxt} \\
\mathbf{\Delta}(\beta \,\&\, \alpha)\,;\beta \,\&\, \alpha \vdash_\iota \beta & \text{\&-elim} \\
\mathbf{\Delta}(\beta \,\&\, \alpha)\,,\beta;\alpha \vdash_\iota \beta & \text{toCtxt} \\
\mathbf{\Delta}(\beta \,\&\, \alpha)\,,\beta;\vdash_\iota \alpha \Rightarrow \beta & \text{DdRu2} \\
\mathbf{\Delta}(\beta \,\&\, \alpha)\,;\beta \vdash_\iota \alpha \Rightarrow \beta & \text{fromCtxt}
\end{array}$$

Antecedent as consequent:

$$\begin{array}{rr}
\boxed{\text{Cons}} & \\
\Sigma;\Gamma,\alpha \vdash_\iota \beta & \text{prem} \\
\Sigma;\vdash_\iota \mathbf{\Delta}(\alpha) & \text{defAnt} \\
\Sigma;\alpha \vdash_\iota \alpha & \text{ass} \\
\Sigma;\Gamma,\alpha \vdash_\iota \alpha \,\&\, \beta & \text{\&-intro} \\
\Sigma;\Gamma,\alpha \vdash_\iota \alpha & \text{\&-elim}
\end{array}$$

Context as consequent:

$$\begin{array}{rr}
\boxed{\text{ConsCtxt}} & \\
\Sigma_1,\alpha,\Sigma_2;\Gamma \vdash_\iota \beta & \text{prem} \\
\Sigma_1,\alpha,\sigma_1,\sigma_2,\ldots,\sigma_{n-1};\sigma_n \,\&\, \Gamma \vdash_\iota \beta & \text{fromCtxt} \\
\vdots & \\
\Sigma_1;\alpha \,\&\, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n \,\&\, \Gamma \vdash_\iota \beta & \text{fromCtxt} \\
\Sigma_1;\alpha \,\&\, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n \,\&\, \Gamma \vdash_\iota \alpha \,\&\, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n \,\&\, \gamma_1 & \text{Cons} \\
\Sigma_1;\alpha \,\&\, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n \,\&\, \Gamma \vdash_\iota \alpha & \text{\&-elim} \\
\Sigma_1,\alpha;\sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n \,\&\, \Gamma \vdash_\iota \alpha & \text{toCtxt} \\
\vdots & \\
\Sigma_1,\alpha,\Sigma_2;\Gamma \vdash_\iota \alpha & \text{toCtxt}
\end{array}$$

with $\Sigma_2 \equiv \sigma_1,\sigma_2,\ldots,\sigma_n$ and $\Gamma \equiv \gamma_1 \,\&\, \gamma_2 \,\&\, \cdots \,\&\, \gamma_m$. (If Γ is empty, then setting $\gamma_1 = \top$ yields a proof for this case.)

Cut rule where the formula to be cut appears inside a context:

$$\begin{array}{rr}
\boxed{\text{CutCtxt}} & \\
\Sigma;\Gamma \vdash_\iota \alpha & \text{prem} \\
\Sigma_1,\alpha,\Sigma_2;\Delta \vdash_\iota \beta & \text{prem} \\
\Sigma,\Gamma;\vdash_\iota \alpha & \text{toCtxt*} \\
\Sigma_1,\alpha,\sigma_1,\sigma_2,\ldots,\sigma_{n-1};\sigma_n \,\&\, \Delta \vdash_\iota \beta & \text{fromCtxt} \\
\vdots & \\
\Sigma_1;\alpha \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_1)))), &
\end{array}$$

$$\dfrac{\ldots, \alpha \,\&\, (\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_m))) \vdash_\iota \beta}{\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, (\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_1))))}\,\text{fromCtxt}\quad\text{defAnt}$$
$$\dfrac{\Sigma_1; \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_1)))}{\Sigma, \Gamma, \Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_1)))}\,\&\text{-elim}\quad\text{MP}$$

$$\dfrac{\begin{array}{l}\Sigma_1, \Sigma, \Gamma; \alpha, \sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_1))),\\ \quad \alpha \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_2)))),\\ \quad \ldots, \alpha \,\&\, (\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_m))) \vdash_\iota \beta\end{array}}{\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, (\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_1))))}\,\text{AnDc}\quad\text{defAnt}$$
$$\dfrac{\Sigma_1; \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_2)))}{\Sigma, \Gamma, \Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_2)))}\,\&\text{-elim}\quad\text{MP}$$

$$\dfrac{\begin{array}{l}\Sigma_1, \Sigma, \Gamma; \alpha, \sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_1))),\\ \quad \sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_2))),\\ \quad \alpha \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_3)))),\\ \quad \ldots, \alpha \,\&\, (\sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_m))) \vdash_\iota \beta\end{array}}{\begin{array}{l}\Sigma_1, \Sigma, \Gamma; \alpha, \sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_1))),\\ \quad \ldots, \sigma_1 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \delta_m)) \vdash_\iota \beta\end{array}}\,\text{AnDc}$$
$$\dfrac{\Sigma, \Sigma_1, \Gamma; \Gamma, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \Delta \vdash_\iota \beta}{\Sigma, \Sigma_1, \Gamma; \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \Delta \vdash_\iota \beta}\,\text{Cut}\quad\text{toCtxt*}$$
$$\dfrac{\Sigma, \Sigma_1, \Gamma; \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \Delta \vdash_\iota \beta}{\Sigma, \Sigma_1, \Gamma, \Sigma_2; \Delta \vdash_\iota \beta}\,\text{toCtxt*}$$

with $\Sigma_2 \equiv \sigma_1, \sigma_2, \ldots, \sigma_n$ and $\Delta \equiv \delta_1, \delta_2, \ldots, \delta_m$; we denote a repeated application of the toCtxt rule as toCtxt*.

Variant of the previous rule:

$$\dfrac{\begin{array}{l}\Sigma; \Gamma \vdash_\iota \alpha\\ \Sigma_1, \alpha, \Sigma_2; \Delta \vdash_\iota \beta\end{array}}{\Sigma, \Sigma_1, \Gamma, \Sigma_2; \Delta \vdash_\iota \beta}\,\text{CutCtxt}$$

$$\boxed{\text{CutCtxtCtxt}}$$

$$\dfrac{\Sigma_1, \alpha, \Sigma_2; \vdash_\iota \boldsymbol{\Delta}(\delta_1)}{\Sigma_1, \alpha, \sigma_1, \ldots, \sigma_{n-1}; \sigma_n \vdash_\iota \boldsymbol{\Delta}(\delta_1)}\,\text{defAnt}\quad\text{fromCtxt}$$

$$\vdots$$

$$\dfrac{\Sigma_1; \alpha \,\&\, \sigma_1 \,\&\, \cdots \,\&\, \sigma_n \vdash_\iota \boldsymbol{\Delta}(\delta_1)}{\Sigma_1, \Sigma, \Gamma, \Sigma_2; \Delta \vdash_\iota \beta}\,\text{fromCtxt}\quad\text{WeakCtxtR}$$

with again $\Sigma_2 \equiv \sigma_1, \sigma_2, \ldots, \sigma_n$.

Associativity of the conjunction.

AssocConj1

$\Sigma;\Gamma \vdash_\iota \alpha \,\&\, (\beta \,\&\, \gamma)$	prem
$\Sigma;\Gamma \vdash_\iota \alpha$	&-elim
$\Sigma;\Gamma \vdash_\iota \beta \,\&\, \gamma$	&-elim
$\Sigma;\Gamma \vdash_\iota \beta$	&-elim
$\Sigma;\Gamma \vdash_\iota \gamma$	&-elim
$\Sigma;\Gamma \vdash_\iota \alpha \,\&\, \beta$	&-intro
$\Sigma;\Gamma \vdash_\iota (\alpha \,\&\, \beta) \,\&\, \gamma$	&-intro

AssocConj2

$\Sigma;\Gamma \vdash_\iota (\alpha \,\&\, \beta) \,\&\, \gamma$	prem
$\Sigma;\Gamma \vdash_\iota \alpha \,\&\, \beta$	&-elim
$\Sigma;\Gamma \vdash_\iota \gamma$	&-elim
$\Sigma;\Gamma \vdash_\iota \alpha$	&-elim
$\Sigma;\Gamma \vdash_\iota \beta$	&-elim
$\Sigma;\Gamma \vdash_\iota \beta \,\&\, \gamma$	&-intro
$\Sigma;\Gamma \vdash_\iota \alpha \,\&\, (\beta \,\&\, \gamma)$	&-intro

DefAssocConj1

$\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, (\beta \,\&\, \gamma))$	prem
$\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha)$	&-elim
$\Sigma;\Gamma \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta \,\&\, \gamma)$	&-elim
$\Sigma,\Gamma;\alpha \vdash_\iota \boldsymbol{\Delta}(\beta \,\&\, \gamma)$	DdRu1
$\Sigma,\Gamma;\alpha \vdash_\iota \boldsymbol{\Delta}(\beta)$	&-elim
$\Sigma,\Gamma;\alpha \vdash_\iota \beta \Rightarrow \boldsymbol{\Delta}(\gamma)$	&-elim
$\Sigma,\Gamma;\vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta)$	DdRu2
$\Sigma;\Gamma \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta)$	fromCtxt(*)
$\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, \beta)$	&-intro
$\Sigma,\Gamma,\alpha;\beta \vdash_\iota \boldsymbol{\Delta}(\gamma)$	DdRu1
$\Sigma,\Gamma;\alpha \,\&\, \beta \vdash_\iota \boldsymbol{\Delta}(\gamma)$	fromCtxt
$\Sigma,\Gamma;\vdash_\iota (\alpha \,\&\, \beta) \Rightarrow \boldsymbol{\Delta}(\gamma)$	DdRu2
$\Sigma;\Gamma \vdash_\iota (\alpha \,\&\, \beta) \Rightarrow \boldsymbol{\Delta}(\gamma)$	fromCtxt(*)
$\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}((\alpha \,\&\, \beta) \,\&\, \gamma)$	&-intro

DefAssocConj2

$\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}((\alpha \,\&\, \beta) \,\&\, \gamma)$	prem
$\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, \beta)$	&-elim
$\Sigma;\Gamma \vdash_\iota (\alpha \,\&\, \beta) \Rightarrow \boldsymbol{\Delta}(\gamma)$	&-elim
$\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha)$	&-elim
$\Sigma;\Gamma \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta)$	&-elim
$\Sigma,\Gamma;\alpha \,\&\, \beta \vdash_\iota \boldsymbol{\Delta}(\gamma)$	DdRu1
$\Sigma,\Gamma,\alpha;\beta \vdash_\iota \boldsymbol{\Delta}(\gamma)$	toCtxt
$\Sigma,\Gamma,\alpha;\vdash_\iota \beta \Rightarrow \boldsymbol{\Delta}(\gamma)$	DdRu2
$\Sigma,\Gamma;\alpha \vdash_\iota \beta \Rightarrow \boldsymbol{\Delta}(\gamma)$	fromCtxt
$\Sigma,\Gamma;\alpha \vdash_\iota \boldsymbol{\Delta}(\beta)$	DdRu1
$\Sigma,\Gamma;\alpha \vdash_\iota \boldsymbol{\Delta}(\beta \,\&\, \gamma)$	&-intro
$\Sigma,\Gamma;\vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta \,\&\, \gamma)$	DdRu2
$\Sigma;\Gamma \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta \,\&\, \gamma)$	fromCtxt(*)
$\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, (\beta \,\&\, \gamma))$	&-intro

where fromCtxt(*) indicates a repeated use of the fromCtxt rule. Note that these rules not yet show that everywhere in a formula, we can apply associativity of the conjunction; that will have to wait until property 34.

Decomposition and unification of context:

$$\frac{}{\boxed{\text{CtxtDc}}}$$

$$\frac{\Sigma_1, \alpha \,\&\, \beta, \Sigma_2; \Gamma \vdash_\iota \gamma}{\Sigma_1, \alpha \,\&\, \beta, \sigma_1, \sigma_2, \ldots, \sigma_{n-1}; \sigma_n \,\&\, \Gamma \vdash_\iota \gamma}\ \text{prem}$$

$$\frac{\Sigma_1, \alpha \,\&\, \beta, \sigma_1, \sigma_2, \ldots, \sigma_{n-1}; \sigma_n \,\&\, \Gamma \vdash_\iota \gamma}{\Sigma_1, \alpha \,\&\, \beta, \sigma_1, \sigma_2, \ldots, \sigma_{n-2}; \sigma_{n-1} \,\&\, (\sigma_n \,\&\, \gamma_1), \ldots, \sigma_{n-1} \,\&\, (\sigma_n \,\&\, \gamma_m) \vdash_\iota \gamma}\ \text{fromCtxt}$$

$$\vdots$$

$$\frac{\Sigma_1; (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))), \ldots, (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m)))) \vdash_\iota \gamma}{\Sigma_1; \vdash_\iota \boldsymbol{\Delta}((\alpha \,\&\, \beta) \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))}\ \text{fromCtxt}\ \ \text{defAnt}$$

$$\frac{\Sigma_1; \vdash_\iota \boldsymbol{\Delta}((\alpha \,\&\, \beta) \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))}{\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))))}\ \text{DefAssocConj2}$$

$$\frac{\Sigma_1; \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))) \vdash_\iota \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))}{\Sigma_1; \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))) \vdash_\iota (\alpha \,\&\, \beta) \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))}\ \text{ass}\ \ \text{AssocConj1}$$

$$\frac{\Sigma_1; \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))), (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_2)))), \ldots, (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m)))) \vdash_\iota \gamma}{\Sigma_1; \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))), \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_2))), (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_3)))), \ldots, (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m)))) \vdash_\iota \gamma}\ \text{Cut}$$

$$\vdots$$

$$\frac{\Sigma_1; \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))), \ldots, \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m))) \vdash_\iota \gamma}{\Sigma_1, \alpha; \beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)), \ldots, \beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m)) \vdash_\iota \gamma}\ \text{toCtxt}$$

$$\vdots$$

$$\frac{\Sigma_1, \alpha, \beta, \sigma_1, \sigma_2, \ldots; \sigma_n \,\&\, \Gamma \vdash_\iota \gamma}{\Sigma_1, \alpha, \beta, \Sigma_2; \Gamma \vdash_\iota \gamma}\ \text{toCtxt}$$

with $\Sigma_2 \equiv \sigma_1, \sigma_2, \ldots, \sigma_n$ and $\Gamma \equiv \gamma_1, \gamma_2, \ldots, \gamma_m$.

$$\frac{}{\boxed{\text{CtxtU}}}$$

$$\frac{\Sigma_1, \alpha, \beta, \Sigma_2; \Gamma \vdash_\iota \gamma}{\Sigma_1, \alpha, \beta, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_{n-1}; \sigma_n \,\&\, \Gamma \vdash_\iota \gamma}\ \text{prem}$$

$$\frac{\Sigma_1, \alpha, \beta, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_{n-1}; \sigma_n \,\&\, \Gamma \vdash_\iota \gamma}{\Sigma_1, \alpha, \beta, \sigma_1, \sigma_2, \ldots, \sigma_{n-2}; \sigma_{n-1} \,\&\, (\sigma_n \,\&\, \gamma_1), \ldots, \sigma_{n-1} \,\&\, (\sigma_n \,\&\, \gamma_m) \vdash_\iota \gamma}\ \text{fromCtxt}$$

$$\vdots$$

$$\frac{\Sigma_1; \alpha \,\&\, (\beta \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))))), \ldots, \alpha \,\&\, (\beta \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m))))) \vdash_\iota \gamma}{\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))))}\ \text{fromCtxt}\ \ \text{defAnt}$$

$$\frac{\Sigma_1; \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))))}{\Sigma_1; \vdash_\iota \boldsymbol{\Delta}((\alpha \,\&\, \beta) \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))}\ \text{DefAssocConj1}$$

$$\frac{\Sigma_1; (\alpha \,\&\, \beta) \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)) \vdash_\iota (\alpha \,\&\, \beta) \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))}{\Sigma_1; (\alpha \,\&\, \beta) \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)) \vdash_\iota \alpha \,\&\, (\beta \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))}\ \text{ass}\ \ \text{AssocConj2}$$

$$\frac{\Sigma_1; (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))), \alpha \,\&\, (\beta \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_2))))), \ldots, \alpha \,\&\, (\beta \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m))))) \vdash_\iota \gamma}{\Sigma_1; (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))),}\ \text{Cut}$$

$$
\begin{array}{rl}
(\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_2)))),\\
\alpha \,\&\, (\beta \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_3))))),\\
\ldots, \alpha \,\&\, (\beta \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m))))) \vdash_\iota \gamma & \text{Cut}\\
\vdots & \\
\Sigma_1; (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1)))),\\
\ldots, (\alpha \,\&\, \beta) \,\&\, (\sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m)))) \vdash_\iota \gamma & \text{Cut}\\
\Sigma_1, \alpha \,\&\, \beta; \sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_1))),\\
\ldots, \sigma_1 \,\&\, (\sigma_2 \,\&\, (\cdots \,\&\, (\sigma_n \,\&\, \gamma_m))) \vdash_\iota \gamma & \text{toCtxt}\\
\vdots & \\
\Sigma_1, \alpha \,\&\, \beta, \Sigma_2; \Gamma \vdash_\iota \gamma & \text{toCtxt}
\end{array}
$$

Two variants of $\vee$-introduction:

$$
\begin{array}{rl}
\boxed{\vee\text{-intro}} & \\
UC(\alpha), UC(\beta) & \\
\boldsymbol{\Delta}(\alpha)\,; \alpha \vdash_\iota \alpha & \text{AssCtxt}\\
\boldsymbol{\Delta}(\alpha \vee \beta)\,; \neg\alpha \,\&\, \neg\beta \vdash_\iota \neg\alpha \,\&\, \neg\beta & \text{AssCtxt}\\
\boldsymbol{\Delta}(\alpha \vee \beta)\,; \neg\alpha \,\&\, \neg\beta \vdash_\iota \neg\alpha & \&\text{-elim}\\
\boldsymbol{\Delta}(\alpha \vee \beta)\,; \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{defCons}\\
\boldsymbol{\Delta}(\alpha \vee \beta)\,; \alpha \vdash_\iota \alpha & \text{CutCtxt}\\
\boldsymbol{\Delta}(\alpha \vee \beta)\,; \alpha, \neg\alpha \,\&\, \neg\beta \vdash_\iota \alpha \vee \beta & \text{contra}\\
\boldsymbol{\Delta}(\alpha \vee \beta)\,; \alpha \vdash_\iota \alpha \vee \beta & \text{SeDe}
\end{array}
$$

$$
\begin{array}{rl}
\boxed{\vee\text{-intro}} & \\
UC(\alpha), UC(\beta) & \\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,; \alpha \,\&\, \neg\beta \vdash_\iota \alpha \,\&\, \neg\beta & \text{AssCtxt}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha; \neg\beta \vdash_\iota \alpha \,\&\, \neg\beta & \text{toCtxt}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha, \neg\beta; \vdash_\iota \alpha \,\&\, \neg\beta & \text{toCtxt}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha, \neg\beta; \vdash_\iota \alpha & \&\text{-elim}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha, \neg\beta; \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{defCons}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha, \neg\beta; \neg\alpha \vdash_\iota \neg\alpha & \text{ass}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha; \neg\beta \,\&\, \neg\alpha \vdash_\iota \neg\alpha & \text{fromCtxt}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,; \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{DefCtxt}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,; \alpha \vdash_\iota \alpha & \text{ass}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha; \vdash_\iota \alpha & \text{toCtxt}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha; \neg\beta \,\&\, \neg\alpha \vdash_\iota \neg(\neg\beta \,\&\, \neg\alpha) & \text{contra}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,, \alpha; \vdash_\iota \beta \vee \alpha & \text{SeDe}\\
\boldsymbol{\Delta}(\alpha \,\&\, \neg\beta)\,; \alpha \vdash_\iota \beta \vee \alpha & \text{fromCtxt}
\end{array}
$$

$$
\begin{array}{rl}
\boxed{\vee\text{-elim}} & \\
\Sigma_1; \Gamma_1 \vdash_\iota \alpha \vee \beta & \text{prem}
\end{array}
$$

$$
\begin{array}{rl}
\Sigma_2; \Gamma_2, \alpha \vdash_\iota \gamma & \text{prem} \\
\Sigma_3; \Gamma_3, \beta \vdash_\iota \gamma & \text{prem} \\
\Sigma_2; \Gamma_2, \alpha \vdash_\iota \boldsymbol{\Delta}(\gamma) & \text{defCons} \\
\Sigma_3; \Gamma_3, \beta \vdash_\iota \boldsymbol{\Delta}(\gamma) & \text{defCons} \\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\gamma)) & \text{Ddef} \\
\Sigma_2; \Gamma_2, \neg\boldsymbol{\Delta}(\gamma) \vdash_\iota \neg\alpha & \text{CoPo1} \\
\Sigma_3; \Gamma_3, \neg\boldsymbol{\Delta}(\gamma) \vdash_\iota \neg\beta & \text{CoPo1} \\
\Sigma_2, \Sigma_3; \Gamma_2, \Gamma_3, \neg\boldsymbol{\Delta}(\gamma) \vdash_\iota \neg\alpha \,\&\, \neg\beta & \&\text{-intro} \\
\Sigma_2, \Sigma_3, \Sigma_1; \Gamma_1, \Gamma_2, \Gamma_3, \neg\boldsymbol{\Delta}(\gamma) \vdash_\iota \boldsymbol{\Delta}(\gamma) & \text{contra} \\
\Sigma_2, \Sigma_3, \Sigma_1; \Gamma_1, \Gamma_2, \Gamma_3 \vdash_\iota \boldsymbol{\Delta}(\gamma) & \text{SeAs} \\
\Sigma_2, \Sigma_3, \Sigma_1, \Gamma_1, \Gamma_2, \Gamma_3; \vdash_\iota \boldsymbol{\Delta}(\gamma) & \text{toCtxt}^* \\
\Sigma_2, \Sigma_3, \Sigma_1, \Gamma_1, \Gamma_2, \Gamma_3; \Gamma_2, \neg\gamma \vdash_\iota \neg\alpha & \text{CoPo1} \\
\Sigma_3; \Sigma_2, \Sigma_1, \Gamma_1, \Gamma_2, \Gamma_3; \Gamma_3, \neg\gamma \vdash_\iota \neg\beta & \text{CoPo1} \\
\Sigma_2, \Sigma_3, \Sigma_1, \Gamma_1, \Gamma_2, \Gamma_3; \Gamma_2, \Gamma_3, \neg\gamma \vdash_\iota \neg\alpha \,\&\, \neg\beta & \&\text{-intro} \\
\Sigma_2, \Sigma_3, \Gamma_1, \Gamma_2, \Gamma_3, \Sigma_1; \Gamma_2, \Gamma_3, \neg\gamma \vdash_\iota \gamma & \text{contra} \\
\Sigma_2, \Sigma_3, \Gamma_1, \Gamma_2, \Gamma_3, \Sigma_1; \Gamma_2, \Gamma_3 \vdash_\iota \gamma & \text{SeAs} \\
\Sigma_1, \Sigma_2, \Sigma_3, \Gamma_1, \Gamma_2, \Gamma_3; \Gamma_2, \Gamma_3 \vdash_\iota \gamma & \text{WeakCtxtL} \\
\Sigma_1, \Sigma_2, \Sigma_3, \Gamma_1, \Gamma_2; \Gamma_2, \Gamma_3 \vdash_\iota \gamma & \text{FromCtxt2}^* \\
\Sigma_1, \Sigma_2, \Sigma_3, \Gamma_1; \Gamma_2, \Gamma_3 \vdash_\iota \gamma & \text{FromCtxt2}^* \\
\Sigma_1, \Sigma_2, \Sigma_3; \Gamma_1, \Gamma_2, \Gamma_3 \vdash_\iota \gamma & \text{FromCtxt2}^*
\end{array}
$$

where toCtxt* indicates a repeated application of the toCtxt rule.

Simultaneous generalisation. For this rule, x must not be a free variable of Σ or Γ.

$$
\begin{array}{rl}
\boxed{\text{SimGen}} & \\
\Sigma; \Gamma, \alpha \vdash_\iota \beta & \text{prem} \\
\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{defAnt} \\
\Sigma; \vdash_\iota \forall x(\boldsymbol{\Delta}(\alpha)) & \forall\text{-intro} \\
\Sigma; \forall x(\alpha) \vdash_\iota \forall x(\alpha) & \text{ass} \\
\Sigma; \forall x(\alpha) \vdash_\iota \alpha & \forall\text{-elim} \\
\Sigma; \Gamma, \forall x(\alpha) \vdash_\iota \beta & \text{Cut} \\
\Sigma; \Gamma, \forall x(\alpha) \vdash_\iota \forall x(\beta) & \forall\text{-intro}
\end{array}
$$

Substitution and generalisation; y must not be a free variable of Σ or Γ and x must not be a free variable of α.

First we suppose that $x \not\equiv y$; choose z such that z does not occur in Σ, Γ or $\forall x([x/y]\alpha)$.

$$\boxed{\text{SG}}$$

$$\begin{array}{rr}
\Sigma; \Gamma \vdash_\iota \alpha & \text{prem} \\
[z/x]\Sigma; [z/x]\Gamma \vdash_\iota \alpha & \text{subst} \\
[z/x]\Sigma; [z/x]\Gamma \vdash_\iota [x/y]\alpha & \text{subst} \\
[z/x]\Sigma; [z/x]\Gamma \vdash_\iota \forall x([x/y]\alpha) & \forall\text{-intro} \\
\Sigma; \Gamma \vdash_\iota \forall x([x/y]\alpha) & \text{subst}
\end{array}$$

If $x \equiv y$, then we get the conclusion using a single application of the $\forall$-intro rule.

Renaming of bound variables in generalisations. For this rule, y must not be a free variable of α.

$$\boxed{\text{RenG}}$$

$$\frac{UC(\alpha)}{\begin{array}{rr}
\forall x(\boldsymbol{\Delta}(\alpha)); \forall x(\alpha) \vdash_\iota \forall x(\alpha) & \text{AssCtxt} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \forall x(\alpha) \vdash_\iota \alpha & \forall\text{-elim} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \forall x(\alpha) \vdash_\iota \forall y([y/x]\alpha) & \text{SG}
\end{array}}$$

Renaming of bound variables in particularisations. For this rule, y must not be a free variable of α.

$$\boxed{\text{RenP}}$$

$$\frac{UC(\alpha)}{\begin{array}{rr}
\forall y(\boldsymbol{\Delta}([y/x]\alpha)); \forall y(\neg\,[y/x]\alpha) \vdash_\iota \forall y(\alpha) & \text{RenG} \\
\vdash_\iota \forall x(\boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha))) & \text{Ddef} \\
\forall x(\boldsymbol{\Delta}(\alpha)) \vdash_\iota \forall x(\boldsymbol{\Delta}(\alpha)) & \text{ass} \\
\forall x(\boldsymbol{\Delta}(\alpha)) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \forall\text{-elim} \\
\forall x(\boldsymbol{\Delta}(\alpha)) \vdash_\iota \forall y([y/x]\boldsymbol{\Delta}(\alpha) & \text{SG} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \forall y(\neg\,[y/x]\alpha) \vdash_\iota \forall y(\alpha) & \text{CutCtxt} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \exists y(\alpha) \vdash_\iota \exists y([y/x]\alpha) & \text{CoPo1}
\end{array}}$$

Particularisation in antecedent when x is not a free variable of Σ, Γ or β:

PartAnt	
$\Sigma; \Gamma, \alpha \vdash_\iota \beta$	prem
$\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\beta))$	Ddef
$\boldsymbol{\Delta}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta)$	ass
$\boldsymbol{\Delta}(\beta)\,; \vdash_\iota \boldsymbol{\Delta}(\beta)$	toCtxt
$\Sigma, \boldsymbol{\Delta}(\beta)\,; \Gamma, \neg\beta \vdash_\iota \alpha$	CoPo1
$\Sigma, \boldsymbol{\Delta}(\beta)\,; \Gamma, \neg\beta \vdash_\iota \forall x(\alpha)$	$\forall$-intro
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\alpha)$	defAnt
$\Sigma; \vdash_\iota \boldsymbol{\Delta}(\forall x(\alpha))$	$\forall$-intro
$\Sigma, \boldsymbol{\Delta}(\beta)\,; \Gamma, \neg\forall x(\neg\alpha) \vdash_\iota \beta$	CoPo3
$\Sigma; \Gamma, \alpha, \neg\forall x(\neg\alpha) \vdash_\iota \beta$	Weak
$\Sigma; \boldsymbol{\Delta}(\beta)\,, \Gamma, \neg\forall x(\neg\alpha) \vdash_\iota \beta$	FromCtxt2
$\Sigma; \Gamma, \alpha \vdash_\iota \boldsymbol{\Delta}(\beta)$	defCons
$\Sigma; \Gamma, \neg\boldsymbol{\Delta}(\beta) \vdash_\iota \neg\alpha$	CoPo1
$\Sigma; \Gamma, \neg\boldsymbol{\Delta}(\beta) \vdash_\iota \forall x(\neg\alpha)$	$\forall$-intro
$\Sigma; \Gamma, \neg\forall x(\neg\alpha) \vdash_\iota \boldsymbol{\Delta}(\beta)$	CoPo3
$\Sigma; \Gamma, \exists x(\alpha) \vdash_\iota \beta$	Cut

Particularisation in consequent:

PartCons	
$\Sigma; \Gamma \vdash_\iota \alpha$	prem
$\Sigma; \Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha)$	defCons
$\Sigma; \Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha)\ \&\ \alpha$	&-intro
$\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha))$	DefCons
$\boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha)$	ass
$\vdash_\iota \boldsymbol{\Delta}(\alpha) \Rightarrow \boldsymbol{\Delta}(\alpha)$	DdRu2
$\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha)$	&-intro
$\vdash_\iota \forall x(\boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha))$	$\forall$-intro
$\forall x(\neg(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha)) \vdash_\iota \forall x(\neg(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha))$	ass
$\forall x(\neg(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha)) \vdash_\iota \neg(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha)$	$\forall$-elim
$\Sigma; \Gamma, \forall x(\neg(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha)) \vdash_\iota \neg\forall x(\neg(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha))$	contra
$\Sigma; \Gamma \vdash_\iota \exists x(\boldsymbol{\Delta}(\alpha)\ \&\ \alpha)$	SeDe

Note that we need $\boldsymbol{\Delta}(\alpha)$ in the conclusion. Consider our running example of a theory describing real numbers, where from

$$x = 0.2 \vdash_\iota \frac{1}{x} = 5$$

we cannot conclude

$$x = 0.2 \vdash_\iota \exists x \left(\frac{1}{x} = 5\right)$$

since this would entail (via the defCons rule)

$$x = 0.2 \vdash_\iota \forall x(\neg(x = 0))$$

which is clearly an unsound sequent.

We can simplify the conclusion if we provide an extra premise:

PartCons2	
$\Sigma_1; \Gamma \vdash_\iota \alpha$	prem
$\Sigma_2; \vdash_\iota \forall x(\mathbf{\Delta}(\alpha))$	prem
$\Sigma_2; \forall x(\neg\alpha) \vdash_\iota \forall x(\neg\alpha)$	ass
$\Sigma_2; \forall x(\neg\alpha) \vdash_\iota \neg\alpha$	∀-elim
$\Sigma_1, \Sigma_2; \Gamma, \forall x(\neg\alpha) \vdash_\iota \neg\forall x(\neg\alpha)$	contra
$\Sigma_1, \Sigma_2; \Gamma \vdash_\iota \exists x(\alpha)$	SeDe

In the example above, the extra premise would be

$$\vdash_\iota \forall x \left(\mathbf{\Delta} \left(\frac{1}{x} = 5 \right) \right)$$

i.e.,

$$\vdash_\iota \forall x(\neg(x = 0))$$

∃-ElimAnt	
$\Sigma; \Gamma, \exists x(\alpha) \vdash_\iota \beta$	prem
$\Sigma; \vdash_\iota \mathbf{\Delta}(\forall x(\alpha))$	defAnt
$\Sigma; \vdash_\iota \mathbf{\Delta}(\alpha)$	∀-elim
$\Sigma; \alpha \vdash_\iota \alpha$	ass
$\Sigma; \alpha \vdash_\iota \exists x(\alpha)$	PartCons2
$\Sigma; \Gamma, \alpha \vdash_\iota \beta$	Cut

For this rule, x must not be a free variable of t:

Existence	
$UC(t)$	
$\mathbf{\Delta}(t)\,; \neg(x = t) \vdash_\iota \neg(x = t)$	AssCtxt
$\mathbf{\Delta}(t)\,; \forall x(\neg(x = t)) \vdash_\iota \forall x(\neg(x = t))$	SimGen
$\mathbf{\Delta}(t)\,; \forall x(\neg(x = t)) \vdash_\iota \neg(x = t)$	∀-elim
$\mathbf{\Delta}(t)\,; \forall x(\neg(x = t)) \vdash_\iota \neg(t = t)$	subst
$\mathbf{\Delta}(t)\,; \forall x(\neg(x = t)) \vdash_\iota \neg(t = t)$	subst
$\mathbf{\Delta}(t) \vdash_\iota t = t$	eq
$\mathbf{\Delta}(t)\,; \vdash_\iota t = t$	toCtxt
$\mathbf{\Delta}(t)\,; \forall x(\neg(x = t)) \vdash_\iota \exists x(x = t)$	contra
$\mathbf{\Delta}(t)\,; \vdash_\iota \exists x(x = t)$	SeDe
$\mathbf{\Delta}(t) \vdash_\iota \exists x(x = t)$	fromCtxt

The following rule functions as "inverse" of eqSubst:

EqSubst2

$$\begin{array}{rl}
\Sigma;\Gamma \vdash_\iota [t/x]\,\alpha & \text{prem} \\
\boldsymbol{\Delta}(\alpha)\,;\neg\alpha \vdash_\iota \neg\alpha & \text{AssCtxt} \\
\boldsymbol{\Delta}(\alpha)\,,\boldsymbol{\Delta}(t)\,;x=t,\neg\alpha \vdash_\iota \neg\,[t/x]\,\alpha & \text{eqSubst} \\
\Sigma,\boldsymbol{\Delta}(\alpha)\,,\boldsymbol{\Delta}(t)\,;\Gamma,x=t,\neg\alpha \vdash_\iota \alpha & \text{contra} \\
\Sigma,\boldsymbol{\Delta}(\alpha)\,,\boldsymbol{\Delta}(t)\,;\Gamma,x=t \vdash_\iota \alpha & \text{SeAs}
\end{array}$$

Symmetry rule for equality, where t_1 and t_2 are arbitrary terms and x, y and z are three different variable symbols not occurring in t_1 or t_2:

ESy

$$\begin{array}{rl}
UC(t_1), UC(t_2) & \\
x=z \vdash_\iota x=z & \text{ass} \\
x=z, x=y \vdash_\iota y=z & \text{eqSubst} \\
x=x, x=y \vdash_\iota y=x & \text{subst} \\
\vdash_\iota x=x & \text{eq} \\
x=y \vdash_\iota y=x & \text{Cut} \\
\boldsymbol{\Delta}(t_2)\,;x=t_2 \vdash_\iota t_2=x & \text{subst} \\
\boldsymbol{\Delta}(t_1)\,,\boldsymbol{\Delta}(t_2)\,;t_1=t_2 \vdash_\iota t_2=t_1 & \text{subst}
\end{array}$$

ESy2

$$\begin{array}{rl}
\Sigma;\Gamma \vdash_\iota t_1=t_2 & \text{prem} \\
\boldsymbol{\Delta}(t_1)\,,\boldsymbol{\Delta}(t_2)\,;t_1=t_2 \vdash_\iota t_2=t_1 & \text{ESy} \\
\boldsymbol{\Delta}(t_1)\,\&\,\boldsymbol{\Delta}(t_2)\,;t_1=t_2 \vdash_\iota t_2=t_1 & \text{CtxtU} \\
\Sigma;\Gamma \vdash_\iota t_2=t_1 & \text{Cut3}
\end{array}$$

Transitivity rule for equality, where t_1, t_2 and t_3 are arbitrary terms and x, y and z are three different variable symbols not occurring in t_1, t_2 or t_3:

ET

$$\begin{array}{rl}
UC(t_1), UC(t_2), UC(t_3) & \\
x=y \vdash_\iota x=y & \text{ass} \\
x=y, y=z \vdash_\iota x=z & \text{eqSubst} \\
\boldsymbol{\Delta}(t_3)\,;x=y, y=t_3 \vdash_\iota x=t_3 & \text{subst} \\
\boldsymbol{\Delta}(t_2)\,,\boldsymbol{\Delta}(t_3)\,;x=t_2, t_2=t_3 \vdash_\iota x=t_3 & \text{subst} \\
\boldsymbol{\Delta}(t_1)\,,\boldsymbol{\Delta}(t_2)\,,\boldsymbol{\Delta}(t_3)\,;t_1=t_2, t_2=t_3 \vdash_\iota t_1=t_3 & \text{subst}
\end{array}$$

ET2

$$\begin{array}{rl}
\Sigma_1;\Gamma \vdash_\iota t_1 = t_2 & \text{prem}\\
\Sigma_2;\Delta \vdash_\iota t_2 = t_3 & \text{prem}\\
\Sigma_1;\Gamma \vdash_\iota \mathbf{\Delta}(t_1)\,\&\,\mathbf{\Delta}(t_2) & \text{defCons}\\
\Sigma_1;\Gamma \vdash_\iota \mathbf{\Delta}(t_1) & \&\text{-elim}\\
\Sigma_1;\Gamma \vdash_\iota \mathbf{\Delta}(t_2) & \&\text{-elim}\\
\Sigma_2;\Delta \vdash_\iota \mathbf{\Delta}(t_2)\,\&\,\mathbf{\Delta}(t_3) & \text{defCons}\\
\Sigma_2;\Delta \vdash_\iota \mathbf{\Delta}(t_3) & \&\text{-elim}\\
\Sigma_1,\Sigma_2;\Gamma,\Delta \vdash_\iota t_1 = t_2\,\&\,t_2 = t_3 & \&\text{-intro}\\
\Sigma_1,\Sigma_2;\Gamma,\Delta \vdash_\iota \mathbf{\Delta}(t_3)\,\&\,(t_1 = t_2\,\&\,t_2 = t_3) & \&\text{-intro}\\
\Sigma_1,\Sigma_2;\Gamma,\Delta \vdash_\iota \mathbf{\Delta}(t_2)\,\& & \\
\qquad(\mathbf{\Delta}(t_3)\,\&\,(t_1 = t_2\,\&\,t_2 = t_3)) & \&\text{-intro}\\
\Sigma_1,\Sigma_2;\Gamma,\Delta \vdash_\iota \mathbf{\Delta}(t_1)\,\&\,(\mathbf{\Delta}(t_2)\,\& & \\
\qquad(\mathbf{\Delta}(t_3)\,\&\,(t_1 = t_2\,\&\,t_2 = t_3))) & \&\text{-intro}\\
\mathbf{\Delta}(t_1)\,,\mathbf{\Delta}(t_2)\,,\mathbf{\Delta}(t_3)\,;t_1 = t_2, t_2 = t_3 \vdash_\iota t_1 = t_3 & \text{ET}\\
\mathbf{\Delta}(t_1)\,,\mathbf{\Delta}(t_2)\,,\mathbf{\Delta}(t_3)\,;t_1 = t_2\,\&\,t_2 = t_3 \vdash_\iota t_1 = t_3 & \text{AnU}\\
\mathbf{\Delta}(t_1)\,,\mathbf{\Delta}(t_2)\,;\mathbf{\Delta}(t_3)\,\&\,(t_1 = t_2\,\&\,t_2 = t_3) \vdash_\iota t_1 = t_3 & \text{fromCtxt}\\
\mathbf{\Delta}(t_1)\,; & \\
\mathbf{\Delta}(t_2)\,\&\,(\mathbf{\Delta}(t_3)\,\&\,(t_1 = t_2\,\&\,t_2 = t_3)) \vdash_\iota t_1 = t_3 & \text{fromCtxt}\\
\mathbf{\Delta}(t_1)\,\& & \\
(\mathbf{\Delta}(t_2)\,\&\,(\mathbf{\Delta}(t_3)\,\&\,(t_1 = t_2\,\&\,t_2 = t_3))) \vdash_\iota t_1 = t_3 & \text{fromCtxt}\\
\Sigma_1,\Sigma_2;\Gamma,\Delta \vdash_\iota t_1 = t_3 & \text{Cut}
\end{array}$$

Replacement rule for equality with predicate symbols, where $x_1, x_2, \ldots, x_n$ are different variable symbols not occurring in the terms $t_1, t_2, \ldots, t_n, t'_1, t'_2, \ldots, t'_n$:

ERp

$$\begin{array}{rl}
UC(t_1), UC(t'_1), \ldots, UC(t_n), UC(t'_n) & \\
p(x_1, x_2, \ldots, x_n) \vdash_\iota p(x_1, x_2, \ldots, x_n) & \text{ass}\\
\mathbf{\Delta}(t'_1)\,;p(x_1, x_2, \ldots, x_n), x_1 = t'_1 \vdash_\iota p(t'_1, x_2, \ldots, x_n) & \text{eqSubst}\\
\mathbf{\Delta}(t'_1)\,,\mathbf{\Delta}(t'_2)\,;p(x_1, x_2, \ldots, x_n), x_1 = t'_1, x_2 = t'_2 \vdash_\iota p(t'_1, t'_2, x_3, \ldots, x_n) & \text{eqSubst}\\
\vdots & \\
\mathbf{\Delta}(t'_1)\,,\mathbf{\Delta}(t'_2)\,,\ldots,\mathbf{\Delta}(t'_n)\,;p(x_1, x_2, \ldots, x_n), & \\
\qquad x_1 = t'_1, x_2 = t'_2, \ldots, x_n = t'_n \vdash_\iota p(t'_1, t'_2, \ldots, t'_n) & \text{eqSubst}\\
\vdots & \\
\mathbf{\Delta}(t_n)\,,\mathbf{\Delta}(t'_1),\ldots,\mathbf{\Delta}(t'_n)\,;p(x_1,\ldots,x_{n-1},t_n), & \\
\qquad x_1 = t'_1, x_2 = t'_2, \ldots, x_{n-1} = t'_{n-1}, t_n = t'_n \vdash_\iota p(t'_1, t'_2, \ldots, t'_n) & \text{subst}\\
\vdots &
\end{array}$$

$$\boldsymbol{\Delta}(t_2)\,,\boldsymbol{\Delta}(t_3)\,,\ldots,\boldsymbol{\Delta}(t_n)\,,\boldsymbol{\Delta}(t'_1)\,,\ldots,\boldsymbol{\Delta}(t'_n)\,;\; p(x_1,t_2,\ldots,t_n),x_1=t'_1,t_2=t'_2,\ldots,t_n=t'_n\vdash_\iota p(t'_1,t'_2,\ldots,t'_n) \quad \text{subst}$$

$$\boldsymbol{\Delta}(t_1)\,,\boldsymbol{\Delta}(t_2)\,,\ldots,\boldsymbol{\Delta}(t_n)\,,\boldsymbol{\Delta}(t'_1)\,,\ldots,\boldsymbol{\Delta}(t'_n)\,;\; p(t_1,t_2,\ldots),t_1=t'_1,t_2=t'_2,\ldots,t_n=t'_n\vdash_\iota p(t'_1,t'_2,\ldots,t'_n) \quad \text{subst}$$

Variant of ERp:

$$\boxed{\text{ERp2}}$$

$$\begin{aligned}
\Sigma_1;\Gamma_1 &\vdash_\iota t_1=t'_1 && \text{prem}\\
\Sigma_2;\Gamma_2 &\vdash_\iota t_2=t'_2 && \text{prem}\\
&\vdots\\
\Sigma_n;\Gamma_n &\vdash_\iota t_n=t'_n && \text{prem}\\
p(x_1,x_2,\ldots,x_n) &\vdash_\iota p(x_1,x_2,\ldots,x_n) && \text{ass}\\
\boldsymbol{\Delta}(t'_1)\,;p(x_1,x_2,\ldots,x_n),x_1=t'_1 &\vdash_\iota p(t'_1,x_2,\ldots,x_n) && \text{eqSubst}\\
\Sigma_1;\Gamma_1 &\vdash_\iota \boldsymbol{\Delta}(t_1)\,\&\,\boldsymbol{\Delta}(t'_1) && \text{defCons}\\
\Sigma_1;\Gamma_1 &\vdash_\iota \boldsymbol{\Delta}(t'_1) && \&\text{-elim}\\
\Sigma_1,\Gamma_1;p(x_1,x_2,\ldots,x_n),x_1=t'_1 &\vdash_\iota p(t'_1,x_2,\ldots,x_n) && \text{CutCtxt}\\
\Sigma_1,\Gamma_1,\boldsymbol{\Delta}(t'_2)\,;p(x_1,x_2,\ldots,x_n),x_1=t'_1,x_2=t'_2 &\vdash_\iota p(t'_1,t'_2,x_3,\ldots,x_n) && \text{eqSubst}\\
\Sigma_2;\Gamma_2 &\vdash_\iota \boldsymbol{\Delta}(t_2)\,\&\,\boldsymbol{\Delta}(t'_2) && \text{defCons}\\
\Sigma_2;\Gamma_2 &\vdash_\iota \boldsymbol{\Delta}(t'_2) && \&\text{-elim}\\
\Sigma_2,\Sigma_1,\Gamma_1,\Gamma_2;p(x_1,x_2,\ldots,x_n),x_1=t'_1,x_2=t'_2 &\vdash_\iota p(t'_1,t'_2,x_3,\ldots,x_n) && \text{CutCtxt}\\
&\vdots\\
\Sigma_n,\ldots,\Sigma_1,\Gamma_1,\ldots,\Gamma_n;p(x_1,x_2,\ldots,x_n),& \\
x_1=t'_1,x_2=t'_2,\ldots,x_n=t'_n &\vdash_\iota p(t'_1,t'_2,\ldots,t'_n) && \text{CutCtxt}\\
\boldsymbol{\Delta}(t_n)\,,\Sigma_n,\ldots,\Sigma_1,\Gamma_1,\ldots,\Gamma_n;& \\
p(x_1,\ldots,x_{n-1},t_n),& \\
x_1=t'_1,\ldots,x_{n-1}=t'_{n-1},t_n=t'_n &\vdash_\iota p(t'_1,t'_2,\ldots,t'_n) && \text{subst}\\
\Sigma_n;\Gamma_n &\vdash_\iota \boldsymbol{\Delta}(t_n) && \&\text{-elim}\\
\Sigma_n,\Gamma_n,\Sigma_{n-1},\ldots,\Sigma_1,& \\
\Gamma_1,\ldots,\Gamma_{n-1};p(x_1,\ldots,x_{n-1},t_n),& \\
x_1=t'_1,\ldots,x_{n-1}=t'_{n-1},t_n=t'_n &\vdash_\iota p(t'_1,t'_2,\ldots,t'_n) && \text{CutCtxt}\\
\Sigma_n,\Gamma_n;&\vdash_\iota t_n=t'_n && \text{toCtxt}^*\\
\Sigma_n,\Gamma_n,\Sigma_{n-1},\ldots,\Sigma_1,& \\
\Gamma_n,\Gamma_1,\ldots,\Gamma_{n-1};p(x_1,\ldots,x_{n-1},t_n),& \\
x_1=t'_1,\ldots,x_{n-1}=t'_{n-1} &\vdash_\iota p(t'_1,t'_2,\ldots,t'_n) && \text{Cut}\\
&\vdots\\
\Sigma_1,\Gamma_1,\ldots,\Sigma_n,\Gamma_n;p(t_1,\ldots,t_n) &\vdash_\iota p(t'_1,t'_2,\ldots,t'_n) && \text{Cut}
\end{aligned}$$

where $x_1,x_2,\ldots,x_n$ are different variable symbols not occurring in the terms $t_1,t_2,\ldots,t_n,t'_1,t'_2,\ldots,t'_n,\Sigma_1,\ldots,\Sigma_n,\Gamma_1,\Gamma_2,\ldots,\Gamma_n$ and toCtxt* denotes a repeated application of the toCtxt rule.

Replacement rule for equality with function symbols, where $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n$ are different variable symbols not occurring in the terms $t_1, t_2, t_n, \ldots, t'_1, t'_2, \ldots, t'_n$.

$$\boxed{\text{ERf}}$$

$$\frac{UC(t_1), UC(t'_1), \ldots, UC(t_n), UC(t'_n)}{f(x_1, x_2, \ldots, x_n) = f(y_1, y_2, \ldots, y_n) \vdash_\iota f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)}\ \text{ass}$$

$$\frac{\boldsymbol{\Delta}(t'_1)\,; f(x_1, x_2, \ldots, x_n) = f(y_1, y_2, \ldots, y_n), y_1 = t'_1 \vdash_\iota f(x_1, \ldots, x_n) = f(t'_1, y_2, \ldots, y_n)}{}\ \text{eqSubst}$$

$$\vdots$$

$$\frac{\boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n), y_1 = t'_1, \ldots, y_n = t'_n \vdash_\iota f(x_1, \ldots, x_n) = f(t'_1, \ldots, t'_n)}{}\ \text{eqSubst}$$

$$\frac{\boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; f(x_1, \ldots, x_n) = f(x_1, y_2, \ldots, y_n), x_1 = t'_1, y_2 = t'_2, \ldots, y_n = t'_n \vdash_\iota f(x_1, \ldots, x_n) = f(t'_1, \ldots, t'_n)}{}\ \text{subst}$$

$$\frac{\boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; f(x_1, \ldots, x_n) = f(x_1, x_2, y_3, \ldots, y_n), x_1 = t'_1, x_2 = t'_2, y_3 = t'_3, \ldots, y_n = t'_n \vdash_\iota f(x_1, \ldots, x_n) = f(t'_1, \ldots, t'_n)}{}\ \text{subst}$$

$$\vdots$$

$$\frac{\boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; f(x_1, \ldots, x_n) = f(x_1, \ldots, x_n), x_1 = t'_1, \ldots, x_n = t'_n \vdash_\iota f(x_1, \ldots, x_n) = f(t'_1, \ldots, t'_n)}{}\ \text{subst}$$

$$\frac{\vdash_\iota f(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)}{}\ \text{eq}$$

$$\frac{\boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; x_1 = t'_1, \ldots, x_n = t'_n \vdash_\iota f(x_1, \ldots, x_n) = f(t'_1, \ldots, t'_n)}{}\ \text{Cut}$$

$$\frac{\boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; x_1 = t'_1, \ldots, x_{n-1} = t'_{n-1}, t_n = t'_n \vdash_\iota f(x_1, \ldots, x_{n-1}, t_n) = f(t'_1, \ldots, t'_n)}{}\ \text{subst}$$

$$\vdots$$

$$\frac{\boldsymbol{\Delta}(t_2), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; x_1 = t'_1, t_2 = t'_2, \ldots, t_n = t'_n \vdash_\iota f(x_1, t_2, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{}\ \text{subst}$$

$$\frac{\boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; t_1 = t'_1, \ldots, t_n = t'_n \vdash_\iota f(t_1, t_2, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{}\ \text{subst}$$

Variant of ERf:

$$\boxed{\text{ERf2}}$$

$$\frac{\Sigma_1; \Gamma_1 \vdash_\iota t_1 = t'_1}{}\ \text{prem}$$

$$\frac{\Sigma_2; \Gamma_2 \vdash_\iota t_2 = t'_2}{}\ \text{prem}$$

$$\vdots$$

$$\frac{\Sigma_n; \Gamma_n \vdash_\iota t_n = t'_n}{}\ \text{prem}$$

$$\frac{\boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n)\,; t_1 = t'_1, t_2 = t'_2, \ldots, t_n = t'_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{}\ \text{ERf}$$

$$\frac{\Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n); \Gamma_1, t_2 = t'_2, \ldots, t_n = t'_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\vdots}\ \text{Cut}$$

$$\frac{\Sigma_{n-1}, \ldots, \Sigma, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n); \Gamma_1, \ldots, \Gamma_{n-1}, t_n = t'_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_n, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_n); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{Cut}$$

$$\frac{}{\Sigma_1; \Gamma_1 \vdash_\iota \boldsymbol{\Delta}(t_1) \,\&\, \boldsymbol{\Delta}(t'_1)}\ \text{defCons}$$

$$\frac{\Sigma_1; \Gamma_1 \vdash_\iota \boldsymbol{\Delta}(t_1) \,\&\, \boldsymbol{\Delta}(t'_1)}{\Sigma_1; \Gamma_1 \vdash_\iota \boldsymbol{\Delta}(t_1)}\ \text{\&-elim}$$

$$\frac{\Sigma_1; \Gamma_1 \vdash_\iota \boldsymbol{\Delta}(t_1) \,\&\, \boldsymbol{\Delta}(t'_1)}{\Sigma_1; \Gamma_1 \vdash_\iota \boldsymbol{\Delta}(t'_1)}\ \text{\&-elim}$$

$$\frac{}{\vdots}$$

$$\frac{\Sigma_n; \Gamma_n \vdash_\iota \boldsymbol{\Delta}(t'_n) \,\&\, \boldsymbol{\Delta}(t_n)}{\Sigma_n; \Gamma_n \vdash_\iota \boldsymbol{\Delta}(t'_n)}\ \text{\&-elim}$$

$$\frac{\Sigma_n, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_{n-1}), \Gamma_n; \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_n, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_{n-1}); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{CutCtxt}$$

$$\frac{\Sigma_n, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_{n-1}); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_n, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_{n-1}); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{FromCtxt2}$$

$$\frac{\Sigma_{n-1}, \Sigma_n, \Sigma_{n-2}, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_{n-2}), \Gamma_{n-1}; \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_{n-1}, \Sigma_n, \Sigma_{n-2}, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_{n-2}); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{CutCtxt}$$

$$\frac{\Sigma_{n-1}, \Sigma_n, \Sigma_{n-2}, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_{n-2}); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_{n-1}, \Sigma_n, \Sigma_{n-2}, \ldots, \Sigma_1, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \boldsymbol{\Delta}(t'_1), \ldots, \boldsymbol{\Delta}(t'_{n-2}); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{FromCtxt2}$$

$$\frac{}{\vdots}$$

$$\frac{\Sigma_1, \ldots, \Sigma_n, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n), \Gamma_1; \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_1, \ldots, \Sigma_n, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{CutCtxt}$$

$$\frac{\Sigma_1, \ldots, \Sigma_n, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_n); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_n, \Sigma_1, \ldots, \Sigma_{n-1}, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_{n-1}), \Gamma_n; \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{FromCtxt2}$$

$$\frac{\Sigma_n, \Sigma_1, \ldots, \Sigma_{n-1}, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_{n-1}); \Gamma_n; \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_n, \Sigma_1, \ldots, \Sigma_{n-1}, \boldsymbol{\Delta}(t_1), \ldots, \boldsymbol{\Delta}(t_{n-1}); \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{FromCtxt2}$$

$$\frac{}{\vdots}$$

$$\frac{\Sigma_1, \ldots, \Sigma_n; \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}{\Sigma_1, \ldots, \Sigma_n; \Gamma_1, \ldots, \Gamma_n \vdash_\iota f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \text{FromCtxt2}$$

3.8.1 Equivalent and interchangeable formulae

In this section, we will investigate some equivalences.

Recall that already in the Hermes calculus, we called two formulae α and β **equivalent** when $\alpha \dashv\vdash \beta$ and **equivalent under the condition** γ when

$\alpha, \gamma \vdash \beta$ and $\beta, \gamma \vdash \alpha$.

It will appear that in the PITFOL calculus, these notions will be replaced by the following definitions: we call α and β **interchangeable** when

$$\begin{cases} \boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta \\ \boldsymbol{\Delta}(\beta)\,;\beta \vdash_\iota \alpha \\ \quad \boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\beta) \\ \quad \boldsymbol{\Delta}(\beta) \vdash_\iota \boldsymbol{\Delta}(\alpha) \end{cases}$$

We will call these four sequents the **interchangeability conditions**; when these are derivable, we will write $\alpha \rightleftharpoons \beta$.

Given a list of formulae Σ, we call α and β **interchangeable under the context** Σ when

$$\begin{cases} \Sigma, \boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta \\ \Sigma, \boldsymbol{\Delta}(\beta)\,;\beta \vdash_\iota \alpha \\ \quad \Sigma; \boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\beta) \\ \quad \Sigma; \boldsymbol{\Delta}(\beta) \vdash_\iota \boldsymbol{\Delta}(\alpha) \end{cases}$$

which we will denote as $\alpha \overset{\Sigma}{\rightleftharpoons} \beta$.

Furthermore, we extend these notions to terms: we call t_1 and t_2 interchangeable when

$$\begin{cases} \boldsymbol{\Delta}(t_1) \vdash_\iota t_1 = t_2 \\ \boldsymbol{\Delta}(t_2) \vdash_\iota t_1 = t_2 \end{cases}$$

(which we will also call the **interchangeability conditions** and also denote as $t_1 \rightleftharpoons t_2$) and interchangeable under the context Σ when

$$\begin{cases} \Sigma; \boldsymbol{\Delta}(t_1) \vdash_\iota t_1 = t_2 \\ \Sigma; \boldsymbol{\Delta}(t_2) \vdash_\iota t_1 = t_2 \end{cases}$$

which we denote as $t_1 \overset{\Sigma}{\rightleftharpoons} t_2$.

We call α **replaceable** by β when

$$\begin{cases} \boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta \\ \boldsymbol{\Delta}(\alpha)\,;\beta \vdash_\iota \alpha \end{cases}$$

and t_1 replaceable by t_2 when

$$\boldsymbol{\Delta}(t_1) \vdash_\iota t_1 = t_2$$

We denote this as $\alpha \rightharpoonup \beta$ and $t_1 \rightharpoonup t_2$.

Semantically, this expresses that α and β (resp. t_1 and t_2) have the same interpretation, except that α may be undefined when β is defined (we could say that β is "defined in more cases than α"). In other words, when α is defined, it has the same interpretation as β; when α is undefined, β may have another interpretation.

For example, $\alpha \equiv x = \iota z_{w\neq 0}(z = y)$ and $\beta \equiv x = y$ express both that x and y are equal, but β is always defined whereas α is only defined when $w \neq 0$; and indeed, it is easy to derive

$$x = \iota z_{w\neq 0}(z = y) \rightharpoonup x = y.$$

We call α **replaceable** by β under the context Σ when

$$\begin{cases} \Sigma, \mathbf{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta \\ \Sigma, \mathbf{\Delta}(\alpha)\,;\beta \vdash_\iota \alpha \end{cases}$$

and t_1 replaceable by t_2 when

$$\Sigma; \mathbf{\Delta}(t_1) \vdash_\iota t_1 = t_2$$

We denote this as $\alpha \overset{\Sigma}{\rightharpoonup} \beta$ and $t_1 \overset{\Sigma}{\rightharpoonup} t_2$.

Theorem 23 (replacement theorem) *1. Given $\alpha \rightharpoonup \beta$, then $A(\alpha) \rightharpoonup A(\beta)$, i.e.,*

$$\begin{cases} \mathbf{\Delta}(A(\alpha))\,;A(\alpha) \vdash_\iota A(\beta) \\ \mathbf{\Delta}(A(\alpha))\,;A(\beta) \vdash_\iota A(\alpha) \end{cases}$$

where $A(\alpha)$ is a formula possibly containing α and $A(\beta)$ is the same formula where a number of instances of α are replaced by β, and $t(\alpha) \rightharpoonup t(\beta)$, i.e.,

$$\mathbf{\Delta}(t(\alpha)) \vdash_\iota t(\alpha) = t(\beta)$$

where $t(\alpha)$ is a term possibly containing α and $t(\beta)$ is the same term where a number of instances of α are replaced by β.

These results only hold if the uniqueness conditions for $A(\alpha)$, resp. $t(\alpha)$ can be derived.

2. Given $t_1 \rightharpoonup t_2$, then analogously, $A(t_1) \rightharpoonup A(t_2)$ and $t(t_1) \rightharpoonup t(t_2)$, i.e.,

$$\begin{cases} \mathbf{\Delta}(A(t_1))\,;A(t_1) \vdash_\iota A(t_2) \\ \mathbf{\Delta}(A(t_1))\,;A(t_2) \vdash_\iota A(t_1) \\ \qquad\qquad \mathbf{\Delta}(t(t_1)) \vdash_\iota t(t_1) = t(t_2) \end{cases}$$

These results only hold if the uniqueness conditions for $A(t_1)$, resp. $t(t_1)$ can be derived.

3. *Given a formula α and the formula $\widetilde{\alpha}$ obtained from α by renaming all its bound variables (as in the statement of the ι-rule). Then $\alpha \rightharpoonup \widetilde{\alpha}$.*

   Given a term t and the term $\widetilde{t}$ obtained from t by renaming all its bound variables. Then $t \rightharpoonup \widetilde{t}$.

   These results only hold if the uniqueness conditions for α, resp. t can be derived.

Informally, the theorem then expresses that under the context $\mathbf{\Delta}(A(\alpha))$, one may always replace a formula α by a formula β that is "more defined" than α. In other words, this expresses that when α is undefined (when $w = 0$ in the example above), we could interpret it in any way we like. To clarify this, consider

$$\alpha \rightharpoonup (\mathbf{\Delta}(\alpha) \Rightarrow \alpha) \,\&\, (\neg\mathbf{\Delta}(\alpha) \Rightarrow \beta)$$

where β is an arbitrary formula. One checks easily that this choice of β fulfills the requirements of the lemma; semantically, this means that under the context $\mathbf{\Delta}(A(\alpha))$, we can interpret α as $(\mathbf{\Delta}(\alpha) \Rightarrow \alpha) \,\&\, (\neg\mathbf{\Delta}(\alpha) \Rightarrow \beta)$, where the interpretation of $(\mathbf{\Delta}(\alpha) \Rightarrow \alpha) \,\&\, (\neg\mathbf{\Delta}(\alpha) \Rightarrow \beta)$ coincides with α when α is defined and with our arbitrary formula γ when α is undefined. In the first example above, this expresses that when α is undefined (when $w = 0$), we could interpret it in any way we like (for example as $x = y$).

In this theorem, we see the formal counterpart of the semantic requirement that when evaluating a valid sequent, we never should encounter an invalid term or formula. Indeed, using this theorem, we can derive another sequent in which the invalid term or formula is replaced by a term or formula that interprets in any way we like (see also the following corollary), so it would indeed be impossible for our interpretation to depend on it.

Proof.

We prove this by induction on the nesting depth of the ι-terms in $A(\alpha)$, $t(\alpha)$ for the first part, in $A(t_1)$, $t(t_1)$ for the second part, α and t for the third part.

In the base case, there are no ι-terms present in the formulae and terms under consideration. We handle this case by induction on $\operatorname{cpl} A(\alpha)$, $\operatorname{cpl} t(\alpha)$, $\operatorname{cpl} A(t_1)$, $\operatorname{cpl} t(t_1)$, $\operatorname{cpl}(\alpha) + 1$ and $\operatorname{cpl}(t) + 1$.

For (1) we have the following cases:

- $t(\alpha) \equiv x$ Since there are no subformulae to replace, $t(\alpha) \equiv t(\beta)$ and we have to derive $\vdash_\iota x = x$, which is trivial.

- $t(\alpha) \equiv f(t_1(\alpha), t_2(\alpha), \ldots, t_n(\alpha))$ Induction on t_1 yields $\mathbf{\Delta}(t_1(\alpha)) \vdash_\iota t_1(\alpha) = t_1(\beta)$ and we get similar sequents for $t_2, \ldots, t_n$. We can use these in the following derivations:

$$
\begin{array}{rr}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(t(\alpha))) & \text{Ddef} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_1(\alpha)) \,\&\, \boldsymbol{\Delta}(t_2(\alpha)) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}(t_n(\alpha)) & \text{ass} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_1(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_1(\alpha) = t_1(\beta) & \text{Cut} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_2(\alpha)) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}(t_n(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_2(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_2(\alpha) = t_2(\beta) & \text{Cut} \\
\vdots & \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_n(\alpha) = t_n(\beta) & \text{Cut} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t(\alpha) = t(\beta) & \text{ERf2}
\end{array}
$$

- $A(\alpha) \equiv p(t_1(\alpha), t_2(\alpha), \ldots, t_n(\alpha))$ We treat the case $A(\alpha) \equiv t_1(\alpha) = t_1(\beta)$ likewise. Induction on t_1 yields $\boldsymbol{\Delta}(t_1(\alpha)) \vdash_\iota t_1(\alpha) = t_1(\beta)$ and we get similar sequents for $t_2, \ldots, t_n$. We can use these in the following derivations:

$$
\begin{array}{rr}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(t(\alpha))) & \text{Ddef} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_1(\alpha)) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}(t_n(\alpha)) & \text{ass} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_1(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_1(\alpha) = t_1(\beta) & \text{Cut} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_2(\alpha)) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}(t_n(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_2(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_2(\alpha) = t_2(\beta) & \text{Cut} \\
\vdots & \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_n(\alpha) = t_n(\beta) & \text{Cut} \\
\boldsymbol{\Delta}(t(\alpha)) \,; p(t_1(\alpha), t_2(\alpha), \ldots, t_n(\alpha)) \vdash_\iota p(t_1(\beta), t_2(\beta), \ldots, t_n(\beta)) & \text{ERp2}
\end{array}
$$

$$
\begin{array}{rr}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(t(\alpha))) & \text{Ddef} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_1(\alpha)) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}(t_n(\alpha)) & \text{ass} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_1(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_1(\alpha) = t_1(\beta) & \text{Cut} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_1(\beta) = t_1(\alpha) & \text{ESy2} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_2(\alpha)) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}(t_n(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota \boldsymbol{\Delta}(t_2(\alpha)) & \&\text{-elim} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_2(\alpha) = t_2(\beta) & \text{Cut} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_2(\beta) = t_2(\alpha) & \text{ESy2} \\
\vdots & \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t_n(\beta) = t_n(\alpha) & \text{ESy2} \\
\boldsymbol{\Delta}(t(\alpha)) \,; p(t_1(\beta), t_2(\beta), \ldots, t_n(\beta)) \vdash_\iota p(t_1(\alpha), t_2(\alpha), \ldots, t_n(\alpha)) & \text{ERp2}
\end{array}
$$

- $A(\alpha) \equiv B(\alpha)\,\&\,C(\alpha)$

$$
\begin{array}{rl}
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\alpha)\,\&\,C(\alpha) \vdash_\iota B(\alpha)\,\&\,C(\alpha) & \text{AssCtxt}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\alpha)\,\&\,C(\alpha) \vdash_\iota B(\alpha) & \&\text{-elim}\\
\mathbf{\Delta}(B(\alpha))\,;B(\alpha) \vdash_\iota B(\beta) & \text{induction}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\alpha)\,\&\,C(\alpha) \vdash_\iota B(\beta) & \text{Cut3}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\alpha)\,\&\,C(\alpha) \vdash_\iota C(\alpha) & \&\text{-elim}\\
\mathbf{\Delta}(C(\alpha))\,;C(\alpha) \vdash_\iota C(\beta) & \text{induction}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\alpha)\,\&\,C(\alpha) \vdash_\iota C(\beta) & \text{Cut3}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\alpha)\,\&\,C(\alpha) \vdash_\iota B(\beta)\,\&\,C(\beta) & \&\text{-intro}
\end{array}
$$

$$
\begin{array}{rl}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))) & \text{Ddef}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha)) \vdash_\iota \mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha)) & \text{ass}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha)) \vdash_\iota \mathbf{\Delta}(B(\alpha)) & \&\text{-elim}\\
\mathbf{\Delta}(B(\alpha))\,;B(\beta) \vdash_\iota B(\alpha) & \text{induction}\\
\mathbf{\Delta}(B(\alpha))\,;\vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{defAnt}\\
\mathbf{\Delta}(B(\alpha)) \vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{fromCtxt}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha)) \vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{Cut}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha)) \vdash_\iota B(\alpha) \Rightarrow \mathbf{\Delta}(C(\alpha)) & \&\text{-elim}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\alpha) \vdash_\iota \mathbf{\Delta}(C(\alpha)) & \text{DdRu1}\\
\mathbf{\Delta}(C(\alpha))\,;C(\beta) \vdash_\iota C(\alpha) & \text{induction}\\
\mathbf{\Delta}(C(\alpha))\,;\vdash_\iota \mathbf{\Delta}(C(\beta)) & \text{defAnt}\\
\mathbf{\Delta}(C(\alpha)) \vdash_\iota \mathbf{\Delta}(C(\beta)) & \text{fromCtxt}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\alpha) \vdash_\iota \mathbf{\Delta}(C(\beta)) & \text{Cut}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;\vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{toCtxt}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta) \vdash_\iota B(\beta) & \text{ass}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;\vdash_\iota \mathbf{\Delta}(B(\alpha)) & \text{toCtxt}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta) \vdash_\iota \mathbf{\Delta}(B(\alpha))\,\&\,B(\beta) & \&\text{-intro}\\
\mathbf{\Delta}(B(\alpha))\,\&\,B(\beta) \vdash_\iota B(\alpha) & \text{fromCtxt}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta) \vdash_\iota B(\alpha) & \text{Cut}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta) \vdash_\iota \mathbf{\Delta}(C(\beta)) & \text{Cut}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;\vdash_\iota B(\beta) \Rightarrow \mathbf{\Delta}(C(\beta)) & \text{DdRu2}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;\vdash_\iota \mathbf{\Delta}(B(\beta)\,\&\,C(\beta)) & \&\text{-intro}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta)\,\&\,C(\beta) \vdash_\iota B(\beta)\,\&\,C(\beta) & \text{ass}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta)\,\&\,C(\beta) \vdash_\iota B(\beta) & \&\text{-elim}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta)\,\&\,C(\beta) \vdash_\iota B(\alpha) & \text{Cut}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta)\,\&\,C(\beta) \vdash_\iota C(\beta) & \&\text{-elim}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta)\,\&\,C(\beta) \vdash_\iota \mathbf{\Delta}(C(\alpha)) & \text{Cut}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta)\,\&\,C(\beta) \vdash_\iota \mathbf{\Delta}(C(\alpha))\,\&\,C(\beta) & \&\text{-intro}\\
\mathbf{\Delta}(C(\alpha))\,\&\,C(\beta) \vdash_\iota C(\alpha) & \text{fromCtxt}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta)\,\&\,C(\beta) \vdash_\iota C(\alpha) & \text{Cut}\\
\mathbf{\Delta}(B(\alpha)\,\&\,C(\alpha))\,;B(\beta)\,\&\,C(\beta) \vdash_\iota B(\alpha)\,\&\,C(\alpha) & \&\text{-intro}
\end{array}
$$

- $A(\alpha) \equiv \neg B(\alpha)$

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(B(\alpha))) & \text{Ddef} \\
\mathbf{\Delta}(B(\alpha)) \vdash_\iota \mathbf{\Delta}(B(\alpha)) & \text{ass} \\
\mathbf{\Delta}(B(\alpha)) \,;\, \vdash_\iota \mathbf{\Delta}(B(\alpha)) & \text{toCtxt} \\
\mathbf{\Delta}(B(\alpha)) \,;\, B(\beta) \vdash_\iota B(\alpha) & \text{induction} \\
\mathbf{\Delta}(B(\alpha)) \,;\, \neg B(\alpha) \vdash_\iota \neg B(\beta) & \text{CoPo1}
\end{array}$$

$$\begin{array}{rr}
\mathbf{\Delta}(B(\alpha)) \,;\, B(\alpha) \vdash_\iota B(\beta) & \text{induction} \\
\mathbf{\Delta}(B(\alpha)) \,;\, B(\beta) \vdash_\iota B(\alpha) & \text{induction} \\
\mathbf{\Delta}(B(\alpha)) \,;\, \vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{defAnt} \\
\mathbf{\Delta}(B(\alpha)) \,;\, \neg B(\beta) \vdash_\iota \neg B(\alpha) & \text{CoPo1}
\end{array}$$

- $A(\alpha) \equiv \forall x(B(\alpha))$

$$\begin{array}{rr}
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\alpha)) \vdash_\iota \forall x(B(\alpha)) & \text{AssCtxt} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\alpha)) \vdash_\iota B(\alpha) & \forall\text{-elim} \\
\mathbf{\Delta}(B(\alpha)) \,;\, B(\alpha) \vdash_\iota B(\beta) & \text{induction} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\alpha)) \vdash_\iota B(\beta) & \text{Cut3} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\alpha)) \vdash_\iota \forall x(B(\beta)) & \forall\text{-intro}
\end{array}$$

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\forall x(B(\alpha)))) & \text{Ddef} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \vdash_\iota \mathbf{\Delta}(\forall x(B(\alpha))) & \text{ass} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \vdash_\iota \mathbf{\Delta}(B(\alpha)) & \forall\text{-elim} \\
\mathbf{\Delta}(B(\alpha)) \,;\, B(\beta) \vdash_\iota B(\alpha) & \text{induction} \\
\mathbf{\Delta}(B(\alpha)) \,;\, \vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{defAnt} \\
\mathbf{\Delta}(B(\alpha)) \vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{fromCtxt} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{Cut} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{toCtxt} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \vdash_\iota \mathbf{\Delta}(\forall x(B(\beta))) & \forall\text{-intro} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\beta)) \vdash_\iota \forall x(B(\beta)) & \text{ass} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\beta)) \vdash_\iota B(\beta) & \forall\text{-elim} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \vdash_\iota \mathbf{\Delta}(B(\alpha)) & \text{toCtxt} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\beta)) \vdash_\iota \mathbf{\Delta}(B(\alpha)) \,\&\, B(\beta) & \&\text{-intro} \\
\mathbf{\Delta}(B(\alpha)) \,\&\, B(\beta) \vdash_\iota B(\alpha) & \text{fromCtxt} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\beta)) \vdash_\iota B(\alpha) & \text{Cut} \\
\mathbf{\Delta}(\forall x(B(\alpha))) \,;\, \forall x(B(\beta)) \vdash_\iota \forall x(B(\alpha)) & \forall\text{-intro}
\end{array}$$

The cases for (2) are similar, essentially by replacing $A(\alpha)$ by $A(t_1)$, $A(\beta)$ by $A(t_2)$, ...; we only explicitly mention

- $t(t_1) \equiv x$ Either no replacements occur and $t(t_1) \equiv t(t_2)$, in which case we have to derive $\vdash_\iota x = x$, which is trivial. Else, the whole term $t(t_1)$ is replaced by t_2 (i.e., $t(t_2) \equiv t_2$) and we have to derive $\vdash_\iota t_1 = t_2$, which is given.

For (3), we have:

- $\alpha \equiv \beta \,\&\, \gamma$ Because we performed induction on $\mathrm{cpl}(\alpha) + 1$, and not on $\mathrm{cpl}(\alpha)$, we can apply induction on (1), yielding

$$\begin{cases} \boldsymbol{\Delta}(\beta \,\&\, \gamma)\,;\beta \,\&\, \gamma \vdash_\iota \widetilde{\beta} \,\&\, \gamma & (1) \\ \boldsymbol{\Delta}(\beta \,\&\, \gamma)\,;\widetilde{\beta} \,\&\, \gamma \vdash_\iota \beta \,\&\, \gamma & (2) \end{cases}$$

and also

$$\begin{cases} \boldsymbol{\Delta}\Big(\widetilde{\beta} \,\&\, \gamma\Big)\,;\widetilde{\beta} \,\&\, \gamma \vdash_\iota \widetilde{\beta} \,\&\, \widetilde{\gamma} & (3) \\ \boldsymbol{\Delta}\Big(\widetilde{\beta} \,\&\, \gamma\Big)\,;\widetilde{\beta} \,\&\, \widetilde{\gamma} \vdash_\iota \widetilde{\beta} \,\&\, \gamma & (4) \end{cases}$$

and

$$\begin{cases} \boldsymbol{\Delta}\Big(\widetilde{\beta} \,\&\, \gamma\Big)\,;\widetilde{\beta} \,\&\, \gamma \vdash_\iota \beta \,\&\, \gamma & (5) \\ \boldsymbol{\Delta}\Big(\widetilde{\beta} \,\&\, \gamma\Big)\,;\beta \,\&\, \gamma \vdash_\iota \widetilde{\beta} \,\&\, \gamma & (6) \end{cases}$$

using which we can derive $\boldsymbol{\Delta}(\beta \,\&\, \gamma)\,;\beta \,\&\, \gamma \vdash_\iota \widetilde{\beta} \,\&\, \widetilde{\gamma}$ by applying Cut3 on (1) and (3), and

$$\begin{array}{rr} \boldsymbol{\Delta}\Big(\widetilde{\beta} \,\&\, \gamma\Big)\,;\widetilde{\beta} \,\&\, \widetilde{\gamma} \vdash_\iota \beta \,\&\, \gamma & \text{Cut3 on (4) and (5)} \\ \boldsymbol{\Delta}(\beta \,\&\, \gamma)\,;\vdash_\iota \boldsymbol{\Delta}\Big(\widetilde{\beta} \,\&\, \gamma\Big) & \text{DefAnt on (2)} \\ \boldsymbol{\Delta}(\beta \,\&\, \gamma)\,;\widetilde{\beta} \,\&\, \widetilde{\gamma} \vdash_\iota \beta \,\&\, \gamma & \text{CutCtxt} \end{array}$$

- $\alpha \equiv \forall x(\beta)$

$$\begin{array}{rr} \boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \forall x(\beta) & \text{AssCtxt} \\ \boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta & \forall\text{-elim} \\ \boldsymbol{\Delta}(\beta)\,;\beta \vdash_\iota \widetilde{\beta} & \text{induction} \\ \boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \widetilde{\beta} & \text{Cut3} \\ \boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \forall x(\widetilde{\beta}) & \forall\text{-intro} \\ \boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \forall y([y/x]\widetilde{\beta}) & \text{SG} \end{array}$$

$$\begin{array}{rr}
\boldsymbol{\Delta}\Big(\forall y([y/x]\widetilde{\beta})\Big)\,;\forall y([y/x]\widetilde{\beta}\vdash_\iota \forall y([y/x]\widetilde{\beta}) & \text{AssCtxt}\\
\boldsymbol{\Delta}\Big(\forall y([y/x]\widetilde{\beta})\Big)\,;\forall y([y/x]\widetilde{\beta}\vdash_\iota [y/x]\widetilde{\beta} & \forall\text{-elim}\\
\boldsymbol{\Delta}\Big(\forall y([y/x]\widetilde{\beta})\Big)\,;\forall y([y/x]\widetilde{\beta}\vdash_\iota \widetilde{\beta} & \text{subst}\\
\boldsymbol{\Delta}\Big(\widetilde{\beta}\Big)\,;\widetilde{\beta}\vdash_\iota \beta & \text{induction}\\
\boldsymbol{\Delta}\Big(\forall y([y/x]\widetilde{\beta})\Big)\,;\forall y([y/x]\widetilde{\beta}\vdash_\iota \beta & \text{Cut3}\\
\boldsymbol{\Delta}\Big(\forall y([y/x]\widetilde{\beta})\Big)\,;\forall y([y/x]\widetilde{\beta}\vdash_\iota \forall x(\beta) & \forall\text{-intro}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\forall x(\beta))) & \text{Ddef}\\
\boldsymbol{\Delta}(\forall x(\beta))\vdash_\iota \boldsymbol{\Delta}(\forall x(\beta)) & \text{ass}\\
\boldsymbol{\Delta}(\forall x(\beta))\vdash_\iota \boldsymbol{\Delta}(\beta) & \forall\text{-elim}\\
\boldsymbol{\Delta}(\beta)\,;\widetilde{\beta}\vdash_\iota \beta & \text{induction}\\
\boldsymbol{\Delta}(\beta)\,;\vdash_\iota \boldsymbol{\Delta}\Big(\widetilde{\beta}\Big) & \text{defAnt}\\
\boldsymbol{\Delta}(\beta)\vdash_\iota \boldsymbol{\Delta}\Big(\widetilde{\beta}\Big) & \text{fromCtxt}\\
\boldsymbol{\Delta}(\forall x(\beta))\vdash_\iota \boldsymbol{\Delta}\Big(\widetilde{\beta}\Big) & \text{Cut}\\
\boldsymbol{\Delta}(\forall x(\beta))\vdash_\iota \forall x(\boldsymbol{\Delta}\Big(\widetilde{\beta}\Big)) & \forall\text{-intro}\\
\boldsymbol{\Delta}(\forall x(\beta))\vdash_\iota \forall y([y/x]\boldsymbol{\Delta}\Big(\widetilde{\beta}\Big)) & \text{SG}\\
\boldsymbol{\Delta}(\forall x(\beta))\,;\forall y([y/x]\widetilde{\beta}\vdash_\iota \forall x(\beta) & \text{CutCtxt}
\end{array}$$

The other cases are analogous.

This concludes the base case of the induction on the nesting depths of ι-terms; we handle the induction step again by structural induction. The cases are identical to the cases above, except for (1) the extra case

- $t(\alpha)\equiv \iota x_{\psi(\alpha)}(\varphi(\alpha))$ We have to our disposal the sequent

  $$\psi(\alpha)\vdash_\iota \exists x(\varphi(\alpha))\,\&\,\forall x\forall y((\varphi(\alpha)\,\&\,[y/x]\varphi(\alpha))\Rightarrow x=y)$$

  where y does not occur in $\varphi(\alpha)$.

  First, we will derive

  $$\psi(\beta)\vdash_\iota \exists x(\varphi(\beta))\,\&\,\forall x\forall w((\varphi(\beta)\,\&\,[w/x]\varphi(\beta))\Rightarrow x=w)$$

  where w does not occur in $\varphi(\alpha)$.

  Choose z different from x and not occurring in $\varphi(\alpha)$ and $\varphi(\beta)$ in the following derivation:

$$
\begin{array}{rll}
\boldsymbol{\Delta}(\psi(\alpha))\,;\psi(\alpha) \vdash_\iota \psi(\beta) & & \text{induction}\\
\vdash_\iota \boldsymbol{\Delta}(\psi(\alpha)) & & \text{defAnt}\\
\psi(\alpha) \vdash_\iota \psi(\beta) & & \text{CutCtxt}\\
\psi(\beta) \vdash_\iota \exists x(\varphi(\alpha)) \,\&\, \forall x\forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & & \text{Cut}\\
\boldsymbol{\Delta}(\forall y(\dots))\,;\forall y(\dots) \vdash_\iota \forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & & \text{AssCtxt}\\
\boldsymbol{\Delta}(\forall y(\dots))\,;\forall y(\dots) \vdash_\iota \forall z((\varphi(\alpha) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z) & & \text{RenG}\\
\boldsymbol{\Delta}(\forall z(\dots))\,;\forall z(\dots) \vdash_\iota \forall z((\varphi(\alpha) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z) & & \text{AssCtxt}\\
\boldsymbol{\Delta}(\forall z(\dots))\,;\forall z(\dots) \vdash_\iota \forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & & \text{RenG}\\
\boldsymbol{\Delta}(\forall y(\dots)) \vdash_\iota \boldsymbol{\Delta}(\forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y)) & & \text{Ddef}\\
\boldsymbol{\Delta}(\forall y(\dots)) \vdash_\iota \boldsymbol{\Delta}(\forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y)) & & \text{ass}\\
\boldsymbol{\Delta}(\forall y(\dots)) \vdash_\iota \boldsymbol{\Delta}(\forall z((\varphi(\alpha) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z)) & & \text{RenG}\\
\boldsymbol{\Delta}(\forall y(\dots))\,;\forall z(\dots) \vdash_\iota \forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & & \text{RenG}\\
\boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota \exists x(\varphi(\alpha)) \,\&\, \forall x\forall z((\varphi(\alpha) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z) & & \text{induction}\\
\psi(\beta) \vdash_\iota \exists x(\varphi(\alpha)) \,\&\, \forall x\forall z((\varphi(\alpha) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z) & & \text{Cut3}\\
\boldsymbol{\Delta}(\varphi(\alpha))\,;\varphi(\alpha) \vdash_\iota \varphi(\beta) & & \text{induction}\\
\boldsymbol{\Delta}(\varphi(\alpha))\,;\varphi(\beta) \vdash_\iota \varphi(\alpha) & & \text{induction}\\
\boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota \exists x(\varphi(\beta)) \,\&\, \forall x\forall z((\varphi(\beta) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z) & & \text{induction}\\
\psi(\beta) \vdash_\iota \exists x(\varphi(\beta)) \,\&\, \forall x\forall z((\varphi(\beta) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z) & & \text{Cut3}\\
\boldsymbol{\Delta}([z/x]\varphi(\alpha))\,;[z/x]\varphi(\alpha) \vdash_\iota [z/x]\varphi(\beta) & & \text{subst}\\
\boldsymbol{\Delta}([z/x]\varphi(\alpha))\,;[z/x]\varphi(\beta) \vdash_\iota [z/x]\varphi(\alpha) & & \text{subst}\\
\boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota \exists x(\varphi(\beta)) \,\&\, \forall x\forall z((\varphi(\beta) \,\&\, [z/x]\varphi(\beta)) \Rightarrow x = z) & & \text{induction}\\
\psi(\beta) \vdash_\iota \exists x(\varphi(\beta)) \,\&\, \forall x\forall z((\varphi(\beta) \,\&\, [z/x]\varphi(\beta)) \Rightarrow x = z) & & \text{Cut3}\\
\psi(\beta) \vdash_\iota \exists x(\varphi(\beta)) & & \\
\qquad \&\, \forall x\forall w((\varphi(\beta) \,\&\, [w/x]\varphi(\beta)) \Rightarrow x = w) & & \text{analogous}
\end{array}
$$

We see here why we needed to perform induction on the nesting depths of the ι-terms first: the formula $\exists!x(\varphi(\alpha))$ can have a complexity much larger than $\mathrm{cpl}(\alpha)$, so we cannot apply structural induction on $\exists!x(\varphi(\alpha))$. But the nesting depth of the ι-terms of $\exists!x(\varphi(\alpha))$ is always 1 less than that of $\iota x_{\psi(\alpha)}(\varphi(\alpha))$, so the induction on the nesting depths helps us out here.

Choose z different from x and not occurring in $\varphi(\alpha)$, $\psi(\alpha)$, $\varphi(\beta)$ and $\psi(\beta)$. Using both uniqueness conditions, we can derive

$$
\begin{array}{rll}
\boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota \forall x\forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & & \text{AssCtxt}\\
\boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota \forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & & \forall\text{-elim}\\
\boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota (\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y & & \forall\text{-elim}\\
\boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota (\widetilde{\varphi(\alpha)} \,\&\, [z/x]\widetilde{\varphi(\beta)}) \Rightarrow x = z & & \text{analogous}\\
\psi(\alpha), \boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota ([\iota x_{\psi(\alpha)}(\varphi(\alpha))/x]\,\widetilde{\varphi(\alpha)} \,\&\, [z/x]\widetilde{\varphi(\beta)}) & & \\
\qquad \Rightarrow \iota x_{\psi(\alpha)}(\varphi(\alpha)) = z & & \text{subst}\\
\psi(\beta), \psi(\alpha), \boldsymbol{\Delta}(\dots)\,;\dots \vdash_\iota ([\iota x_{\psi(\alpha)}(\varphi(\alpha))/x]\,\widetilde{\varphi(\alpha)} \,\&\, [\iota x_{\psi(\beta)}(\varphi(\beta))/x]\,\widetilde{\varphi(\beta)}) & &
\end{array}
$$

$$
\begin{array}{rl}
\Rightarrow \iota x_{\psi(\alpha)}(\varphi(\alpha)) = \iota x_{\psi(\beta)}(\varphi(\beta)) & \text{subst} \\
\psi(\alpha) \vdash_\iota \left[\iota x_{\psi(\alpha)}(\varphi(\alpha))/x\right] \widetilde{\varphi(\alpha)} & \text{iota} \\
\psi(\beta) \vdash_\iota \left[\iota x_{\psi(\beta)}(\varphi(\beta))/x\right] \widetilde{\varphi(\beta)} & \text{iota} \\
\psi(\alpha) \vdash_\iota \left[\iota x_{\psi(\beta)}(\varphi(\beta))/x\right] \widetilde{\varphi(\beta)} & \text{Cut} \\
\psi(\alpha) \vdash_\iota \left[\iota x_{\psi(\alpha)}(\varphi(\alpha))/x\right] \widetilde{\varphi(\alpha)} \;\&\; \left[\iota x_{\psi(\beta)}(\varphi(\beta))/x\right] \widetilde{\varphi(\beta)} & \&\text{-intro} \\
\psi(\alpha); \vdash_\iota \left[\iota x_{\psi(\alpha)}(\varphi(\alpha))/x\right] \widetilde{\varphi(\alpha)} \;\&\; \left[\iota x_{\psi(\beta)}(\varphi(\beta))/x\right] \widetilde{\varphi(\beta)} & \text{toCtxt} \\
\psi(\alpha), \boldsymbol{\Delta}(\dots); \dots \vdash_\iota \iota x_{\psi(\alpha)}(\varphi(\alpha)) = \iota x_{\psi(\beta)}(\varphi(\beta)) & \text{MP} \\
\psi(\alpha) \vdash_\iota \iota x_{\psi(\alpha)}(\varphi(\alpha)) = \iota x_{\psi(\beta)}(\varphi(\beta)) & \text{Cut3}
\end{array}
$$

and an analogous case for (2).

For (3), the extra case is

- $t \equiv \iota x_\psi(\varphi)$

$$
\begin{array}{rl}
\boldsymbol{\Delta}(\psi)\,; \psi \vdash_\iota \widetilde{\psi} & \text{induction} \\
\boldsymbol{\Delta}(\psi)\,; \widetilde{\psi} \vdash_\iota \psi & \text{induction} \\
\psi \vdash_\iota \iota x_\psi(\varphi) = \iota x_{\widetilde{\psi}}(\varphi) & \text{induction} \\
\boldsymbol{\Delta}(\varphi)\,; \varphi \vdash_\iota \widetilde{\varphi} & \text{induction} \\
\boldsymbol{\Delta}(\varphi)\,; \widetilde{\varphi} \vdash_\iota \varphi & \text{induction} \\
\boldsymbol{\Delta}([y/x]\varphi)\,; [y/x]\varphi \vdash_\iota [y/x]\widetilde{\varphi} & \text{subst} \\
\boldsymbol{\Delta}([y/x]\varphi)\,; [y/x]\widetilde{\varphi} \vdash_\iota [y/x]\varphi & \text{subst} \\
\widetilde{\psi} \vdash_\iota \iota x_{\widetilde{\psi}}(\varphi) = \iota x_{\widetilde{\psi}}([y/x]\widetilde{\varphi}) & \text{induction} \\
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{Ddef} \\
\psi \vdash_\iota \widetilde{\psi} & \text{CutCtxt} \\
\psi \vdash_\iota \iota x_{\widetilde{\psi}}(\varphi) = \iota x_{\widetilde{\psi}}([y/x]\widetilde{\varphi}) & \text{Cut} \\
\psi \vdash_\iota \iota x_\psi(\varphi) = \iota x_{\widetilde{\psi}}([y/x]\widetilde{\varphi}) & \text{ET2}
\end{array}
$$

□

As indicated earlier, some of the proofs need to be slightly modified when the definedness of some subformulae is $\top$. In the case $A(\alpha) \equiv B(\alpha)\ \&\ C(\alpha)$ there are a lot of possible variants in the derivation of $\boldsymbol{\Delta}(B(\alpha)\,\&\,C(\alpha))\,; B(\beta)\ \&\ C(\beta) \vdash_\iota B(\alpha)\ \&\ C(\alpha)$; hence we will elaborate this case.

The proof already given handles the case $\boldsymbol{\Delta}(B(\alpha)) \not\equiv \top \not\equiv \boldsymbol{\Delta}(C(\alpha))$. If $\boldsymbol{\Delta}(B(\beta))$ and/or $\boldsymbol{\Delta}(C(\beta))$ would be $\top$, the convention we adopted earlier holds: the proof stays valid by removing the lines from the proof in which the consequent is $\top$.

If $\boldsymbol{\Delta}(B(\alpha)) \equiv \top$, then the proof simplifies to

$$
\begin{array}{rr}
B(\beta) \vdash_\iota B(\alpha) & \text{induction} \\
\vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{defAnt} \\
\mathbf{\Delta}(C(\alpha)) ; C(\beta) \vdash_\iota C(\alpha) & \text{induction} \\
\mathbf{\Delta}(C(\alpha)) ; \vdash_\iota \mathbf{\Delta}(C(\beta)) & \text{defAnt} \\
\mathbf{\Delta}(C(\alpha)) ; B(\beta) \vdash_\iota \mathbf{\Delta}(C(\beta)) & \text{Weak} \\
\mathbf{\Delta}(C(\alpha)) ; \vdash_\iota B(\beta) \Rightarrow \mathbf{\Delta}(C(\beta)) & \text{DdRu2} \\
\mathbf{\Delta}(C(\alpha)) ; \vdash_\iota \mathbf{\Delta}(B(\beta) \,\&\, C(\beta)) & \&\text{-intro} \\
\mathbf{\Delta}(C(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota B(\beta) \,\&\, C(\beta) & \text{ass} \\
\mathbf{\Delta}(C(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota B(\beta) & \&\text{-elim} \\
\mathbf{\Delta}(C(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota B(\alpha) & \text{Cut} \\
\mathbf{\Delta}(C(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota C(\beta) & \&\text{-elim} \\
\mathbf{\Delta}(C(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota C(\alpha) & \text{Cut} \\
\mathbf{\Delta}(C(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota B(\alpha) \,\&\, C(\alpha) & \&\text{-intro}
\end{array}
$$

Again, If $\mathbf{\Delta}(B(\beta))$ and/or $\mathbf{\Delta}(C(\beta))$ is $\top$, remove the appropriate lines from the proof.

Finally, if $\mathbf{\Delta}(C(\alpha)) \equiv \top$, we have

$$
\begin{array}{rr}
\mathbf{\Delta}(B(\alpha)) ; B(\beta) \vdash_\iota B(\alpha) & \text{induction} \\
\mathbf{\Delta}(B(\alpha)) ; \vdash_\iota \mathbf{\Delta}(B(\beta)) & \text{defAnt} \\
C(\beta) \vdash_\iota C(\alpha) & \text{induction} \\
\vdash_\iota \mathbf{\Delta}(C(\beta)) & \text{defAnt} \\
\mathbf{\Delta}(B(\alpha)) ; B(\beta) \vdash_\iota \mathbf{\Delta}(C(\beta)) & \text{Weak} \\
\mathbf{\Delta}(B(\alpha)) ; \vdash_\iota B(\beta) \Rightarrow \mathbf{\Delta}(C(\beta)) & \text{DdRu2} \\
\mathbf{\Delta}(B(\alpha)) ; \vdash_\iota \mathbf{\Delta}(B(\beta) \,\&\, C(\beta)) & \&\text{-intro} \\
\mathbf{\Delta}(B(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota B(\beta) \,\&\, C(\beta) & \text{ass} \\
\mathbf{\Delta}(B(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota B(\beta) & \&\text{-elim} \\
\mathbf{\Delta}(B(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota B(\alpha) & \text{Cut} \\
\mathbf{\Delta}(B(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota C(\beta) & \&\text{-elim} \\
\mathbf{\Delta}(B(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota C(\alpha) & \text{Cut} \\
\mathbf{\Delta}(B(\alpha)) ; B(\beta) \,\&\, C(\beta) \vdash_\iota B(\alpha) \,\&\, C(\alpha) & \&\text{-intro}
\end{array}
$$

Corollary 24 (Replacement in sequents) *Given $\alpha \rightharpoonup \beta$, then*

$$
\frac{\Sigma; \Gamma \vdash_\iota A(\alpha)}{\Sigma; \Gamma \vdash_\iota A(\beta)} \qquad \frac{\Sigma; \Gamma, A(\alpha) \vdash_\iota \gamma}{\Sigma; \Gamma, A(\beta) \vdash_\iota \gamma} \qquad \frac{\Sigma_1, A(\alpha), \Sigma_2; \Gamma \vdash_\iota \gamma}{\Sigma_1, A(\beta), \Sigma_2; \Gamma \vdash_\iota \gamma}
$$

Analogously, given

$$
\mathbf{\Delta}(t_1) \vdash_\iota t_1 = t_2,
$$

then

$$
\frac{\Sigma; \Gamma \vdash_\iota A(t_1)}{\Sigma; \Gamma \vdash_\iota A(t_2)} \qquad \frac{\Sigma; \Gamma, A(t_1) \vdash_\iota \gamma}{\Sigma; \Gamma, A(t_2) \vdash_\iota \gamma} \qquad \frac{\Sigma_1, A(t_1), \Sigma_2; \Gamma \vdash_\iota \gamma}{\Sigma_1, A(t_2), \Sigma_2; \Gamma \vdash_\iota \gamma}
$$

Proof.

We only present the proof for replacement of subformulae; the proofs for replacement of subterms are analogous.

$$\begin{array}{rr} \Sigma; \Gamma \vdash_\iota A(\alpha) & \text{prem} \\ \boldsymbol{\Delta}(A(\alpha)) \,; A(\alpha) \vdash_\iota A(\beta) & \text{Theorem 23} \\ \Sigma; \Gamma \vdash_\iota A(\beta) & \text{Cut3} \end{array}$$

$$\begin{array}{rr} \Sigma; \Gamma, A(\alpha) \vdash_\iota \gamma & \text{prem} \\ \boldsymbol{\Delta}(A(\alpha)) \,; A(\beta) \vdash_\iota A(\alpha) & \text{Theorem 23} \\ \Sigma; \vdash_\iota \boldsymbol{\Delta}(A(\alpha)) & \text{defAnt} \\ \Sigma; A(\beta) \vdash_\iota A(\alpha) & \text{CutCtxt} \\ \Sigma; \Gamma, A(\beta) \vdash_\iota \gamma & \text{Cut} \end{array}$$

For the last rule, we have

$$\Sigma_1, A(\alpha), \sigma_1, \sigma_2, \ldots, \sigma_n; \Gamma \vdash_\iota \gamma$$

where $\Sigma_2 \equiv \sigma_1, \ldots, \sigma_n$. Repeatedly applying fromCtxt, we get

$$\Sigma_1; A(\alpha) \,\&\, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n \,\&\, \Gamma \vdash_\iota \gamma$$

Using the previous case, we can transform this into

$$\Sigma_1; A(\beta) \,\&\, \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n \,\&\, \Gamma \vdash_\iota \gamma$$

and repeatedly applying toCtxt yields the desired sequent. □

Theorem 25 (interchange theorem) *1. If α and β are two interchangeable formulae, then so are $A(\alpha)$ and $A(\beta)$, where $A(\alpha)$ is a formula possibly containing α and $A(\beta)$ is the same formula where a number of occurrences of α are replaced by β, and analogously, $t(\alpha)$ and $t(\beta)$ are two interchangeable terms, where $t(\alpha)$ is a term possibly containing α and $t(\beta)$ is the same term where a number of instances of α are replaced by β.*

These results only hold if the uniqueness conditions for $A(\alpha)$ or $A(\beta)$, resp. $t(\alpha)$ or $t(\beta)$ can be derived.
If the uniqueness conditions for $A(\alpha)$ or $t(\alpha)$ hold, then the uniqueness conditions for $A(\beta)$, resp. $t(\beta)$ hold too and vice versa.

2. If t_1 and t_2 are two interchangeable terms, then so are $A(t_1)$ and $A(t_2)$, resp. $t(t_1)$ and $t(t_2)$, with a similar remark about the uniqueness conditions as in the first case.

3. *Given a formula γ and the formula $\widetilde{\gamma}$ obtained from γ by renaming all its bound variables. Then γ and $\widetilde{\gamma}$ are interchangeable. Analogously, a term t is interchangeable with $\widetilde{t}$.*

Proof.

1. Suppose for example that we are given the uniqueness conditions for $A(\alpha)$.

$$\begin{array}{rr}
\boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta & \text{given} \\
\boldsymbol{\Delta}(\beta)\,;\beta \vdash_\iota \alpha & \text{given} \\
\boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{given} \\
\boldsymbol{\Delta}(\alpha)\,;\beta \vdash_\iota \alpha & \text{CutCtxt} \\
\boldsymbol{\Delta}(A(\alpha))\,;A(\alpha) \vdash_\iota A(\beta) & \text{Th. 23} \\
\boldsymbol{\Delta}(A(\alpha))\,;A(\beta) \vdash_\iota A(\alpha) & \text{Th. 23} \\
\boldsymbol{\Delta}(A(\alpha))\,;\vdash_\iota \boldsymbol{\Delta}(A(\beta)) & \text{defAnt} \\
\boldsymbol{\Delta}(A(\alpha)) \vdash_\iota \boldsymbol{\Delta}(A(\beta)) & \text{fromCtxt} \\
\boldsymbol{\Delta}(\beta) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{given} \\
\boldsymbol{\Delta}(\beta)\,;\alpha \vdash_\iota \beta & \text{CutCtxt} \\
\boldsymbol{\Delta}(A(\beta))\,;A(\beta) \vdash_\iota A(\alpha) & \text{Th. 23} \\
\boldsymbol{\Delta}(A(\beta))\,;A(\alpha) \vdash_\iota A(\beta) & \text{Th. 23} \\
\boldsymbol{\Delta}(A(\beta))\,;\vdash_\iota \boldsymbol{\Delta}(A(\alpha)) & \text{defAnt} \\
\boldsymbol{\Delta}(A(\beta)) \vdash_\iota \boldsymbol{\Delta}(A(\alpha)) & \text{fromCtxt}
\end{array}$$

$$\begin{array}{rr}
\boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta & \text{given} \\
\boldsymbol{\Delta}(\beta)\,;\beta \vdash_\iota \alpha & \text{given} \\
\boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{given} \\
\boldsymbol{\Delta}(\alpha)\,;\beta \vdash_\iota \alpha & \text{CutCtxt} \\
\boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t(\alpha) = t(\beta) & \text{Th. 23} \\
\boldsymbol{\Delta}(\beta) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{given} \\
\boldsymbol{\Delta}(\beta)\,;\alpha \vdash_\iota \beta & \text{CutCtxt} \\
\boldsymbol{\Delta}(t(\beta)) \vdash_\iota t(\beta) = t(\alpha) & \text{Th. 23} \\
\boldsymbol{\Delta}(t(\beta)) \vdash_\iota t(\alpha) = t(\beta) & \text{ESy2}
\end{array}$$

2. Analogously.

3. Analogously, using theorem 23.3.

□

Note that if we are given an equivalence $\alpha \dashv\vdash_\iota \beta$, then we can apply the previous theorem, since using the defAnt rule on $\alpha \vdash_\iota \beta$ yields $\vdash_\iota \boldsymbol{\Delta}(\alpha)$ and

we can obtain $\vdash_\iota \mathbf{\Delta}(\beta)$ analogously. From this observation, it is not difficult to derive the two equivalences required for the application of the previous theorem.

Finally, we remark that that given two out of three of

$$\begin{cases} \mathbf{\Delta}(\alpha) \,\&\, \alpha \dashv\vdash_\iota \mathbf{\Delta}(\beta) \,\&\, \beta \\ \mathbf{\Delta}(\alpha) \,\&\, \neg\alpha \dashv\vdash_\iota \mathbf{\Delta}(\beta) \,\&\, \neg\beta \\ \mathbf{\Delta}(\alpha) \dashv\vdash_\iota \mathbf{\Delta}(\beta) \end{cases}$$

one can derive the remaining sequent and hence apply the previous theorem.

This is similar to the two-valued setting, where given one out of two of

$$\begin{cases} \alpha \dashv\vdash \beta \\ \neg\alpha \dashv\vdash \neg\beta \end{cases}$$

is sufficient to derive the other sequent, and this is sufficient to apply a similar substitution theorem in the two-valued calculus.

Looking at the semantics, this analogy becomes even stronger: for each truth-value v, we have to deduce

$$\alpha \text{ has truth-value } v \quad \text{if and only if} \quad \beta \text{ has truth-value } v$$

and if the interpretation has n truth values, then it is sufficient to derive only $n-1$ of these equivalences.

We will only give the derivations of the $\vdash_\iota$ direction; the derivations for $\dashv_\iota$ are analogous.

$$\begin{array}{rr}
\mathbf{\Delta}(\beta) \,\&\, \beta \vdash_\iota \mathbf{\Delta}(\alpha) \,\&\, \alpha & \text{prem} \\
\mathbf{\Delta}(\beta) \vdash_\iota \mathbf{\Delta}(\alpha) & \text{prem} \\
\mathbf{\Delta}(\beta) \,\&\, \beta \vdash_\iota \alpha & \&\text{-elim} \\
\mathbf{\Delta}(\beta)\,;\beta \vdash_\iota \alpha & \text{toCtxt} \\
\mathbf{\Delta}(\beta)\,;\vdash_\iota \mathbf{\Delta}(\alpha) & \text{toCtxt} \\
\mathbf{\Delta}(\beta)\,;\neg\alpha \vdash_\iota \neg\beta & \text{CoPo1} \\
\mathbf{\Delta}(\alpha) \vdash_\iota \mathbf{\Delta}(\beta) & \text{prem} \\
\mathbf{\Delta}(\alpha)\,;\neg\alpha \vdash_\iota \neg\beta & \text{CutCtxt} \\
\mathbf{\Delta}(\alpha) \,\&\, \neg\alpha \vdash_\iota \neg\beta & \text{fromCtxt} \\
\mathbf{\Delta}(\alpha) \,\&\, \neg\alpha \vdash_\iota \mathbf{\Delta}(\beta) & \text{defCons} \\
\mathbf{\Delta}(\alpha) \,\&\, \neg\alpha \vdash_\iota \mathbf{\Delta}(\beta) \,\&\, \neg\beta & \&\text{-intro}
\end{array}$$

$\mathbf{\Delta}(\beta)\ \&\ \neg\beta \vdash_\iota \mathbf{\Delta}(\alpha)\ \&\ \neg\alpha$	prem
$\mathbf{\Delta}(\beta) \vdash_\iota \mathbf{\Delta}(\alpha)$	prem
$\mathbf{\Delta}(\beta)\ \&\ \neg\beta \vdash_\iota \neg\alpha$	&-elim
$\mathbf{\Delta}(\beta)\,;\neg\beta \vdash_\iota \neg\alpha$	toCtxt
$\mathbf{\Delta}(\beta)\,;\vdash_\iota \mathbf{\Delta}(\alpha)$	toCtxt
$\mathbf{\Delta}(\beta)\,;\alpha \vdash_\iota \beta$	CoPo4
$\mathbf{\Delta}(\alpha) \vdash_\iota \mathbf{\Delta}(\beta)$	prem
$\mathbf{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta$	CutCtxt
$\mathbf{\Delta}(\alpha)\ \&\ \alpha \vdash_\iota \beta$	fromCtxt
$\mathbf{\Delta}(\alpha)\ \&\ \alpha \vdash_\iota \mathbf{\Delta}(\beta)$	defCons
$\mathbf{\Delta}(\alpha)\ \&\ \alpha \vdash_\iota \mathbf{\Delta}(\beta)\ \&\ \beta$	&-intro

$\mathbf{\Delta}(\alpha)\ \&\ \neg\alpha \vdash_\iota \mathbf{\Delta}(\beta)\ \&\ \neg\beta$	prem
$\mathbf{\Delta}(\alpha)\ \&\ \alpha \vdash_\iota \mathbf{\Delta}(\beta)\ \&\ \beta$	prem
$\mathbf{\Delta}(\alpha)\ \&\ \neg\alpha \vdash_\iota \mathbf{\Delta}(\beta)$	&-elim
$\mathbf{\Delta}(\alpha)\,;\neg\alpha \vdash_\iota \mathbf{\Delta}(\beta)$	toCtxt
$\mathbf{\Delta}(\alpha)\ \&\ \alpha \vdash_\iota \mathbf{\Delta}(\beta)$	&-elim
$\mathbf{\Delta}(\alpha)\,;\alpha \vdash_\iota \mathbf{\Delta}(\beta)$	toCtxt
$\mathbf{\Delta}(\alpha)\,;\vdash_\iota \mathbf{\Delta}(\beta)$	rem
$\mathbf{\Delta}(\alpha) \vdash_\iota \mathbf{\Delta}(\beta)$	fromCtxt

Corollary 26 (Interchange of equivalent subformulae in sequents) *Given two interchangeable formulae α and β. Then*

$$\frac{\Sigma;\Gamma \vdash_\iota A(\alpha)}{\Sigma;\Gamma \vdash_\iota A(\beta)} \qquad \frac{\Sigma;\Gamma, A(\alpha) \vdash_\iota \gamma}{\Sigma;\Gamma, A(\beta) \vdash_\iota \gamma} \qquad \frac{\Sigma_1, A(\alpha), \Sigma_2;\Gamma \vdash_\iota \gamma}{\Sigma_1, A(\beta), \Sigma_2;\Gamma \vdash_\iota \gamma}$$

The analogous theorem for two interchangeable terms also holds.

Proof.

As in the previous theorem, we quickly obtain that the required sequents for corollary 24 are derivable. □

Theorem 27 (replacement under context) *1. Given $\alpha \overset{\Sigma}{\perp} \beta$, then $A(\alpha) \overset{\Sigma}{\perp} A(\beta)$, i.e.,*

$$\begin{cases} \Sigma, \mathbf{\Delta}(A(\alpha))\,; A(\alpha) \vdash_\iota A(\beta) \\ \Sigma, \mathbf{\Delta}(A(\alpha))\,; A(\beta) \vdash_\iota A(\alpha) \end{cases}$$

where $A(\alpha)$ is a formula possibly containing α and $A(\beta)$ is the same formula where a number of instances of α are replaced by β, and $t(\alpha) \overset{\Sigma}{=\!=} t(\beta)$, i.e.,

$$\Sigma; \boldsymbol{\Delta}(t(\alpha)) \vdash_\iota t(\alpha) = t(\beta)$$

where $t(\alpha)$ is a term possibly containing α and $t(\beta)$ is the same term where a number of instances of α are replaced by β.

These results only hold if all of the following restrictions are met:

- *If α is replaced by β inside a $\forall x$ quantifier, then x must not be a free variable of Σ.*
- *The uniqueness conditions for $A(\alpha)$, resp. $t(\alpha)$ must be derivable.*
- *When α is replaced by β inside a ι-term $\iota x_{\psi(\alpha)}(\varphi(\alpha))$, then the uniqueness conditions for both $\iota x_{\psi(\alpha)}(\varphi(\alpha))$ and $\iota x_{\psi(\beta)}(\varphi(\beta))$ must be derivable and x must not be a free variable of Σ.*

2. *Given $t_1 \overset{\Sigma}{=\!=} t_2$ then analogously $A(t_1) \overset{\Sigma}{=\!=} A(t_2)$ and $t(t_1) \overset{\Sigma}{=\!=} t(t_2)$, i.e.,*

$$\begin{cases} \Sigma, \boldsymbol{\Delta}(A(t_1)) \,; A(t_1) \vdash_\iota A(t_2) \\ \Sigma, \boldsymbol{\Delta}(A(t_1)) \,; A(t_2) \vdash_\iota A(t_1) \\ \Sigma; \boldsymbol{\Delta}(t(t_1)) \vdash_\iota t(t_1) = t(t_2) \end{cases}$$

with analogous restrictions as in the first case.

Proof.

Analogous to theorem 23, essentially by adding Σ in front of the context of all sequents involved.

For the case $A(\alpha) \equiv \forall x(B(\alpha))$, the restriction is necessary to be able to apply the $\forall$-intro rule in the proof of theorem 23.

For the case $t(\alpha) \equiv \iota x_{\psi(\alpha)}(\varphi(\alpha))$, the problem is that in general, given $\psi(\alpha) \vdash_\iota \exists! x(\varphi(\alpha))$ we only can derive $\sigma; \psi(\beta) \vdash_\iota \exists! x(\varphi(\beta))$. Given both uniqueness conditions, we can proceed to derive the required sequent in an analogous manner as in theorem 23, where moreover we choose z such that it is not a free variable of σ. There, we also need the condition that x must not be free in σ. □

Note that all restrictions above are necessary:

The restriction on replacements inside $\forall$ quantifiers is motivated by the following counterexample: under the context $x = y$, the terms x and y are interchangeable (the necessary sequents are easily derived). If the theorem would hold in this case, we would obtain $x = y; \forall x(f(x) = x) \vdash_\iota \forall x(f(x) =$

y); that this sequent is unsound can be easily appreciated by noting that the antecedent expresses that the interpretation of f is the identity function, whereas the consequent expresses that f is to be interpreted as a constant function.

If the replacement occurs inside a ι-term, it is indeed possible that the uniqueness conditions of $t(t_1)$ are derivable but not those of $t(t_2)$. Indeed, under the context $x = y$, the terms x and y are interchangeable, but the uniqueness condition of $\iota x(x = y)$ is easily derivable, whereas the uniqueness condition of $\iota x(x = x)$ is not.

Finally, it is necessary to require that x must not be a free variable of Σ in the last restriction. Consider for example in the theory of real numbers the two terms $\iota x_{y\geq 0}(x \cdot y = 1 \,\&\, x \geq 0)$ and $\iota x_{x\geq 0}(x \cdot x = 1 \,\&\, x \geq 0)$. The interpretation of the former is "$\frac{1}{y}$ when $y \geq 0$ and undefined otherwise"; the interpretation of the latter is "1 when $x \geq 0$ and undefined otherwise", so we clearly do not expect

$$x = y \vdash_\iota \iota x_{y\geq 0}(x \cdot y = 1 \,\&\, x \geq 0) = \iota x_{x\geq 0}(x \cdot x = 1 \,\&\, x \geq 0)$$

to be derivable. However, the uniqueness conditions of both ι-terms are derivable, and under $x = y$, the definiens and domain formulae of both ι-terms are interchangeable.

When the replacement occurs inside a ι-term $\iota x_{\psi(\alpha)}\varphi(\alpha)$, we have to supply both the uniqueness condition $\psi(\alpha) \vdash_\iota \exists! x\varphi(\alpha)$ and $\psi(\beta) \vdash_\iota \exists! x\varphi(\beta)$. If we only have the first one, then a sufficient condition for being able to derive the other one is that x be not a free variable of $\Sigma = \sigma_1, \sigma_2, \ldots, \sigma_n$ and that $\psi(\beta) \vdash_\iota \sigma_1 \,\&\, \sigma_2 \,\&\, \cdots \,\&\, \sigma_n$. Indeed, we then can perform the following derivation, choosing z different from x and not occurring free in $\varphi(\beta)$ and Σ. Note that we can easily obtain that $[z/x]\varphi(\alpha)$ is interchangeable with $[z/x]\varphi(\beta)$ under the context Σ, which we will use in the following derivation:

$$\begin{array}{rl}
\psi(\alpha) \vdash_\iota \exists x(\varphi(\alpha)) & \text{\&-elim} \\
\Sigma, \boldsymbol{\Delta}(\exists x(\varphi(\alpha)))\,; \exists x(\varphi(\alpha)) \vdash_\iota \exists x(\varphi(\beta)) & \text{Th. 29} \\
\Sigma; \psi(\alpha) \vdash_\iota \exists x(\varphi(\beta)) & \text{Cut3} \\
\psi(\alpha) \vdash_\iota \forall x \forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & \text{\&-elim} \\
\boldsymbol{\Delta}(\forall x \forall y(\ldots)), \forall x \forall y(\ldots) \vdash_\iota \forall x \forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & \text{AssCtxt} \\
\boldsymbol{\Delta}(\forall x \forall y(\ldots)), \forall x \forall y(\ldots) \vdash_\iota \forall y((\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y) & \forall\text{-elim} \\
\boldsymbol{\Delta}(\forall x \forall y(\ldots)), \forall x \forall y(\ldots) \vdash_\iota (\varphi(\alpha) \,\&\, [y/x]\varphi(\alpha)) \Rightarrow x = y & \forall\text{-elim} \\
\boldsymbol{\Delta}(\forall x \forall y(\ldots)), \forall x \forall y(\ldots) \vdash_\iota (\varphi(\alpha) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z & \text{subst} \\
\Sigma, \boldsymbol{\Delta}(\ldots)\,; & \\
(\varphi(\alpha) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z \vdash_\iota (\varphi(\beta) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z & \text{Th. 29} \\
\Sigma; \boldsymbol{\Delta}(\forall x \forall y(\ldots)), \forall x \forall y(\ldots) \vdash_\iota (\varphi(\beta) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z & \text{Cut3} \\
\Sigma, \boldsymbol{\Delta}(\ldots)\,; &
\end{array}$$

$$
\begin{array}{rl}
(\varphi(\beta) \,\&\, [z/x]\varphi(\alpha)) \Rightarrow x = z \vdash_\iota (\varphi(\beta) \,\&\, [z/x]\varphi(\beta)) \Rightarrow x = z & \text{Th. 29} \\
\Sigma; \boldsymbol{\Delta}(\forall x \forall y(\dots)), \forall x \forall y(\dots) \vdash_\iota (\varphi(\beta) \,\&\, [z/x]\varphi(\beta)) \Rightarrow x = z & \text{Cut3} \\
\Sigma; \boldsymbol{\Delta}(\forall x \forall y(\dots)), \forall x \forall y(\dots) \vdash_\iota \forall z((\varphi(\beta) \,\&\, [z/x]\varphi(\beta)) \Rightarrow x = z) & \forall\text{-intro} \\
\Sigma; \boldsymbol{\Delta}(\forall x \forall y(\dots)), \forall x \forall y(\dots) \vdash_\iota \forall w((\varphi(\beta) \,\&\, [w/x]\varphi(\beta)) \Rightarrow x = w) & \text{RenG} \\
\Sigma; \boldsymbol{\Delta}(\forall x \forall y(\dots)), \forall x \forall y(\dots) \vdash_\iota \forall x \forall w((\varphi(\beta) \,\&\, [w/x]\varphi(\beta)) \Rightarrow x = w) & \forall\text{-intro} \\
\Sigma; \psi(\alpha) \vdash_\iota \exists! x(\varphi(\beta)) & \&\text{-elim} \\
\Sigma, \boldsymbol{\Delta}(\psi(\beta)); \psi(\beta) \vdash_\iota \psi(\alpha) & \text{Th. 29} \\
\Sigma, \boldsymbol{\Delta}(\psi(\beta)); \psi(\beta) \vdash_\iota \exists! x(\varphi(\beta)) & \text{Cut} \\
\vdash_\iota \boldsymbol{\Delta}(\psi(\beta)) & \text{defAnt} \\
\psi(\beta) \vdash_\iota \psi(\beta) & \text{ass} \\
\psi(\beta) \vdash_\iota \boldsymbol{\Delta}(\psi(\beta)) \,\&\, \psi(\beta) & \&\text{-intro} \\
\psi(\beta) \vdash_\iota \sigma \,\&\, \boldsymbol{\Delta}(\psi(\beta)) \,\&\, \psi(\beta) & \&\text{-intro} \\
\Sigma; \boldsymbol{\Delta}(\psi(\beta)) \,\&\, \psi(\beta) \vdash_\iota \exists! x(\varphi(\beta)) & \text{fromCtxt} \\
\Sigma \,\&\, \boldsymbol{\Delta}(\psi(\beta)) \,\&\, \psi(\beta) \vdash_\iota \exists! x(\varphi(\beta)) & \text{fromCtxt} \\
\psi(\beta) \vdash_\iota \exists! x(\varphi(\beta)) & \text{Cut}
\end{array}
$$

Corollary 28 (Replacement under context in sequents) *Given* $\alpha \overset{\Pi}{\triangle} \beta$ *where* Π *is a list of formulae, then*

$$\frac{\Sigma; \Gamma \vdash_\iota A(\alpha)}{\Sigma, \Pi; \Gamma \vdash_\iota A(\beta)} \qquad \frac{\Sigma; \Gamma, A(\alpha) \vdash_\iota \gamma}{\Sigma, \Pi; \Gamma, A(\beta) \vdash_\iota \gamma} \qquad \frac{\Sigma_1, A(\alpha), \Sigma_2; \Gamma \vdash_\iota \gamma}{\Sigma_1, \Pi, A(\beta), \Sigma_2; \Gamma \vdash_\iota \gamma}$$

provided that the following restrictions are obeyed:

- *If* α *is replaced by* β *inside a* $\forall x$ *quantifier, then* x *must not be a free variable of* Π.
- *When* α *is replaced by* β *inside a* ι*-term* $\iota x_{\psi(\alpha)}(\varphi(\alpha))$*, then the uniqueness conditions for* $\iota x_{\psi(\beta)}(\varphi(\beta))$ *must be derivable and* x *must not be a free variable of* Π.

Analogously, given

$$\Pi; \boldsymbol{\Delta}(t_1) \vdash_\iota t_1 = t_2,$$

then

$$\frac{\Sigma; \Gamma \vdash_\iota A(t_1)}{\Sigma, \Pi; \Gamma \vdash_\iota A(t_2)} \qquad \frac{\Sigma; \Gamma, A(t_1) \vdash_\iota \gamma}{\Sigma, \Pi; \Gamma, A(t_2) \vdash_\iota \gamma} \qquad \frac{\Sigma_1, A(t_1), \Sigma_2; \Gamma \vdash_\iota \gamma}{\Sigma_1, \Pi, A(t_2), \Sigma_2; \Gamma \vdash_\iota \gamma}$$

with similar restrictions as above.

Proof.

Analogous to corollary 24; we show the first two cases explicitly:

$$\begin{array}{rr}
\Sigma; \Gamma \vdash_\iota A(\alpha) & \text{prem} \\
\Pi, \boldsymbol{\Delta}(A(\alpha)) \,; A(\alpha) \vdash_\iota A(\beta) & \text{Theorem 27.1} \\
\Sigma, \Pi; \Gamma \vdash_\iota A(\beta) & \text{Cut3}
\end{array}$$

$$\begin{array}{rr}
\Sigma; \Gamma, A(\alpha) \vdash_\iota \gamma & \text{prem} \\
\Pi, \boldsymbol{\Delta}(A(\alpha)) \,; A(\beta) \vdash_\iota A(\alpha) & \text{Theorem 27.1} \\
\Sigma; \vdash_\iota \boldsymbol{\Delta}(A(\alpha)) & \text{defAnt} \\
\Sigma, \Pi; A(\beta) \vdash_\iota A(\alpha) & \text{CutCtxt} \\
\Sigma, \Pi; \Gamma, A(\beta) \vdash_\iota \gamma & \text{Cut}
\end{array}$$

□

Remark that using the WeakCtxtL rule, we can also derive

$$\frac{\Sigma; \Gamma \vdash_\iota A(\alpha)}{\Pi, \Sigma; \Gamma \vdash_\iota A(\beta)} \qquad \frac{\Sigma; \Gamma, A(\alpha) \vdash_\iota \gamma}{\Pi, \Sigma; \Gamma, A(\beta) \vdash_\iota \gamma} \qquad \frac{\Sigma_1, A(\alpha), \Sigma_2; \Gamma \vdash_\iota \gamma}{\Pi, \Sigma_1, A(\beta), \Sigma_2; \Gamma \vdash_\iota \gamma}$$

Theorem 29 (interchange under context) *1. If α and β are interchangeable formulae under the context σ, then $A(\alpha)$ and $A(\beta)$ are interchangeable formulae under the context σ and $t(\alpha)$ and $t(\beta)$ are interchangeable terms under the context σ.*

These results are subject to the following restrictions:

- *If α is replaced by β inside a $\forall x$ quantifier, then x must not be a free variable of σ.*
- *The uniqueness conditions for $A(\alpha)$ or $A(\beta)$, resp. $t(\alpha)$ or $t(\beta)$ must be derivable.*
  We also prove that, except when the replacement occurs inside a ι-term, if the uniqueness conditions for $A(\alpha)$ or $t(\alpha)$ hold, then the uniqueness conditions for $A(\beta)$, resp. $t(\beta)$ hold too and vice versa.
- *When α is replaced by β inside a ι-term $\iota x_{\psi(\alpha)}(\varphi(\alpha))$, then the uniqueness conditions for both $\iota x_{\psi(\alpha)}(\varphi(\alpha))$ and $\iota x_{\psi(\beta)}(\varphi(\beta))$ must be derivable and x must not be a free variable of σ.*

2. If t_1 and t_2 are interchangeable terms under the context σ, then $A(t_1)$ and $A(t_2)$ are interchangeable formulae under the context σ and $t(t_1)$ and $t(t_2)$ are interchangeable terms under the context σ, with similar restrictions as in the first case.

Proof.
Analogous to theorem 25. □

Corollary 30 *Given two formulae α and β which are interchangeable under the context Π. Then*

$$\frac{\Sigma;\Gamma \vdash_\iota A(\alpha)}{\Sigma,\Pi;\Gamma \vdash_\iota A(\beta)} \qquad \frac{\Sigma;\Gamma, A(\alpha) \vdash_\iota \gamma}{\Sigma,\Pi;\Gamma, A(\beta) \vdash_\iota \gamma} \qquad \frac{\Sigma_1, A(\alpha), \Sigma_2;\Gamma \vdash_\iota \gamma}{\Sigma_1,\Pi, A(\beta), \Sigma_2;\Gamma \vdash_\iota \gamma}$$

subject to the same restrictions as corollary 28.

The analogous theorem for two interchangeable terms also holds.

Proof.
Analogous to corollary 26. □

Corollary 31 (Cut rule for interchangeability) *If α and β are interchangeable under the context σ and we have $\psi \vdash_\iota \sigma$, then α and β are interchangeable under the context ψ. (The analogous property for t_1 and t_2 holds too.)*

Proof.
From the interchangeability conditions for the interchangeability of α and β under σ, we get the required interchangeability conditions for interchangeability under ψ using the CutCtxt rule. For example,

$\sigma, \mathbf{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta$	interchangeability
$\psi \vdash_\iota \sigma$	prem
$\psi, \mathbf{\Delta}(\alpha)\,;\alpha \vdash_\iota \beta$	CutCtxt

□

Theorem 32 *If α and β, resp. t_1 and t_2, are interchangeable under the context σ, then so are $\mathbf{\Delta}(\alpha)$ and $\mathbf{\Delta}(\beta)$, resp. $\mathbf{\Delta}(t_1)$ and $\mathbf{\Delta}(t_2)$.*

Proof.
We have to derive

$$\begin{cases} \sigma, \mathbf{\Delta}(\mathbf{\Delta}(\alpha))\,;\mathbf{\Delta}(\alpha) \vdash_\iota \mathbf{\Delta}(\beta) \\ \sigma, \mathbf{\Delta}(\mathbf{\Delta}(\beta))\,;\mathbf{\Delta}(\beta) \vdash_\iota \mathbf{\Delta}(\alpha) \\ \sigma;\mathbf{\Delta}(\mathbf{\Delta}(\alpha)) \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\beta)) \\ \sigma;\mathbf{\Delta}(\mathbf{\Delta}(\beta)) \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\alpha)) \end{cases}$$

$$\begin{cases} \sigma, \boldsymbol{\Delta}(\boldsymbol{\Delta}(t_1))\,;\boldsymbol{\Delta}(t_1) \vdash_\iota \boldsymbol{\Delta}(t_2) \\ \sigma, \boldsymbol{\Delta}(\boldsymbol{\Delta}(t_2))\,;\boldsymbol{\Delta}(t_2) \vdash_\iota \boldsymbol{\Delta}(t_1) \\ \qquad \sigma; \boldsymbol{\Delta}(\boldsymbol{\Delta}(t_1)) \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(t_2)) \\ \qquad \sigma; \boldsymbol{\Delta}(\boldsymbol{\Delta}(t_2)) \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(t_1)) \end{cases}$$

which is easy. □

Theorem 33 *For each formula α and term t of the* PITFOL *calculus, if the substitution $[t/x]\alpha$ is defined, then also the substitution $[t/x]\boldsymbol{\Delta}(\alpha)$ is defined.*

If the uniqueness conditions for α and t are derivable, then $\boldsymbol{\Delta}([t/x]\alpha) \vdash_\iota [t/x]\boldsymbol{\Delta}(\alpha)$.

The analogous theorem for terms τ of the PITFOL *calculus also holds.*

Proof.

By induction on the complexity of τ and α.

- $\tau \equiv x$ The consequent of the sequent to derive is $\top$; hence, by convention, we have nothing to derive.
- $\tau \equiv y$ with $y \not\equiv x$ We have nothing to derive.
- $\tau \equiv f(t_1, t_2, \ldots, t_n)$ We have to derive

$$\boldsymbol{\Delta}([t/x]t_1) \,\&\, \boldsymbol{\Delta}([t/x]t_2) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}([t/x]t_n) \vdash_\iota [t/x]\boldsymbol{\Delta}(t_1) \,\&\, [t/x]\boldsymbol{\Delta}(t_2) \,\&\, \cdots \,\&\, [t/x]\boldsymbol{\Delta}(t_n)$$

  which is easy by applying induction on $t_1, t_2, \ldots, t_n$.

- $\tau \equiv \iota y_\psi(\varphi)$
  - $x \equiv y$ or x is not a free variable of φ If x is not a free variable of ψ, then we have to derive $\psi \vdash_\iota \psi$, which is easy. Else, we have to derive $\boldsymbol{\Delta}(t) \,\&\, [t/x]\psi \vdash_\iota [t/x]\psi$:

$$\begin{array}{rr} \vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\ \psi \vdash_\iota \psi & \text{ass} \\ \boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota [t/x]\psi & \text{subst} \\ \boldsymbol{\Delta}(t) \,\&\, [t/x]\psi \vdash_\iota [t/x]\psi & \text{fromCtxt} \end{array}$$

  - $x \not\equiv y$ and x is a free variable of φ We again have to derive $\boldsymbol{\Delta}(t) \,\&\, [t/x]\psi \vdash_\iota [t/x]\psi$, cfr. supra.
- $\alpha \equiv p(t_1, t_2, \ldots, t_n)$ Analogous to the case $\tau \equiv f(t_1, t_2, \ldots, t_n)$.

- $\alpha \equiv \beta \,\&\, \gamma$

$$\begin{array}{rr}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma))) & \text{defAnt} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma)) \vdash_\iota \boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma)) & \text{ass} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma)) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta) & \&\text{-elim} \\
\boldsymbol{\Delta}([t/x]\beta) \vdash_\iota [t/x]\boldsymbol{\Delta}(\beta) & \text{induction} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma)) \vdash_\iota [t/x]\boldsymbol{\Delta}(\beta) & \&\text{-elim} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma)) \vdash_\iota [t/x]\beta \Rightarrow \boldsymbol{\Delta}([t/x]\gamma) & \&\text{-elim} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma))\,;[t/x]\beta \vdash_\iota \boldsymbol{\Delta}([t/x]\gamma) & \text{DdRu1} \\
\boldsymbol{\Delta}([t/x]\gamma) \vdash_\iota [t/x]\boldsymbol{\Delta}(\gamma) & \text{induction} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma))\,;[t/x]\beta \vdash_\iota [t/x]\boldsymbol{\Delta}(\gamma) & \text{Cut} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma))\,;\vdash_\iota [t/x]\beta \Rightarrow [t/x]\boldsymbol{\Delta}(\gamma) & \text{DdRu2} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma)) \vdash_\iota [t/x]\beta \Rightarrow [t/x]\boldsymbol{\Delta}(\gamma) & \text{fromCtxt} \\
\boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma)) \vdash_\iota [t/x]\boldsymbol{\Delta}(\beta \,\&\, \gamma) & \&\text{-intro}
\end{array}$$

- $\alpha \equiv \neg\beta$ Induction on β immediately yields the required sequent.
- $\alpha \equiv \forall y(\beta)$ If $y \equiv x$ then we have to derive $\boldsymbol{\Delta}(\forall x(\beta)) \vdash_\iota \boldsymbol{\Delta}(\forall x(\beta))$, which is easy; else, induction on β yields $\boldsymbol{\Delta}([t/x]\beta) \vdash_\iota [t/x]\boldsymbol{\Delta}(\beta)$ and using the SimGen rule, we obtain the required sequent.

□

Using this theorem, we can derive the following rule.

$$\begin{array}{rr}
\boxed{\text{PartCons3}} & \\
\Sigma;\Gamma \vdash_\iota [t/x]\alpha & \text{prem} \\
\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}([t/x]\alpha) & \text{defCons} \\
\boldsymbol{\Delta}([t/x]\alpha) \vdash_\iota [t/x]\boldsymbol{\Delta}(\alpha) & \text{Th. 33} \\
\Sigma;\Gamma \vdash_\iota [t/x]\boldsymbol{\Delta}(\alpha) & \text{Cut} \\
\Sigma;\Gamma \vdash_\iota [t/x]\boldsymbol{\Delta}(\alpha) \,\&\, [t/x]\alpha & \&\text{-intro} \\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha)) & \text{Ddef} \\
\boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{ass} \\
\vdash_\iota \boldsymbol{\Delta}(\alpha) \Rightarrow \boldsymbol{\Delta}(\alpha) & \text{DdRu2} \\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha) & \&\text{-intro} \\
\vdash_\iota \forall x(\boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha)) & \forall\text{-intro} \\
\forall x(\neg(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha)) \vdash_\iota \forall x(\neg(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha)) & \text{ass} \\
\forall x(\neg(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha)) \vdash_\iota \neg(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha) & \forall\text{-elim} \\
\boldsymbol{\Delta}(t)\,;\forall x(\neg(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha)) \vdash_\iota \neg([t/x]\boldsymbol{\Delta}(\alpha) \,\&\, [t/x]\alpha) & \text{subst} \\
\Sigma, \boldsymbol{\Delta}(t)\,;\Gamma, \forall x(\neg(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha)) \vdash_\iota \neg\forall x(\neg(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha)) & \text{contra} \\
\Sigma, \boldsymbol{\Delta}(t)\,;\Gamma \vdash_\iota \exists x(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha) & \text{SeDe}
\end{array}$$

Note that just as in the PartCons rule, $\boldsymbol{\Delta}(\alpha)$ is needed in the conclusion. For a counterexample, consider the valid sequent $x \neq 0 \vdash_\iota \frac{1}{\frac{1}{x}} \neq 0$, from which we

cannot deduce $x \neq 0 \vdash_\iota \exists x(\frac{1}{x} \neq 0)$—otherwise the defCons rule would yield $x \neq 0 \vdash_\iota \forall x(x \neq 0)$.

Property 34 *Associativity of the conjunction*

Proof.

We will prove that the & operator is associative, i.e., we may substitute $\alpha \,\&\, (\beta \,\&\, \gamma)$ for $(\alpha \,\&\, \beta) \,\&\, \gamma$ and vice versa. We will derive the four sequents comprising the interchangeability conditions.

$$\begin{array}{rr}
\mathbf{\Delta}(\alpha \,\&\, (\beta \,\&\, \gamma))\,;\alpha \,\&\, (\beta \,\&\, \gamma) \vdash_\iota \alpha \,\&\, (\beta \,\&\, \gamma) & \text{AssCtxt} \\
\mathbf{\Delta}(\alpha \,\&\, (\beta \,\&\, \gamma))\,;\alpha \,\&\, (\beta \,\&\, \gamma) \vdash_\iota (\alpha \,\&\, \beta) \,\&\, \gamma & \text{AssocConj1}
\end{array}$$

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\alpha \,\&\, (\beta \,\&\, \gamma))) & \text{Ddef} \\
\mathbf{\Delta}(\alpha \,\&\, (\beta \,\&\, \gamma)) \vdash_\iota \mathbf{\Delta}(\alpha \,\&\, (\beta \,\&\, \gamma)) & \text{ass} \\
\mathbf{\Delta}(\alpha \,\&\, (\beta \,\&\, \gamma)) \vdash_\iota \mathbf{\Delta}((\alpha \,\&\, \beta) \,\&\, \gamma) & \text{DefAssocConj1}
\end{array}$$

The derivation of the other two sequents is similar. □

Property 35 *Commutativity of the conjunction* $\alpha \,\&\, \beta$ *when* $\mathbf{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \mathbf{\Delta}(\beta)$ *and* $\mathbf{\Delta}(\beta \,\&\, \alpha) \vdash_\iota \mathbf{\Delta}(\alpha)$ *are given*

Proof.

$$\begin{array}{rr}
\mathbf{\Delta}(\alpha \,\&\, \beta)\,;\alpha \,\&\, \beta \vdash_\iota \alpha \,\&\, \beta & \text{AssCtxt} \\
\mathbf{\Delta}(\alpha \,\&\, \beta)\,;\alpha \,\&\, \beta \vdash_\iota \alpha & \text{\&-elim} \\
\mathbf{\Delta}(\alpha \,\&\, \beta)\,;\alpha \,\&\, \beta \vdash_\iota \beta & \text{\&-elim} \\
\mathbf{\Delta}(\alpha \,\&\, \beta)\,;\alpha \,\&\, \beta \vdash_\iota \beta \,\&\, \alpha & \text{\&-intro}
\end{array}$$

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\alpha \,\&\, \beta)) & \text{Ddef} \\
\mathbf{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \mathbf{\Delta}(\alpha \,\&\, \beta) & \text{ass} \\
\mathbf{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \mathbf{\Delta}(\alpha) & \text{\&-elim} \\
\mathbf{\Delta}(\alpha \,\&\, \beta)\,;\vdash_\iota \mathbf{\Delta}(\beta) & \text{toCtxt} \\
\mathbf{\Delta}(\alpha \,\&\, \beta)\,;\beta \vdash_\iota \mathbf{\Delta}(\alpha) & \text{Weak} \\
\mathbf{\Delta}(\alpha \,\&\, \beta)\,;\vdash_\iota \beta \Rightarrow \mathbf{\Delta}(\alpha) & \text{DdRu2} \\
\mathbf{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \beta \Rightarrow \mathbf{\Delta}(\alpha) & \text{fromCtxt} \\
\mathbf{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \mathbf{\Delta}(\beta \,\&\, \alpha) & \text{\&-intro}
\end{array}$$

Analogously, we derive $\mathbf{\Delta}(\beta \,\&\, \alpha)\,;\beta \,\&\, \alpha \vdash_\iota \alpha \,\&\, \beta$ and $\mathbf{\Delta}(\beta \,\&\, \alpha) \vdash_\iota \mathbf{\Delta}(\alpha \,\&\, \beta)$. □

Note that the conditions given are equivalent with $\boldsymbol{\Delta}(\alpha \,\&\, \beta) \dashv\vdash_\iota \boldsymbol{\Delta}(\beta \,\&\, \alpha)$:

$$\begin{array}{rl} \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\beta \,\&\, \alpha) & \text{prem} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{\&-elim} \end{array}$$

$$\begin{array}{rr} \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{prem} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta)\,; \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{toCtxt} \\ \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha \,\&\, \beta)) & \text{Ddef} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, \beta) & \text{ass} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{\&-elim} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta)\,; \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{toCtxt} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta)\,; \beta \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{Weak} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta)\,; \vdash_\iota \beta \Rightarrow \boldsymbol{\Delta}(\alpha) & \text{DdRu2} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \beta \Rightarrow \boldsymbol{\Delta}(\alpha) & \text{fromCtxt} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\beta \,\&\, \alpha) & \text{\&-intro} \end{array}$$

The sequent $\boldsymbol{\Delta}(\beta \,\&\, \alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, \beta)$ is derived analogously.

Note that instead of requiring $\boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\beta)$, we can equivalently require $\boldsymbol{\Delta}(\alpha)\,; \neg\alpha \vdash_\iota \boldsymbol{\Delta}(\beta)$. Indeed, we have

$$\begin{array}{rr} \boldsymbol{\Delta}(\alpha)\,; \alpha \vdash_\iota \alpha & \text{AssCtxt} \\ \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha)) & \text{Ddef} \\ \boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{ass} \\ \boldsymbol{\Delta}(\alpha)\,; \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{toCtxt} \\ \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\beta)) & \text{Ddef} \\ \boldsymbol{\Delta}(\alpha)\,; \alpha \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\beta)) & \text{Weak*} \\ \boldsymbol{\Delta}(\alpha)\,; \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\boldsymbol{\Delta}(\beta)) & \text{DdRu2} \\ \boldsymbol{\Delta}(\alpha)\,; \vdash_\iota \boldsymbol{\Delta}(\alpha \Rightarrow \boldsymbol{\Delta}(\beta)) & \text{\&-intro} \\ \boldsymbol{\Delta}(\alpha)\,; \alpha \Rightarrow \boldsymbol{\Delta}(\beta) \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta) & \text{ass} \\ \boldsymbol{\Delta}(\alpha)\,; \alpha, \alpha \Rightarrow \boldsymbol{\Delta}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{MP} \\ \boldsymbol{\Delta}(\alpha)\,; \neg\alpha \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{given} \\ \boldsymbol{\Delta}(\alpha)\,; \alpha \Rightarrow \boldsymbol{\Delta}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{rem} \\ \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{fromCtxt} \end{array}$$

and

$$\begin{array}{rr} \boldsymbol{\Delta}(\alpha \,\&\, \beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{given} \\ \boldsymbol{\Delta}(\alpha)\,; \alpha \Rightarrow \boldsymbol{\Delta}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{toCtxt} \\ \boldsymbol{\Delta}(\alpha)\,; \neg\alpha \vdash_\iota \neg\alpha & \text{AssCtxt} \\ \boldsymbol{\Delta}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{ass} \\ \boldsymbol{\Delta}(\alpha)\,; \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{toCtxt} \end{array}$$

$$\begin{array}{rr}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\beta)) & \text{Ddef} \\
\boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \alpha & \text{AssCtxt} \\
\boldsymbol{\Delta}(\alpha)\,;\alpha \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\beta)) & \text{Weak*} \\
\boldsymbol{\Delta}(\alpha)\,;\vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\boldsymbol{\Delta}(\beta)) & \text{DdRu2} \\
\boldsymbol{\Delta}(\alpha)\,;\vdash_\iota \boldsymbol{\Delta}(\alpha \,\&\, \neg\boldsymbol{\Delta}(\beta)) & \text{\&-intro} \\
\boldsymbol{\Delta}(\alpha)\,;\alpha \,\&\, \neg\boldsymbol{\Delta}(\beta) \vdash_\iota \alpha \,\&\, \neg\boldsymbol{\Delta}(\beta) & \text{ass} \\
\boldsymbol{\Delta}(\alpha)\,;\alpha \,\&\, \neg\boldsymbol{\Delta}(\beta) \vdash_\iota \alpha & \text{\&-elim} \\
\boldsymbol{\Delta}(\alpha)\,;\neg\alpha, \alpha \,\&\, \neg\boldsymbol{\Delta}(\beta) \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta) & \text{contra} \\
\boldsymbol{\Delta}(\alpha)\,;\neg\alpha \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta) & \text{SeAs} \\
\boldsymbol{\Delta}(\alpha)\,;\neg\alpha \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{Cut}
\end{array}$$

Finally, we remark that it is not necessary that α or β is always defined to apply the commutative property. For example,

$$\forall x(x = x) \,\&\, y = \frac{1}{x} \rightleftharpoons y = \frac{1}{x} \,\&\, \forall x(x = x)$$
$$x = 0 \,\&\, y = \frac{1}{x} \rightleftharpoons y = \frac{1}{x} \,\&\, x = 0$$

where $\frac{1}{x}$ is as usual shorthand for $\iota z_{x\neq 0}(x \cdot z = 1)$.

Property 36 *Commutativity of the disjunction $\alpha \vee \beta$ when $\boldsymbol{\Delta}(\alpha \vee \beta) \vdash_\iota \boldsymbol{\Delta}(\beta)$ and $\boldsymbol{\Delta}(\beta \vee \alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha)$ are given*

Proof.

$$\begin{array}{rr}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha \vee \beta)) & \text{Ddef} \\
\boldsymbol{\Delta}(\alpha \vee \beta) \vdash_\iota \boldsymbol{\Delta}(\alpha \vee \beta) & \text{ass} \\
\boldsymbol{\Delta}(\alpha \vee \beta) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{\&-elim} \\
\boldsymbol{\Delta}(\alpha \vee \beta)\,;\vdash_\iota \boldsymbol{\Delta}(\beta) & \text{toCtxt} \\
\boldsymbol{\Delta}(\alpha \vee \beta)\,,\neg\beta \vdash_\iota \boldsymbol{\Delta}(\alpha) & \text{Weak} \\
\boldsymbol{\Delta}(\alpha \vee \beta) \vdash_\iota \neg\beta \Rightarrow \boldsymbol{\Delta}(\alpha) & \text{DdRu2} \\
\boldsymbol{\Delta}(\alpha \vee \beta) \vdash_\iota \boldsymbol{\Delta}(\beta \vee \alpha) & \text{\&-intro}
\end{array}$$

We use this result to derive

$$\begin{array}{rr}
\boldsymbol{\Delta}(\alpha \vee \beta)\,;\alpha \vee \beta \vdash_\iota \alpha \vee \beta & \text{AssCtxt} \\
\boldsymbol{\Delta}(\neg\beta \,\&\, \neg\alpha)\,;\neg\beta \,\&\, \neg\alpha \vdash_\iota \neg\beta \,\&\, \neg\alpha & \text{AssCtxt} \\
\boldsymbol{\Delta}(\neg\beta \,\&\, \neg\alpha)\,;\neg\beta \,\&\, \neg\alpha \vdash_\iota \neg\beta & \text{\&-elim} \\
\boldsymbol{\Delta}(\neg\beta \,\&\, \neg\alpha)\,;\neg\beta \,\&\, \neg\alpha \vdash_\iota \neg\alpha & \text{\&-elim} \\
\boldsymbol{\Delta}(\neg\beta \,\&\, \neg\alpha)\,;\neg\beta \,\&\, \neg\alpha \vdash_\iota \neg\alpha \,\&\, \neg\beta & \text{\&-intro} \\
\boldsymbol{\Delta}(\alpha \vee \beta)\,;\neg\beta \,\&\, \neg\alpha \vdash_\iota \neg\alpha \,\&\, \neg\beta & \text{CutCtxt} \\
\boldsymbol{\Delta}(\alpha \vee \beta)\,;\alpha \vee \beta, \neg\beta \,\&\, \neg\alpha \vdash_\iota \beta \vee \alpha & \text{contra} \\
\boldsymbol{\Delta}(\alpha \vee \beta)\,;\alpha \vee \beta \vdash_\iota \beta \vee \alpha & \text{SeDe}
\end{array}$$

Analogously, we derive $\mathbf{\Delta}(\beta \vee \alpha) \vdash_\iota \mathbf{\Delta}(\alpha \vee \beta)$ and $\mathbf{\Delta}(\beta \vee \alpha)\,;\beta \vee \alpha \vdash_\iota \alpha \vee \beta$. □

Analogously to the commutativity of the conjunction, one shows that the conditions given are equivalent with $\mathbf{\Delta}(\alpha \vee \beta) \dashv\vdash_\iota \mathbf{\Delta}(\beta \vee \alpha)$ and that instead of requiring $\mathbf{\Delta}(\alpha \vee \beta) \vdash_\iota \mathbf{\Delta}(\beta)$, we can equivalently require $\mathbf{\Delta}(\alpha)\,;\alpha \vdash_\iota \mathbf{\Delta}(\beta)$.

Property 37 *Interchangeability of* α *and* $\neg\neg\alpha$

Proof.

$$\begin{array}{rr} \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\alpha)) & \text{Ddef} \\ \mathbf{\Delta}(\alpha) \vdash_\iota \mathbf{\Delta}(\alpha) & \text{ass} \\ \mathbf{\Delta}(\alpha)\,; \vdash_\iota \mathbf{\Delta}(\alpha) & \text{toCtxt} \\ \mathbf{\Delta}(\alpha)\,;\alpha \vdash_\iota \neg\neg\alpha & \text{NN1} \end{array}$$

$$\begin{array}{rr} \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\alpha)) & \text{Ddef} \\ \mathbf{\Delta}(\alpha) \vdash_\iota \mathbf{\Delta}(\alpha) & \text{ass} \\ \mathbf{\Delta}(\alpha)\,; \vdash_\iota \mathbf{\Delta}(\alpha) & \text{toCtxt} \\ \mathbf{\Delta}(\alpha)\,;\neg\neg\alpha \vdash_\iota \alpha & \text{NN2} \end{array}$$

Finally, we have to derive $\mathbf{\Delta}(\alpha) \dashv\vdash_\iota \mathbf{\Delta}(\alpha)$, which is trivial. □

Property 38 *Interchangeability of* $\forall x(\alpha \,\&\, \beta)$ *and* $\forall x(\alpha) \,\&\, \forall x(\beta)$

These can be only interchanged if we first also derive

$$\mathbf{\Delta}(\forall x(\alpha) \,\&\, \forall x(\beta)) \vdash_\iota \alpha \Rightarrow \mathbf{\Delta}(\beta) \qquad (*)$$

Proof.

Again, we prove the four sequents necessary for an application of corollary 26.

$$\begin{array}{rr} \mathbf{\Delta}(\forall x(\alpha \,\&\, \beta))\,;\forall x(\alpha \,\&\, \beta) \vdash_\iota \forall x(\alpha \,\&\, \beta) & \text{AssCtxt} \\ \mathbf{\Delta}(\forall x(\alpha \,\&\, \beta))\,;\forall x(\alpha \,\&\, \beta) \vdash_\iota \alpha \,\&\, \beta & \forall\text{-elim} \\ \mathbf{\Delta}(\forall x(\alpha \,\&\, \beta))\,;\forall x(\alpha \,\&\, \beta) \vdash_\iota \alpha & \&\text{-elim} \\ \mathbf{\Delta}(\forall x(\alpha \,\&\, \beta))\,;\forall x(\alpha \,\&\, \beta) \vdash_\iota \beta & \&\text{-elim} \\ \mathbf{\Delta}(\forall x(\alpha \,\&\, \beta))\,;\forall x(\alpha \,\&\, \beta) \vdash_\iota \forall x(\alpha) & \forall\text{-intro} \\ \mathbf{\Delta}(\forall x(\alpha \,\&\, \beta))\,;\forall x(\alpha \,\&\, \beta) \vdash_\iota \forall x(\beta) & \forall\text{-intro} \\ \mathbf{\Delta}(\forall x(\alpha \,\&\, \beta))\,;\forall x(\alpha \,\&\, \beta) \vdash_\iota \forall x(\alpha) \,\&\, \forall x(\beta) & \&\text{-intro} \end{array}$$

$$\begin{array}{rl}
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\forall x(\alpha)\,\&\,\forall x(\beta) \vdash_\iota \forall x(\alpha)\,\&\,\forall x(\beta) & \text{AssCtxt}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\forall x(\alpha)\,\&\,\forall x(\beta) \vdash_\iota \forall x(\alpha) & \&\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\forall x(\alpha)\,\&\,\forall x(\beta) \vdash_\iota \forall x(\beta) & \&\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\forall x(\alpha)\,\&\,\forall x(\beta) \vdash_\iota \alpha & \forall\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\forall x(\alpha)\,\&\,\forall x(\beta) \vdash_\iota \beta & \forall\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\forall x(\alpha)\,\&\,\forall x(\beta) \vdash_\iota \alpha\,\&\,\beta & \&\text{-intro}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\forall x(\alpha)\,\&\,\forall x(\beta) \vdash_\iota \forall x(\alpha\,\&\,\beta) & \forall\text{-intro}
\end{array}$$

$$\begin{array}{rl}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta))) & \text{Ddef}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta)) \vdash_\iota \boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta)) & \text{ass}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta)) \vdash_\iota \boldsymbol{\Delta}(\alpha\,\&\,\beta) & \forall\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta)) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \&\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta)) \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta) & \&\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta)) \vdash_\iota \forall x(\boldsymbol{\Delta}(\alpha)) & \forall\text{-intro}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta))\,;\alpha \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{DdRu1}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta))\,;\forall x(\alpha) \vdash_\iota \forall x(\boldsymbol{\Delta}(\beta)) & \text{SimGen}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta))\,;\vdash_\iota \forall x(\alpha) \Rightarrow \forall x(\boldsymbol{\Delta}(\beta)) & \text{DdRu2}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta)) \vdash_\iota \forall x(\alpha) \Rightarrow \forall x(\boldsymbol{\Delta}(\beta)) & \text{fromCtxt}\\
\boldsymbol{\Delta}(\forall x(\alpha\,\&\,\beta)) \vdash_\iota \boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) & \text{fromCtxt}
\end{array}$$

$$\begin{array}{rl}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))) & \text{Ddef}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) & \text{ass}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \boldsymbol{\Delta}(\forall x(\alpha)) & \&\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \boldsymbol{\Delta}(\alpha) & \forall\text{-elim}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \alpha \Rightarrow \boldsymbol{\Delta}(\beta) & (*)\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \boldsymbol{\Delta}(\alpha\,\&\,\beta) & \&\text{-intro}\\
\boldsymbol{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \forall x(\boldsymbol{\Delta}(\alpha\,\&\,\beta)) & \forall\text{-intro}
\end{array}$$

For the last proof, we need the extra sequent $(*)$: we can derive $\ldots \vdash_\iota \forall x(\alpha) \Rightarrow \boldsymbol{\Delta}(\forall x(\beta))$ but we need $\ldots \vdash_\iota \forall x(\alpha \Rightarrow \boldsymbol{\Delta}(\beta))$. □

Note that we cannot dispense of the sequent $(*)$. Consider the case $\alpha \equiv x = 0$ and $\beta \equiv \frac{1}{x} = 5$ with the usual notations. In this case, $\forall x(\alpha\,\&\,\beta)$ is defined when $\forall x(x = 0 \Rightarrow x \neq 0)$, which is an invalid formula in any interpretation, but $\forall x(\alpha)\,\&\,\forall x(\beta)$ is defined when $\forall x(x = 0) \Rightarrow \forall x(x \neq 0)$, which is a validity in the theory of real numbers. In this case, $(*)$ is the sequent

$$\forall x(x = 0) \Rightarrow \forall x(x \neq 0) \vdash_\iota x = 0 \Rightarrow x \neq 0$$

which is clearly underivable.

In the sequel, we will apply this property in situations where β is of the form $\mathbf{\Delta}(\gamma)$. We will show that in this case, the extra sequent $(*)$ can always be deduced:

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))) & \text{Ddef}\\
\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) & \text{ass}\\
\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \mathbf{\Delta}(\forall x(\alpha)) & \&\text{-elim}\\
\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \mathbf{\Delta}(\alpha) & \forall\text{-elim}\\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\gamma)) & \text{Ddef}\\
\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\vdash_\iota \mathbf{\Delta}(\alpha) & \text{toCtxt}\\
\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\alpha \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\gamma)) & \text{Weak}\\
\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\vdash_\iota \alpha \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(\gamma)) & \text{DdRu2}\\
\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))\,;\vdash_\iota \alpha \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(\gamma)) & \text{fromCtxt}
\end{array}$$

When α is of the form $\mathbf{\Delta}(\gamma)$ and β of the form $\gamma \Rightarrow \mathbf{\Delta}(\delta)$, the extra sequent $(*)$ can always be deduced, too:

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\delta)) & \text{Ddef}\\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\gamma)) & \text{Ddef}\\
\mathbf{\Delta}(\gamma) \vdash_\iota \mathbf{\Delta}(\gamma) & \text{ass}\\
\mathbf{\Delta}(\gamma)\,;\vdash_\iota \mathbf{\Delta}(\gamma) & \text{toCtxt}\\
\mathbf{\Delta}(\gamma)\,;\gamma \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\delta)) & \text{Weak}\\
\mathbf{\Delta}(\gamma)\,;\vdash_\iota \gamma \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(\delta)) & \text{DdRu2}\\
\mathbf{\Delta}(\gamma) \vdash_\iota \gamma \Rightarrow \mathbf{\Delta}(\mathbf{\Delta}(\delta)) & \text{fromCtxt}\\
\mathbf{\Delta}(\gamma) \vdash_\iota \mathbf{\Delta}(\gamma \Rightarrow \mathbf{\Delta}(\delta)) & \&\text{-intro}\\
\vdash_\iota \mathbf{\Delta}(\gamma) \Rightarrow \mathbf{\Delta}(\gamma \Rightarrow \mathbf{\Delta}(\delta)) & \text{DdRu2}\\
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta))) & \text{Ddef}\\
\mathbf{\Delta}(\forall x(\alpha)\,\&\,\forall x(\beta)) \vdash_\iota \mathbf{\Delta}(\gamma) \Rightarrow \mathbf{\Delta}(\gamma \Rightarrow \mathbf{\Delta}(\delta)) & \text{Weak}
\end{array}$$

Property 39 *Interchangeability of α and $\forall x(\alpha)$*

Proof.

This requires that x is not a free variable of α. The four required sequents can be deduced easily. □

Property 40 *Interchangeability of $\forall x \forall y(\alpha)$ and $\forall y \forall x(\alpha)$*

Proof.

$$\begin{array}{rr}
\mathbf{\Delta}(\forall x \forall y(\alpha))\,;\forall x \forall y(\alpha) \vdash_\iota \forall x \forall y(\alpha) & \text{AssCtxt}\\
\mathbf{\Delta}(\forall x \forall y(\alpha))\,;\forall x \forall y(\alpha) \vdash_\iota \forall y(\alpha) & \forall\text{-elim}\\
\mathbf{\Delta}(\forall x \forall y(\alpha))\,;\forall x \forall y(\alpha) \vdash_\iota \alpha & \forall\text{-elim}\\
\mathbf{\Delta}(\forall x \forall y(\alpha))\,;\forall x \forall y(\alpha) \vdash_\iota \forall x(\alpha) & \forall\text{-intro}\\
\mathbf{\Delta}(\forall x \forall y(\alpha))\,;\forall x \forall y(\alpha) \vdash_\iota \forall y(\forall x(\alpha)) & \forall\text{-intro}
\end{array}$$

The deductions of the other three required sequents are similar. □

Property 41 *Interchangeability of* $\forall x(\alpha \Rightarrow \beta)$ *and* $\exists x(\alpha) \Rightarrow \beta$

Proof.

This requires that x is not a free variable of β.

$$\begin{array}{rl}
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,;\forall x(\alpha \Rightarrow \beta) \vdash_\iota \forall x(\alpha \Rightarrow \beta) & \text{AssCtxt}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,;\forall x(\alpha \Rightarrow \beta) \vdash_\iota \alpha \Rightarrow \beta & \forall\text{-elim}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,,\forall x(\alpha \Rightarrow \beta);\alpha \vdash_\iota \beta & \text{DdRu1}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,,\forall x(\alpha \Rightarrow \beta);\exists x(\alpha) \vdash_\iota \beta & \text{PartAnt}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,,\forall x(\alpha \Rightarrow \beta);\vdash_\iota \exists x(\alpha) \Rightarrow \beta & \text{DdRu2}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,;\forall x(\alpha \Rightarrow \beta) \vdash_\iota \exists x(\alpha) \Rightarrow \beta & \text{fromCtxt}
\end{array}$$

$$\begin{array}{rl}
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,;\exists x(\alpha) \Rightarrow \beta \vdash_\iota \exists x(\alpha) \Rightarrow \beta & \text{AssCtxt}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,,\exists x(\alpha) \Rightarrow \beta;\exists x(\alpha) \vdash_\iota \beta & \text{DdRu1}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,,\exists x(\alpha) \Rightarrow \beta;\alpha \vdash_\iota \beta & \exists\text{-ElimAnt}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,,\exists x(\alpha) \Rightarrow \beta;\vdash_\iota \alpha \Rightarrow \beta & \text{DdRu2}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,;\exists x(\alpha) \Rightarrow \beta \vdash_\iota \alpha \Rightarrow \beta & \text{fromCtxt}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,;\exists x(\alpha) \Rightarrow \beta \vdash_\iota \forall x(\alpha \Rightarrow \beta) & \forall\text{-intro}
\end{array}$$

$$\begin{array}{rl}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))) & \text{Ddef}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta)) \vdash_\iota \mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta)) & \text{ass}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta)) \vdash_\iota \mathbf{\Delta}(\alpha \Rightarrow \beta) & \forall\text{-elim}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta)) \vdash_\iota \mathbf{\Delta}(\alpha) & \&\text{-elim}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta)) \vdash_\iota \alpha \Rightarrow \mathbf{\Delta}(\beta) & \&\text{-elim}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta)) \vdash_\iota \mathbf{\Delta}(\forall x(\alpha)) & \forall\text{-intro}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,;\alpha \vdash_\iota \mathbf{\Delta}(\beta) & \text{DdRu1}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,;\exists x(\alpha) \vdash_\iota \mathbf{\Delta}(\beta) & \text{PartAnt}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta))\,;\vdash_\iota \exists x(\alpha) \Rightarrow \mathbf{\Delta}(\beta) & \text{DdRu2}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta)) \vdash_\iota \exists x(\alpha) \Rightarrow \mathbf{\Delta}(\beta) & \text{fromCtxt}\\
\mathbf{\Delta}(\forall x(\alpha \Rightarrow \beta)) \vdash_\iota \mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta) & \text{fromCtxt}
\end{array}$$

$$\begin{array}{rl}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)) & \text{Ddef}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta) \vdash_\iota \mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta) & \text{ass}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta) \vdash_\iota \mathbf{\Delta}(\forall x(\alpha)) & \&\text{-elim}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta) \vdash_\iota \exists x(\alpha) \Rightarrow \mathbf{\Delta}(\beta) & \&\text{-elim}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,;\exists x(\alpha) \vdash_\iota \mathbf{\Delta}(\beta) & \text{DdRu1}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,;\alpha \vdash_\iota \mathbf{\Delta}(\beta) & \exists\text{-ElimAnt}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta)\,;\vdash_\iota \alpha \Rightarrow \mathbf{\Delta}(\beta) & \text{DdRu2}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta) \vdash_\iota \mathbf{\Delta}(\alpha) & \forall\text{-elim}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta) \vdash_\iota \mathbf{\Delta}(\alpha \Rightarrow \beta) & \&\text{-intro}\\
\mathbf{\Delta}(\exists x(\alpha) \Rightarrow \beta) \vdash_\iota \forall x(\mathbf{\Delta}(\alpha \Rightarrow \beta)) & \forall\text{-intro}
\end{array}$$

□

Property 42 $\mathbf{\Delta}(\exists! x(\alpha)) \dashv\vdash_\iota \forall x(\mathbf{\Delta}(\alpha))$

Note that $\forall x(\mathbf{\Delta}(\alpha)) \equiv \mathbf{\Delta}(\exists x(\alpha))$.

Proof.

The $\vdash_\iota$ direction is easy:

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\exists! x(\alpha))) & \text{Ddef} \\
\mathbf{\Delta}(\exists! x(\alpha)) \vdash_\iota \mathbf{\Delta}(\exists! x(\alpha)) & \text{ass} \\
\mathbf{\Delta}(\exists! x(\alpha)) \vdash_\iota \mathbf{\Delta}(\exists x(\alpha)) & \&\text{-elim}
\end{array}$$

For the other direction, we have (note that y does not occur in α):

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\forall x(\alpha))) & \text{Ddef} \\
\forall x(\mathbf{\Delta}(\alpha)) \vdash_\iota \forall x(\mathbf{\Delta}(\alpha)) & \text{ass} \\
\forall x(\mathbf{\Delta}(\alpha)) \vdash_\iota \mathbf{\Delta}(\alpha) & \forall\text{-elim} \\
\forall x(\mathbf{\Delta}(\alpha)) \vdash_\iota \mathbf{\Delta}([y/x]\alpha) & \text{subst} \\
\forall x(\mathbf{\Delta}(\alpha)) \vdash_\iota \mathbf{\Delta}(\alpha) \,\&\, \mathbf{\Delta}([y/x]\alpha) & \&\text{-intro} \\
\forall x(\mathbf{\Delta}(\alpha)) \vdash_\iota \forall y(\mathbf{\Delta}(\alpha) \,\&\, \mathbf{\Delta}([y/x]\alpha)) & \forall\text{-intro} \\
\forall x(\mathbf{\Delta}(\alpha)) \vdash_\iota \forall x\forall y(\mathbf{\Delta}(\alpha) \,\&\, \mathbf{\Delta}([y/x]\alpha)) & \forall\text{-intro} \\
\forall x(\mathbf{\Delta}(\alpha)); \vdash_\iota \forall x\forall y(\mathbf{\Delta}(\alpha) \,\&\, \mathbf{\Delta}([y/x]\alpha)) & \text{toCtxt} \\
\forall x(\mathbf{\Delta}(\alpha)); \vdash_\iota \forall x(\mathbf{\Delta}(\alpha)) & \text{toCtxt} \\
\forall x(\mathbf{\Delta}(\alpha)); \exists x(\alpha) \vdash_\iota \forall x\forall y(\mathbf{\Delta}(\alpha) \,\&\, \mathbf{\Delta}([y/x]\alpha)) & \text{Weak} \\
\forall x(\mathbf{\Delta}(\alpha)); \vdash_\iota \exists x(\alpha) \Rightarrow \forall x\forall y(\mathbf{\Delta}(\alpha) \,\&\, \mathbf{\Delta}([y/x]\alpha)) & \text{DdRu2} \\
\forall x(\mathbf{\Delta}(\alpha)); \vdash_\iota \mathbf{\Delta}(\exists! x(\alpha)) & \&\text{-intro}
\end{array}$$

□

Next, we prove a derived rule, where Σ is a possibly empty list of formulae. In the derivation, z is to be chosen different from x and not free in Σ, ψ_1, ψ_2 and φ_1; w is to be chosen different from z and not free in $\Sigma, \psi_1, \psi_2, \varphi_1$ and φ_2.

For readability, we abbreviate the formula $\forall x\forall y((\varphi_1 \,\&\, [y/x]\varphi_1) \Rightarrow x = y)$ as Ξ.

$$\begin{array}{rr}
\boxed{\text{Eq-}\iota} & \\
\psi_1 \vdash_\iota \exists! x(\varphi_1) & \text{prem} \\
\psi_2 \vdash_\iota \exists! x(\varphi_2) & \text{prem} \\
\Sigma; \psi_1, \psi_2 \vdash_\iota \forall x(\varphi_1 \Leftrightarrow \varphi_2) & \text{prem} \\
\Sigma; \psi_1 \vdash_\iota \forall x\forall y((\varphi_1 \,\&\, [y/x]\varphi_1) \Rightarrow x = y) & \&\text{-elim} \\
\mathbf{\Delta}(\Xi)\,; \Xi \vdash_\iota \forall x\forall y((\varphi_1 \,\&\, [y/x]\varphi_1) \Rightarrow x = y) & \text{AssCtxt} \\
\mathbf{\Delta}(\Xi)\,; \Xi \vdash_\iota \forall y((\varphi_1 \,\&\, [y/x]\varphi_1) \Rightarrow x = y) & \forall\text{-elim} \\
\mathbf{\Delta}(\Xi)\,; \Xi \vdash_\iota (\varphi_1 \,\&\, [y/x]\varphi_1) \Rightarrow x = y & \forall\text{-elim} \\
\mathbf{\Delta}(\Xi)\,; \Xi \vdash_\iota (\varphi_1 \,\&\, [z/x]\varphi_1) \Rightarrow x = z & \text{subst}
\end{array}$$

$$\begin{array}{rl}
\mathbf{\Delta}(\Xi)\,;\Xi \vdash_\iota ([w/x]\varphi_1 \,\&\, [z/x]\varphi_1) \Rightarrow w = z & \text{subst}\\
\mathbf{\Delta}(\Xi)\,,\Xi;[w/x]\varphi_1 \,\&\, [z/x]\varphi_1 \vdash_\iota w = z & \text{DdRu1}\\
\mathbf{\Delta}(\Xi)\,,\Xi,[w/x]\varphi_1;[z/x]\varphi_1 \vdash_\iota w = z & \text{toCtxt}\\
\mathbf{\Delta}(\forall x(\varphi_1 \Leftrightarrow \varphi_2))\,;\forall x(\varphi_1 \Leftrightarrow \varphi_2) \vdash_\iota \forall x(\varphi_1 \Leftrightarrow \varphi_2) & \text{AssCtxt}\\
\mathbf{\Delta}(\forall x(\varphi_1 \Leftrightarrow \varphi_2))\,;\forall x(\varphi_1 \Leftrightarrow \varphi_2) \vdash_\iota \varphi_1 \Leftrightarrow \varphi_2 & \forall\text{-elim}\\
\mathbf{\Delta}(\forall x(\varphi_1 \Leftrightarrow \varphi_2))\,;\forall x(\varphi_1 \Leftrightarrow \varphi_2) \vdash_\iota \varphi_2 \Rightarrow \varphi_1 & \&\text{-elim}\\
\mathbf{\Delta}(\forall x(\varphi_1 \Leftrightarrow \varphi_2))\,;\forall x(\varphi_1 \Leftrightarrow \varphi_2) \vdash_\iota [z/x]\varphi_2 \Rightarrow [z/x]\varphi_1 & \text{subst}\\
\Sigma;\psi_1,\psi_2 \vdash_\iota [z/x]\varphi_2 \Rightarrow [z/x]\varphi_1 & \text{Cut3}\\
\Sigma,\psi_1,\psi_2;[z/x]\varphi_2 \vdash_\iota [z/x]\varphi_1 & \text{DdRu1}\\
\Sigma,\psi_1,\psi_2,\mathbf{\Delta}(\Xi)\,,\Xi,[w/x]\varphi_1;[z/x]\varphi_2 \vdash_\iota w = z & \text{Cut}\\
\varphi_1 \rightleftharpoons \widetilde{\varphi_1} & \text{Th. 25.3}\\
[w/x]\varphi_1 \rightleftharpoons [w/x]\widetilde{\varphi_1} & \text{subst}\\
\Sigma,\psi_1,\psi_2,\mathbf{\Delta}(\dots)\,, & \\
\forall x\forall y(\dots),[w/x]\widetilde{\varphi_1};[z/x]\varphi_2 \vdash_\iota w = z & \text{Th. 26}\\
\varphi_2 \rightleftharpoons \widetilde{\varphi_2} & \text{Th. 25.3}\\
[z/x]\varphi_2 \rightleftharpoons [z/x]\widetilde{\varphi_2} & \text{subst}\\
\Sigma,\psi_1,\psi_2,\mathbf{\Delta}(\Xi)\,, & \\
\Xi,[w/x]\widetilde{\varphi_1};[z/x]\widetilde{\varphi_2} \vdash_\iota w = z & \text{Th. 26}\\
\Sigma,\psi_1,\psi_2,\mathbf{\Delta}(\Xi)\,,\Xi, & \\
[\iota x_{\psi_1}(\varphi_1)/x]\widetilde{\varphi_1};[z/x]\widetilde{\varphi_2} \vdash_\iota \iota x_{\psi_1}(\varphi_1) = z & \text{subst}\\
\Sigma,\psi_1,\psi_2,\mathbf{\Delta}(\dots)\,,\Xi, & \\
[\iota x_{\psi_1}(\varphi_1)/x]\widetilde{\varphi_1};[\iota x_{\psi_2}(\varphi_2)/x]\widetilde{\varphi_2} \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{subst}\\
\Sigma,\psi_1,\psi_2,\mathbf{\Delta}(\Xi)\,,\Xi; & \\
[\iota x_{\psi_1}(\varphi_1)/x]\widetilde{\varphi_1} \,\&\, [\iota x_{\psi_2}(\varphi_2)/x]\widetilde{\varphi_2} \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{fromCtxt}\\
\psi_1 \vdash_\iota [\iota x_{\psi_1}(\varphi_1)/x]\widetilde{\varphi_1} & \text{iota}\\
\psi_1; \vdash_\iota [\iota x_{\psi_1}(\varphi_1)/x]\widetilde{\varphi_1} & \text{toCtxt}\\
\psi_2 \vdash_\iota [\iota x_{\psi_2}(\varphi_2)/x]\widetilde{\varphi_2} & \text{iota}\\
\psi_2; \vdash_\iota [\iota x_{\psi_2}(\varphi_2)/x]\widetilde{\varphi_2} & \text{toCtxt}\\
\psi_1,\psi_2; \vdash_\iota [\iota x_{\psi_1}(\varphi_1)/x]\widetilde{\varphi_1} \,\&\, [\iota x_{\psi_2}(\varphi_2)/x]\widetilde{\varphi_2} & \&\text{-intro}\\
\psi_1,\psi_2,\Sigma,\mathbf{\Delta}(\dots)\,,\forall x\forall y(\dots); \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{Cut}\\
\psi_1 \vdash_\iota \forall x\forall y((\varphi_1 \,\&\, [y/x]\varphi_1) \Rightarrow x = y) & \&\text{-elim}\\
\psi_1; \vdash_\iota \forall x\forall y((\varphi_1 \,\&\, [y/x]\varphi_1) \Rightarrow x = y) & \text{toCtxt}\\
\psi_1,\psi_2,\Sigma,\mathbf{\Delta}(\Xi)\,;\Xi \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{fromCtxt}\\
\psi_1,\psi_2,\Sigma; \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{Cut3}\\
\Sigma,\psi_1,\psi_2; \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{WeakCtxtL}\\
\Sigma,\psi_1;\psi_2 \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{fromCtxt}\\
\Sigma;\psi_1 \,\&\, \psi_2 \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{fromCtxt}\\
\vdash_\iota \mathbf{\Delta}(\psi_2) & \text{defAnt}\\
\Sigma;\psi_1,\psi_2 \vdash_\iota \iota x_{\psi_1}(\varphi_1) = \iota x_{\psi_2}(\varphi_2) & \text{AnDc}
\end{array}$$

In case $\varphi_1 \equiv \varphi_2$ and Σ is empty, one can easily derive the third premise

from the first two, and we get the following rule:

$$\frac{\begin{array}{c}\boxed{\text{Eq-}\iota}\\ \psi_1 \vdash_\iota \exists!x(\varphi)\\ \psi_2 \vdash_\iota \exists!x(\varphi)\end{array}}{\psi_1, \psi_2 \vdash_\iota \iota x_{\psi_1}(\varphi) = \iota x_{\psi_2}(\varphi)}$$

To see where this rule might be useful, consider again the theory of real numbers. One expects to have $x \geq 0 \vdash_\iota \exists!y(y \geq 0 \,\&\, x = y^2)$, expressing that $\sqrt{x}$ exists for non-negative x, and likewise $x \leq 0 \vdash_\iota \exists!y(y \geq 0 \,\&\, x = -y^2)$, expressing that $\sqrt{-x}$ exists for negative x. Using the rule, we then get $x \geq 0, x \leq 0 \vdash_\iota \iota y_{x\geq 0}(y \geq 0 \,\&\, x = y^2) = \iota y_{x\leq 0}(y \geq 0 \,\&\, x = -y^2)$, i.e. if $x = 0$ then $\sqrt{x} = \sqrt{-x}$. From this example, we can see that in general, both ψ_1 and ψ_2 are required in the antecedent of the conclusion of the rule (for any $x \neq 0$, the consequent becomes undefined).

Note that in general, we cannot simplify the third premise to $\psi_1, \psi_2 \vdash_\iota \varphi_1 \Leftrightarrow \varphi_2$. A counterexample in the theory of real numbers is given by the three derivable sequents

$$\begin{gathered} y \neq 0 \vdash_\iota \exists!x(x \cdot y = 1)\\ x = y \,\&\, x \geq 0 \vdash_\iota \exists!x(x \cdot x = 1 \,\&\, x \geq 0)\\ y \neq 0, x = y \,\&\, x \geq 0 \vdash_\iota x \cdot y = 1 \Leftrightarrow (x \cdot x = 1 \,\&\, x \geq 0)\end{gathered}$$

Yet we don't expect

$$y \neq 0, x = y \,\&\, x \geq 0 \vdash_\iota \iota x_{y\neq 0}(x \cdot y = 1) = \iota x_{x=y\&x\geq 0}(x \cdot x = 1 \,\&\, x \geq 0)$$

to be derivable, since the interpretation of the first ι-term is "$\frac{1}{y}$ when $y \neq 0$ and undefined otherwise" and the interpretation of the second one is "1 when $x = y \,\&\, x \geq 0$ and undefined otherwise".
And indeed, the Eq-ι rule does not allow this sequent to be derived, since we cannot derive

$$y \neq 0, x = y \,\&\, x \geq 0 \vdash_\iota \forall x(x \cdot y = 1 \Leftrightarrow (x \cdot x = 1 \,\&\, x \geq 0)).$$

The next rule is an analogue of the RenG rule for ι-terms. It is applicable if y is not a free variable of φ.

In the next derivation, w is to be chosen different from x, y and z and be not a free variable of φ.
For readability, we abbreviate the formula $\forall x \forall y((\varphi \,\&\, [y/x]\,\varphi) \Rightarrow x = y)$ as Ξ.

$\boxed{\text{Ren-}\iota}$

$$\begin{array}{rl}
\psi \vdash_\iota \exists! x(\varphi) & \text{prem} \\
\psi \vdash_\iota \exists x(\varphi) & \&\text{-elim} \\
\forall x(\boldsymbol{\Delta}(\varphi)); \exists x(\varphi) \vdash_\iota \exists y([y/x]\varphi) & \text{RenP} \\
\psi \vdash_\iota \exists y([y/x]\varphi) & \text{CutCtxt} \\
\psi \vdash_\iota \forall x \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y) & \&\text{-elim} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota \forall x \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y) & \text{AssCtxt} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y) & \forall\text{-elim} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota (\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y & \forall\text{-elim} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota ([w/x]\varphi \,\&\, [y/x]\varphi) \Rightarrow w = y & \text{subst} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota ([w/x]\varphi \,\&\, [z/x]\varphi) \Rightarrow w = z & \text{subst} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota ([y/x]\varphi \,\&\, [z/x]\varphi) \Rightarrow y = z & \text{subst} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota \forall z(([y/x]\varphi \,\&\, [z/x]\varphi) \Rightarrow y = z) & \forall\text{-intro} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota \forall x \forall z(([y/x]\varphi \,\&\, [z/x]\varphi) \Rightarrow y = z) & \forall\text{-intro} \\
\psi \vdash_\iota \forall x \forall z(([y/x]\varphi \,\&\, [z/x]\varphi) \Rightarrow y = z) & \text{Cut} \\
\psi \vdash_\iota \exists! y([y/x]\varphi) & \&\text{-intro} \\
\psi \vdash_\iota [\iota y_\psi([y/x]\varphi)/y]\, \widetilde{[y/x]\varphi} & \text{iota} \\
\varphi \rightleftharpoons \widetilde{\varphi} & \text{Th. 25.3} \\
[y/x]\varphi \rightleftharpoons \widetilde{[y/x]\varphi} & \text{Th. 25.3} \\
\boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota (\widetilde{\varphi} \,\&\, \widetilde{[y/x]\varphi}) \Rightarrow x = y & \text{Th. 26} \\
[w/y]\psi \vdash_\iota \exists! x(\varphi) & \text{subst} \\
[w/y]\psi \vdash_\iota \left[\iota x_{[w/y]\psi}(\varphi)/x\right] \widetilde{\varphi} & \text{iota} \\
[w/y]\psi, \boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota (\left[\iota x_{[w/y]\psi}(\varphi)/x\right] \widetilde{\varphi} \,\&\, \widetilde{[y/x]\varphi}) & \\
\qquad \Rightarrow \iota x_{[w/y]\psi}(\varphi) = y & \text{subst} \\
[w/y]\psi \vdash_\iota \exists! y([y/x]\varphi) & \&\text{-intro} \\
[w/y]\psi \vdash_\iota \left[\iota y_{[w/y]\psi}([y/x]\varphi)/y\right] \widetilde{[y/x]\varphi} & \text{iota} \\
[w/y]\psi \vdash_\iota \left[\iota x_{[w/y]\psi}(\varphi)/x\right] \widetilde{\varphi} \,\&\, \left[\iota y_{[w/y]\psi}([y/x]\varphi)/y\right] \widetilde{[y/x]\varphi} & \&\text{-intro} \\
[w/y]\psi; \vdash_\iota \left[\iota x_{[w/y]\psi}(\varphi)/x\right] \widetilde{\varphi} \,\&\, \left[\iota y_{[w/y]\psi}([y/x]\varphi)/y\right] \widetilde{[y/x]\varphi} & \text{toCtxt} \\
[w/y]\psi, \boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota (\left[\iota x_{[w/y]\psi}(\varphi)/x\right] \widetilde{\varphi} \,\&\, \left[\iota y_{[w/y]\psi}([y/x]\varphi)/y\right] \widetilde{[y/x]\varphi}) & \\
\qquad \Rightarrow \iota x_{[w/y]\psi}(\varphi) = \iota y_{[w/y]\psi}([y/x]\varphi) & \text{subst} \\
[w/y]\psi, \boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota \iota x_{[w/y]\psi}(\varphi) = \iota y_{[w/y]\psi}([y/x]\varphi) & \text{MP} \\
\psi, \boldsymbol{\Delta}(\Xi)\,; \Xi \vdash_\iota \iota x_\psi(\varphi) = \iota y_\psi([y/x]\varphi) & \text{subst} \\
\psi; \vdash_\iota \forall x \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y) & \text{toCtxt} \\
\psi, \boldsymbol{\Delta}(\Xi)\,; \vdash_\iota \iota x_\psi(\varphi) = \iota y_\psi([y/x]\varphi) & \text{Cut} \\
\psi; \boldsymbol{\Delta}(\Xi)\,; \vdash_\iota \iota x_\psi(\varphi) = \iota y_\psi([y/x]\varphi) & \text{fromCtxt} \\
\psi; \vdash_\iota \boldsymbol{\Delta}(\forall x \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y)) & \text{defCons} \\
\psi; \vdash_\iota \iota x_\psi(\varphi) = \iota y_\psi([y/x]\varphi) & \text{Cut} \\
\psi \vdash_\iota \iota x_\psi(\varphi) = \iota y_\psi([y/x]\varphi) & \text{fromCtxt}
\end{array}$$

Corollary 43 *Interchangeability of $\iota x_\psi(\varphi)$ and $\iota y_\psi([y/x]\varphi)$ when y is not a free variable of φ and $\psi \vdash_\iota \exists! x(\varphi)$ is given.*

Proof.

The sequent we have to supply is immediately obtained by the Ren-ι rule. □

Theorem 44 *Given terms t_1, t_2 and a formula α or term τ, for which all the uniqueness conditions can be derived.*

If t_1 and t_2 are interchangeable, then so are $[t_1/x]\alpha$ and $[t_2/x]\alpha$ if these substitutions are defined and analogously, so are $[t_1/x]\tau$ and $[t_2/x]\tau$.

Proof.

We prove this by structural induction on α and τ.

- $\tau \equiv x$ We have to show that t_1 and t_2 are interchangeable, which is the statement of the lemma.

- $\tau \equiv y$ where y is a variable symbol different from x Both terms are y.

- $\tau \equiv f(\tau_1, \tau_2, \dots)$ Easy using induction on the τ_i and the ERf2 rule.

- $\tau \equiv \iota y_\psi(\varphi)$ where $x \equiv y$ or x is not a free variable of φ If x is also not a free variable of ψ, then both terms are simply $\iota y_\psi(\varphi)$, so let us suppose x is free in ψ. Then we have to show

$$\begin{cases} \mathbf{\Delta}(t_1) \,\&\, [t_1/x]\psi \vdash_\iota \iota y_{\mathbf{\Delta}(t_1)\&[t_1/x]\psi}(\varphi) = \iota y_{\mathbf{\Delta}(t_2)\&[t_2/x]\psi}(\varphi) \\ \mathbf{\Delta}(t_2) \,\&\, [t_2/x]\psi \vdash_\iota \iota y_{\mathbf{\Delta}(t_1)\&[t_1/x]\psi}(\varphi) = \iota y_{\mathbf{\Delta}(t_2)\&[t_2/x]\psi}(\varphi) \end{cases}$$

  which is not difficult:

$$\begin{array}{rr}
\psi \vdash_\iota \exists!y(\varphi) & \mathrm{UC}(\tau) \\
\boldsymbol{\Delta}(t_1)\,;[t_1/x]\psi \vdash_\iota \exists!y(\varphi) & \text{subst} \\
\boldsymbol{\Delta}(t_1)\,\&\,[t_1/x]\psi \vdash_\iota \exists!y(\varphi) & \text{fromCtxt} \\
\boldsymbol{\Delta}(t_2)\,\&\,[t_2/x]\psi \vdash_\iota \exists!y(\varphi) & \text{analogous} \\
\boldsymbol{\Delta}(t_1)\,\&\,[t_1/x]\psi, \boldsymbol{\Delta}(t_2)\,\&\,[t_2/x]\psi \vdash_\iota \iota y_{\boldsymbol{\Delta}(t_1)\&[t_1/x]\psi}(\varphi) = \iota y_{\boldsymbol{\Delta}(t_2)\&[t_2/x]\psi}(\varphi) & \text{Eq-}\iota \\
\psi \vdash_\iota \psi & \text{ass} \\
\boldsymbol{\Delta}(t_1)\,;[t_1/x]\psi \vdash_\iota [t_1/x]\psi & \text{subst} \\
\boldsymbol{\Delta}([t_1/x]\psi)\,;[t_1/x]\psi \vdash_\iota [t_2/x]\psi & \text{induction} \\
\boldsymbol{\Delta}(t_1)\,;[t_1/x]\psi \vdash_\iota [t_2/x]\psi & \text{Cut3} \\
\boldsymbol{\Delta}(t_1) \vdash_\iota t_1 = t_2 & \text{statement} \\
\boldsymbol{\Delta}(t_1) \vdash_\iota \boldsymbol{\Delta}(t_1)\,\&\,\boldsymbol{\Delta}(t_2) & \text{defCons} \\
\boldsymbol{\Delta}(t_1) \vdash_\iota \boldsymbol{\Delta}(t_2) & \&\text{-elim} \\
\boldsymbol{\Delta}(t_1)\,;\vdash_\iota \boldsymbol{\Delta}(t_2) & \text{toCtxt} \\
\boldsymbol{\Delta}(t_1)\,;[t_1/x]\psi \vdash_\iota \boldsymbol{\Delta}(t_2)\,\&\,[t_2/x]\psi & \&\text{-intro} \\
\boldsymbol{\Delta}(t_1)\,\&\,[t_1/x]\psi \vdash_\iota \boldsymbol{\Delta}(t_2)\,\&\,[t_2/x]\psi & \text{fromCtxt} \\
\boldsymbol{\Delta}(t_1)\,\&\,[t_1/x]\psi \vdash_\iota \iota y_{\boldsymbol{\Delta}(t_1)\&[t_1/x]\psi}(\varphi) = \iota y_{\boldsymbol{\Delta}(t_2)\&[t_2/x]\psi}(\varphi) & \text{Cut}
\end{array}$$

The other sequent is derived analogously.

- $\tau \equiv \iota y_\psi(\varphi)$ where $x \not\equiv y$ and x is a free variable of φ For the substitutions to be defined, y must not be a free variable of t_1 and t_2. We have to show that $\iota y_{\boldsymbol{\Delta}(t_1)\&[t_1/x]\psi}([t_1/x]\varphi)$ and $\iota y_{\boldsymbol{\Delta}(t_2)\&[t_2/x]\psi}([t_2/x]\varphi)$ are interchangeable:

$$\begin{array}{rr}
\psi \vdash_\iota \exists!y(\varphi) & \mathrm{UC}(\tau) \\
\boldsymbol{\Delta}(t_1)\,;[t_1/x]\psi \vdash_\iota \exists!y([t_1/x]\varphi) & \text{subst} \\
\boldsymbol{\Delta}(t_1)\,\&\,[t_1/x]\psi \vdash_\iota \exists!y([t_1/x]\varphi) & \text{fromCtxt} \\
\boldsymbol{\Delta}(t_2)\,\&\,[t_2/x]\psi \vdash_\iota \exists!y([t_2/x]\varphi) & \text{analogous} \\
\boldsymbol{\Delta}([t_1/x]\varphi)\,;[t_1/x]\varphi \vdash_\iota [t_2/x]\varphi & \text{induction} \\
\boldsymbol{\Delta}([t_1/x]\varphi)\,;\vdash_\iota [t_1/x]\varphi \Rightarrow [t_2/x]\varphi & \text{DdRu2} \\
\boldsymbol{\Delta}([t_1/x]\varphi) \vdash_\iota [t_1/x]\varphi \Rightarrow [t_2/x]\varphi & \text{fromCtxt} \\
\boldsymbol{\Delta}([t_2/x]\varphi) \vdash_\iota [t_2/x]\varphi \Rightarrow [t_1/x]\varphi & \text{analogous} \\
\boldsymbol{\Delta}([t_1/x]\varphi)\,,\boldsymbol{\Delta}([t_2/x]\varphi) \vdash_\iota [t_1/x]\varphi \Leftrightarrow [t_2/x]\varphi & \&\text{-intro} \\
\boldsymbol{\Delta}(t_1)\,;[t_1/x]\psi \vdash_\iota \exists y([t_1/x]\varphi) & \&\text{-elim} \\
\boldsymbol{\Delta}(t_1)\,;[t_1/x]\psi \vdash_\iota \forall y(\boldsymbol{\Delta}([t_1/x]\varphi)) & \text{defCons} \\
\boldsymbol{\Delta}(t_1)\,;[z/y][t_1/x]\psi \vdash_\iota \forall y(\boldsymbol{\Delta}([t_1/x]\varphi)) & \text{subst} \\
\boldsymbol{\Delta}(t_1)\,;[z/y][t_1/x]\psi \vdash_\iota \boldsymbol{\Delta}([t_1/x]\varphi) & \forall\text{-elim} \\
\boldsymbol{\Delta}(t_1)\,\&\,[z/y][t_1/x]\psi \vdash_\iota \boldsymbol{\Delta}([t_1/x]\varphi) & \text{toCtxt} \\
\boldsymbol{\Delta}(t_2)\,\&\,[z/y][t_2/x]\psi \vdash_\iota \boldsymbol{\Delta}([t_2/x]\varphi) & \text{analogous} \\
\boldsymbol{\Delta}(t_1)\,\&\,[z/y][t_1/x]\psi, \boldsymbol{\Delta}(t_2)\,\&\,[z/y][t_2/x]\psi \vdash_\iota [t_1/x]\varphi \Leftrightarrow [t_2/x]\varphi & \text{Cut (twice)} \\
\boldsymbol{\Delta}(t_1)\,\&\,[z/y][t_1/x]\psi, \boldsymbol{\Delta}(t_2)\,\&\,[z/y][t_2/x]\psi \vdash_\iota \forall y([t_1/x]\varphi \Leftrightarrow [t_2/x]\varphi) & \forall\text{-intro} \\
\boldsymbol{\Delta}(t_1)\,\&\,[z/y][t_1/x]\psi, \boldsymbol{\Delta}(t_2)\,\&\,[z/y][t_2/x]\psi \vdash_\iota \iota y_{\boldsymbol{\Delta}(t_1)\&[z/y][t_1/x]\psi}([t_1/x]\varphi) &
\end{array}$$

$$\begin{array}{rll} & = \iota y_{\mathbf{\Delta}(t_2)\&[z/y][t_2/x]\psi}([t_2/x]\,\varphi) & \text{Eq-}\iota \\ \mathbf{\Delta}(t_1)\,\&\,[t_1/x]\,\psi, \mathbf{\Delta}(t_2)\,\&\,[t_2/x]\,\psi \vdash_\iota & \iota y_{\mathbf{\Delta}(t_1)\&[t_1/x]\psi}([t_1/x]\,\varphi) & \\ & = \iota y_{\mathbf{\Delta}(t_2)\&[t_2/x]\psi}([t_2/x]\,\varphi) & \text{subst} \end{array}$$

where we choose z different from y and not occurring in t_1, t_2 and φ.

As in the previous case, we can derive $\mathbf{\Delta}(t_1)\&[t_1/x]\,\psi \dashv\vdash_\iota \mathbf{\Delta}(t_2)\&[t_2/x]\,\psi$ and the required sequents are easily obtained.

- The other cases are straightforward.

□

3.9 Completeness

In this section we will prove the completeness of the PITFOL calculus, i.e.,

$$\text{if } \Gamma \models_\iota \alpha \text{ then also } \Gamma \vdash_\iota \alpha$$

Denote $\Gamma \equiv \gamma_1, \gamma_2, \ldots, \gamma_m$. By definition, $\Gamma \models_\iota \alpha$ means that for any interpretation $\mathcal{I}$,

- whenever all γ_i are valid in $\mathcal{I}$, then so is α. Using lemma 20, this is equivalent with

$$\mathcal{D}(\gamma_1)\,\&\,\mathcal{R}(\gamma_1)\,, \mathcal{D}(\gamma_2)\,\&\,\mathcal{R}(\gamma_2)\,, \ldots, \mathcal{D}(\gamma_m)\,\&\,\mathcal{R}(\gamma_m) \models \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha)$$

- all γ_i are defined in $\mathcal{I}$, which is by the same lemma equivalent with

$$\begin{cases} \models \mathcal{D}(\gamma_1) \\ \models \mathcal{D}(\gamma_2) \\ \models \ldots \\ \models \mathcal{D}(\gamma_m) \end{cases}$$

Using the completeness of the Hermes calculus, what we have to prove is reduced to a syntactical problem:

$$\text{if } \begin{cases} \mathcal{D}(\gamma_1)\,\&\,\mathcal{R}(\gamma_1)\,, \ldots, \mathcal{D}(\gamma_m)\,\&\,\mathcal{R}(\gamma_m) \vdash \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha) & \\ \qquad\qquad\qquad\qquad\qquad\quad \vdash \mathcal{D}(\gamma_1) & \\ \qquad\qquad\qquad\qquad\qquad\quad \vdash \mathcal{D}(\gamma_2) & \text{then also } \Gamma \vdash_\iota \alpha \\ \qquad\qquad\qquad\qquad\qquad\quad \vdash \ldots & \\ \qquad\qquad\qquad\qquad\qquad\quad \vdash \mathcal{D}(\gamma_m) & \end{cases}$$

We will first prove some properties we will need later.

Lemma 45 *If the uniqueness condition for $\iota x_\psi(\varphi)$ is derivable, and y is not a free variable of φ, then $\psi;\vdash_\iota [y/x]\varphi \Leftrightarrow y = \iota x_\psi(\varphi)$.*

Proof.

If y is different from x and z (where z is the variable symbol used in the uniqueness condition), we can proceed as follows:

$$\begin{array}{rl}
\psi \vdash_\iota \exists! x(\varphi) & \text{UC}\\
\psi \vdash_\iota \forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) & \text{\&-elim}\\
\boldsymbol{\Delta}(\dots);\forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) \vdash_\iota \forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) & \text{AssCtxt}\\
\boldsymbol{\Delta}(\dots);\forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) \vdash_\iota \forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) & \forall\text{-elim}\\
\forall z(\boldsymbol{\Delta}(\dots));\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) \vdash_\iota \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y) & \text{RenG}\\
\boldsymbol{\Delta}(\dots);\forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) \vdash_\iota \forall y((\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y) & \text{Cut3}\\
\boldsymbol{\Delta}(\dots);\forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) \vdash_\iota (\varphi \,\&\, [y/x]\varphi) \Rightarrow x = y & \forall\text{-elim}\\
\boldsymbol{\Delta}(\dots);\forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) \vdash_\iota (\widetilde{\varphi} \,\&\, [y/x]\varphi) \Rightarrow x = y & \text{Th. 25.3}\\
\psi,\boldsymbol{\Delta}(\dots);\forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) \vdash_\iota ([\iota x_\psi(\varphi)/x]\widetilde{\varphi} \,\&\, [y/x]\varphi) & \\
\qquad \Rightarrow \iota x_\psi(\varphi) = y & \text{subst}\\
\psi;\vdash_\iota \forall x\forall z((\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z) & \text{toCtxt}\\
\psi;\vdash_\iota ([\iota x_\psi(\varphi)/x]\widetilde{\varphi} \,\&\, [y/x]\varphi) & \\
\qquad \Rightarrow \iota x_\psi(\varphi) = y & \text{Cut3}\\
\psi;[\iota x_\psi(\varphi)/x]\widetilde{\varphi} \,\&\, [y/x]\varphi \vdash_\iota \iota x_\psi(\varphi) = y & \text{DdRu1}\\
\psi,[\iota x_\psi(\varphi)/x]\widetilde{\varphi};[y/x]\varphi \vdash_\iota \iota x_\psi(\varphi) = y & \text{toCtxt}\\
\psi \vdash_\iota [\iota x_\psi(\varphi)/x]\widetilde{\varphi} & \text{iota}\\
\psi;[y/x]\varphi \vdash_\iota \iota x_\psi(\varphi) = y & \text{CutCtxt}\\
\psi;[y/x]\varphi \vdash_\iota y = \iota x_\psi(\varphi) & \text{ESy2}
\end{array}$$

If $y \equiv z$, then we cannot apply RenG in the proof above, but by removing the RenG and Cut3 line of the proof above, we get a correct proof for this case.

Finally, if $y \equiv x$, first choose a variable symbol y' not occurring in ψ, different from x and not a free variable of φ. Then the above reasoning yields a proof of

$$\psi;[y'/x]\varphi \vdash_\iota y' = \iota x_\psi(\varphi)$$

Applying the subst rule yields the desired sequent.

For the other direction, we have

$$\begin{array}{rr}
\psi \vdash_\iota [\iota x_\psi(\varphi)/x]\widetilde{\varphi} & \text{iota} \\
\psi; \vdash_\iota [\iota x_\psi(\varphi)/x]\widetilde{\varphi} & \text{toCtxt} \\
\psi, \boldsymbol{\Delta}(\widetilde{\varphi}) ; x = \iota x_\psi(\varphi) \vdash_\iota \widetilde{\varphi} & \text{EqSubst2} \\
\boldsymbol{\Delta}(\widetilde{\varphi}) ; \widetilde{\varphi} \vdash_\iota \boldsymbol{\Delta}(\varphi) \,\&\, \varphi & \text{Th. 25.3} \\
\boldsymbol{\Delta}(\widetilde{\varphi}) ; \widetilde{\varphi} \vdash_\iota \varphi & \&\text{-elim} \\
\psi, \boldsymbol{\Delta}(\widetilde{\varphi}) ; x = \iota x_\psi(\varphi) \vdash_\iota \varphi & \text{Cut3} \\
\psi \vdash_\iota \exists! x(\varphi) & \text{UC} \\
\psi \vdash_\iota \exists x(\varphi) & \&\text{-elim} \\
\psi \vdash_\iota \forall x(\boldsymbol{\Delta}(\varphi)) & \text{defCons} \\
\psi \vdash_\iota \boldsymbol{\Delta}(\varphi) & \forall\text{-elim} \\
\boldsymbol{\Delta}(\varphi) \vdash_\iota \boldsymbol{\Delta}(\widetilde{\varphi}) & \text{Th. 25.3} \\
\psi \vdash_\iota \boldsymbol{\Delta}(\widetilde{\varphi}) & \text{Cut} \\
\psi; x = \iota x_\psi(\varphi) \vdash_\iota \varphi & \text{CutCtxt} \\
\psi; y = \iota x_\psi(\varphi) \vdash_\iota [y/x]\varphi & \text{subst}
\end{array}$$

□

Property 46 *Interchangeability of $[y/x]\varphi$ and $y = \iota x_\psi(\varphi)$ under the context ψ when the uniqueness condition for $\iota x_\psi(\varphi)$ is derivable, and $y \equiv x$ or y is not a free variable of φ. Moreover, $y = \iota x_\psi(\varphi) \rightharpoonup [y/x]\varphi$.*

Proof.

For the interchangeability, we have to derive these sequents:

$$\begin{cases}
\psi, \boldsymbol{\Delta}([y/x]\varphi) ; [y/x]\varphi \vdash_\iota y = \iota x_\psi(\varphi) \\
\psi; y = \iota x_\psi(\varphi) \vdash_\iota [y/x]\varphi \\
\psi; \boldsymbol{\Delta}([y/x]\varphi) \vdash_\iota \psi \\
\psi; \psi \vdash_\iota \boldsymbol{\Delta}([y/x]\varphi)
\end{cases}$$

We derive these sequents easily using the previous lemma:

$$\begin{array}{rr}
\psi; [y/x]\varphi \vdash_\iota y = \iota x_\psi(\varphi) & \text{Lemma 45} \\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([y/x]\varphi)) & \text{Ddef} \\
\psi; \boldsymbol{\Delta}([y/x]\varphi) , [y/x]\varphi \vdash_\iota y = \iota x_\psi(\varphi) & \text{Weak} \\
\psi, \boldsymbol{\Delta}([y/x]\varphi) ; [y/x]\varphi \vdash_\iota y = \iota x_\psi(\varphi) & \text{toCtxt}
\end{array}$$

$$\begin{array}{rr}
\psi; y = \iota x_\psi(\varphi) \vdash_\iota [y/x]\varphi & \text{Lemma 45} \\
\psi \vdash_\iota \exists! x(\varphi) & \text{UC} \\
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\
\psi; \psi, y = \iota x_\psi(\varphi) \vdash_\iota [y/x]\varphi & \text{Weak} \\
\psi; y = \iota x_\psi(\varphi) \vdash_\iota [y/x]\varphi & \text{toCtxt}
\end{array}$$

$$\begin{array}{rr}
\psi \vdash_\iota \exists!x(\varphi) & \text{UC} \\
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\
\psi \vdash_\iota \psi & \text{ass} \\
\psi; \vdash_\iota \psi & \text{toCtxt} \\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([y/x]\varphi)) & \text{Ddef} \\
\psi; \boldsymbol{\Delta}([y/x]\varphi) \vdash_\iota \psi & \text{Weak}
\end{array}$$

$$\begin{array}{rr}
\psi \vdash_\iota \exists!x(\varphi) & \text{UC} \\
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\
\psi; [y/x]\varphi \vdash_\iota y = \iota x_\psi(\varphi) & \text{Lemma 45} \\
\psi; \vdash_\iota \boldsymbol{\Delta}([y/x]\varphi) & \text{defAnt} \\
\psi; \psi \vdash_\iota \boldsymbol{\Delta}([y/x]\varphi) & \text{Weak}
\end{array}$$

We already derived the sequents for $y = \iota x_\psi(\varphi) \rightharpoonup [y/x]\varphi$:

$$\begin{cases} \psi; y = \iota x_\psi(\varphi) \vdash_\iota [y/x]\varphi \\ \psi; [y/x]\varphi \vdash_\iota y = \iota x_\psi(\varphi) \end{cases}$$

□

Lemma 47 *For any formula α of the* PITFOL *calculus, given*

$$\begin{cases} \boldsymbol{\Delta}(\alpha) \,\&\, \alpha \dashv\vdash_\iota \mathcal{D}(\alpha) \,\&\, \mathcal{R}(\alpha) \\ \boldsymbol{\Delta}(\alpha) \dashv\vdash_\iota \mathcal{D}(\alpha) \end{cases}$$

we can derive

$$\alpha \rightharpoonup \mathcal{R}(\alpha)\,, \textit{ i.e., } \begin{cases} \boldsymbol{\Delta}(\alpha)\,; \alpha \vdash_\iota \mathcal{R}(\alpha) \\ \boldsymbol{\Delta}(\alpha)\,; \mathcal{R}(\alpha) \vdash_\iota \alpha \end{cases}$$

Proof.

$$\begin{array}{rr}
\boldsymbol{\Delta}(\alpha) \,\&\, \alpha \vdash_\iota \mathcal{D}(\alpha) \,\&\, \mathcal{R}(\alpha) & \text{given} \\
\boldsymbol{\Delta}(\alpha) \,\&\, \alpha \vdash_\iota \mathcal{R}(\alpha) & \&\text{-elim} \\
\boldsymbol{\Delta}(\alpha)\,; \alpha \vdash_\iota \mathcal{R}(\alpha) & \text{toCtxt}
\end{array}$$

$$\begin{array}{rr}
\mathcal{D}(\alpha) \,\&\, \mathcal{R}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha) \,\&\, \alpha & \text{given} \\
\mathcal{D}(\alpha)\,; \mathcal{R}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha) \,\&\, \alpha & \text{toCtxt} \\
\boldsymbol{\Delta}(\alpha) \vdash_\iota \mathcal{D}(\alpha) & \text{given} \\
\boldsymbol{\Delta}(\alpha)\,; \mathcal{R}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha) \,\&\, \alpha & \text{CutCtxt} \\
\boldsymbol{\Delta}(\alpha)\,; \mathcal{R}(\alpha) \vdash_\iota \alpha & \&\text{-elim}
\end{array}$$

□

Theorem 48 *For any formula α of the* PITFOL *calculus, if the uniqueness conditions for α are derivable, then*

$$\begin{cases} \boldsymbol{\Delta}(\alpha)\ \&\ \alpha \dashv\vdash_\iota \mathcal{D}(\alpha)\ \&\ \mathcal{R}(\alpha) \\ \boldsymbol{\Delta}(\alpha) \dashv\vdash_\iota \mathcal{D}(\alpha) \end{cases}$$

Proof.

We prove this by induction on the complexity of α.

If α has complexity 0, it cannot contain any ι-terms, hence $\boldsymbol{\Delta}(\alpha)$ must be $\top$ and $\mathcal{D}(\alpha)$ is a validity. What we have to prove is reduced to

$$\begin{cases} \alpha \dashv\vdash_\iota \mathcal{D}(\alpha)\ \&\ \mathcal{R}(\alpha) \\ \vdash_\iota \mathcal{D}(\alpha) \end{cases}$$

The last sequent is derivable since we remarked that $\mathcal{D}(\alpha)$ is a validity. We then easily can derive the remaining two sequents:

$$\begin{array}{rr} \vdash_\iota \mathcal{D}(\alpha) & \\ \alpha \vdash_\iota \alpha & \text{ass} \\ \alpha \vdash_\iota \mathcal{D}(\alpha)\ \&\ \alpha & \&\text{-intro} \end{array} \qquad \begin{array}{rr} \vdash_\iota \mathcal{D}(\alpha) & \\ \alpha \vdash_\iota \alpha & \text{ass} \\ \alpha, \mathcal{D}(\alpha) \vdash_\iota \alpha & \text{Weak} \\ \mathcal{D}(\alpha)\ \&\ \alpha \vdash_\iota \alpha & \text{AnU} \end{array}$$

If α has complexity greater than zero, then we have the following cases.

- $\alpha \equiv p(t_1, t_2, \ldots, t_n)$ where we treat $\alpha \equiv t_1 = t_2$ analogously
  If, as usual, we denote the top-level ι-terms of α as $TLI(\alpha) \equiv \iota x_{1\psi_1}(\varphi_1), \iota x_{2\psi_2}(\varphi_2), \ldots, \iota x_{m\psi_m}(\varphi_m)$ and we denote as q the formula obtained from α by replacing all the top-level ι-terms $\iota x_{i\psi_i}(\varphi_i)$ by the corresponding variable symbol u_i, then we have to derive

$$\begin{cases} \psi_1\ \&\ \cdots\ \&\ \psi_m\ \&\ \alpha \dashv\vdash_\iota \mathcal{R}(\psi_1)\ \&\ \cdots\ \&\ \mathcal{R}(\psi_m) \\ \qquad\qquad\qquad\qquad \&\ \exists u_1 \ldots \exists u_m(\mathcal{R}([u_1/x_1]\varphi_1)\ \&\ \cdots\ \&\ \mathcal{R}([u_m/x_m]\varphi_m)\ \&\ q) \\ \psi_1\ \&\ \cdots\ \&\ \psi_m \dashv\vdash_\iota \mathcal{R}(\psi_1)\ \&\ \mathcal{R}(\psi_2)\ \&\ \cdots\ \&\ \mathcal{R}(\psi_m) \end{cases}$$

  where the u_i do not occur in α.

  - We will first handle the second equivalence:

$$\begin{array}{rr}
\psi_1 \vdash_\iota \exists!x_1(\varphi_1) & \text{UC} \\
\vdash_\iota \boldsymbol{\Delta}(\psi_1) & \text{defAnt} \\
\psi_1 \vdash_\iota \psi_1 & \text{ass} \\
\psi_1 \vdash_\iota \boldsymbol{\Delta}(\psi_1) & \text{defCons} \\
\psi_1 \vdash_\iota \boldsymbol{\Delta}(\psi_1) \,\&\, \psi_1 & \&\text{-intro} \\
\boldsymbol{\Delta}(\psi_1) \,\&\, \psi_1 \vdash_\iota \mathcal{D}(\psi_1) \,\&\, \mathcal{R}(\psi_1) & \text{induction} \\
\psi_1 \vdash_\iota \mathcal{D}(\psi_1) \,\&\, \mathcal{R}(\psi_1) & \text{Cut} \\
\psi_1 \vdash_\iota \mathcal{R}(\psi_1) & \&\text{-elim} \\
\vdots & \\
\psi_2 \vdash_\iota \mathcal{R}(\psi_2) & \&\text{-elim} \\
\psi_1, \psi_2 \vdash_\iota \mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) & \&\text{-intro} \\
\psi_1 \,\&\, \psi_2 \vdash_\iota \mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) & \text{AnU} \\
\vdots & \\
\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m \vdash_\iota \mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) \,\&\, \cdots \,\&\, \mathcal{R}(\psi_m) & \text{AnU}
\end{array}$$

and for the other direction

$$\begin{array}{rr}
\mathcal{R}(\psi_1) \vdash_\iota \mathcal{R}(\psi_1) & \text{ass} \\
\psi_1 \vdash_\iota \exists!x_1(\varphi_1) & \text{UC} \\
\vdash_\iota \boldsymbol{\Delta}(\psi_1) & \text{defAnt} \\
\boldsymbol{\Delta}(\psi_1) \vdash_\iota \mathcal{D}(\psi_1) & \text{induction} \\
\vdash_\iota \mathcal{D}(\psi_1) & \text{Cut} \\
\mathcal{R}(\psi_1) \vdash_\iota \mathcal{D}(\psi_1) \,\&\, \mathcal{R}(\psi_1) & \&\text{-intro} \\
\mathcal{D}(\psi_1) \,\&\, \mathcal{R}(\psi_1) \vdash_\iota \boldsymbol{\Delta}(\psi_1) \,\&\, \psi_1 & \text{induction} \\
\mathcal{R}(\psi_1) \vdash_\iota \boldsymbol{\Delta}(\psi_1) \,\&\, \psi_1 & \text{Cut} \\
\mathcal{R}(\psi_1) \vdash_\iota \psi_1 & \text{Cut} \\
\vdots & \\
\mathcal{R}(\psi_2) \vdash_\iota \psi_2 & \text{Cut} \\
\mathcal{R}(\psi_1), \mathcal{R}(\psi_2) \vdash_\iota \psi_1 \,\&\, \psi_2 & \&\text{-intro} \\
\mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) \vdash_\iota \psi_1 \,\&\, \psi_2 & \text{AnU} \\
\vdots & \\
\mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) \,\&\, \cdots \,\&\, \mathcal{R}(\psi_m) \vdash_\iota \psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m & \text{AnU}
\end{array}$$

• For the first equivalence, we proceed as follows.
We will abbreviate

$$\forall u_1 \ldots \forall u_m(\neg(u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q))$$

as Ξ. In the following derivation, we choose z different from $u_1, \ldots, u_m$:

$$
\begin{array}{rll}
\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \alpha \vdash_\iota \alpha & & \text{AssCtxt} \\
\psi_1 \vdash_\iota \iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) & & \text{eq} \\
\psi_2 \vdash_\iota \iota x_{2\psi_2}(\varphi_2) = \iota x_{2\psi_2}(\varphi_2) & & \text{eq} \\
\psi_1, \psi_2 \vdash_\iota \iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) & & \\
\qquad \& \iota x_{2\psi_2}(\varphi_2) = \iota x_{2\psi_2}(\varphi_2) & & \text{\&-intro} \\
\psi_1 \,\&\, \psi_2 \vdash_\iota \iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) & & \\
\qquad \& \iota x_{2\psi_2}(\varphi_2) = \iota x_{2\psi_2}(\varphi_2) & & \text{AnU} \\
\vdots & & \\
\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m \vdash_\iota \iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots & & \\
\qquad \& \iota x_{m\psi_m}(\varphi_m) = \iota x_{m\psi_m}(\varphi_m) & & \text{AnU} \\
\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \vdash_\iota \iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots & & \\
\qquad \& \iota x_{m\psi_m}(\varphi_m) = \iota x_{m\psi_m}(\varphi_m) & & \text{toCtxt} \\
\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \alpha \vdash_\iota \iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots & & \\
\qquad \& \iota x_{m\psi_m}(\varphi_m) = \iota x_{m\psi_m}(\varphi_m) \,\&\, \alpha & & \text{\&-intro} \\
\vdash_\iota z = z & & \text{eq} \\
\psi_1; u_1 = \iota x_{1\psi_1}(\varphi_1) \vdash_\iota z = z & & \text{eqSubst} \\
\psi_1, \psi_2; u_1 = \iota x_{1\psi_1}(\varphi_1), & & \\
\qquad u_2 = \iota x_{2\psi_2}(\varphi_2) \vdash_\iota z = z & & \text{eqSubst} \\
\psi_1, \psi_2; u_1 = \iota x_{1\psi_1}(\varphi_1) & & \\
\qquad \& u_2 = \iota x_{2\psi_2}(\varphi_2) \vdash_\iota z = z & & \text{AnU} \\
\vdots & & \\
\psi_1, \psi_2, \ldots, \psi_m; & & \\
u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots & & \\
\qquad \& u_m = \iota x_{m\psi_m}(\varphi_m) \vdash_\iota z = z & & \text{AnU} \\
\psi_1, \psi_2, \ldots, \psi_m; \vdash_\iota \boldsymbol{\Delta}\big(u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m)\big) & & \text{defAnt} \\
\psi_1, \psi_2, \ldots, \psi_m; \vdash_\iota \forall u_m(\boldsymbol{\Delta}\big(u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m)\big)) & & \forall\text{-intro} \\
\vdots & & \\
\psi_1, \psi_2, \ldots, \psi_m; \vdash_\iota \boldsymbol{\Delta}(\Xi) & & \forall\text{-intro} \\
\psi_1, \psi_2, \ldots, \psi_m; \Xi \vdash_\iota \Xi & & \text{ass} \\
\psi_1, \psi_2, \ldots, \psi_m; \Xi \vdash_\iota \forall u_2 \ldots \forall u_m(\neg(u_1 = \iota x_{1\psi_1}(\varphi_1) & & \\
\qquad \& \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q)) & & \forall\text{-elim} \\
\vdots & & \\
\psi_1, \psi_2, \ldots, \psi_m; \Xi \vdash_\iota \neg(u_1 = \iota x_{1\psi_1}(\varphi_1) & & \\
\qquad \& \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q) & & \forall\text{-elim} \\
\psi_m, \psi_1, \psi_2, \ldots, \psi_{m-1}; \Xi \vdash_\iota \neg\big(u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots & & \\
\qquad \& u_{m-1} = \iota x_{m-1\psi_{m-1}}(\varphi_{m-1}) & & \\
\qquad \& \iota x_{m\psi_m}(\varphi_m) = \iota x_{m\psi_m}(\varphi_m) \,\&\, \big[\iota x_{1\psi_1}(\varphi_1)/u_1\big]\, q\big) & & \text{subst} \\
\psi_{m-1}, \psi_m, & & \\
\psi_1, \psi_2, \ldots, \psi_{m-2}; \Xi \vdash_\iota \neg\big(u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots & & \\
\qquad \& u_{m-2} = \iota x_{m-2\psi_{m-2}}(\varphi_{m-2}) & & \\
\qquad \& u_{m-1} = \iota x_{m-1\psi_{m-1}}(\varphi_{m-1}) & &
\end{array}
$$

$$\& \iota x_{m\psi_m}(\varphi_m) = \iota x_{m\psi_m}(\varphi_m) \,\&\, \left[\iota x_{1\psi_1}(\varphi_1)/u_1\right] q) \quad \text{subst}$$

$$\vdots$$

$$\psi_1, \psi_2, \ldots, \psi_m; \Xi \vdash_\iota \neg(\iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots \\ \& \iota x_{m\psi_m}(\varphi_m) = \iota x_{m\psi_m}(\varphi_m) \,\&\, \alpha) \quad \text{subst}$$

$$\psi_1 \,\&\, \psi_2, \ldots, \psi_m; \Xi \vdash_\iota \neg(\iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots \\ \& \iota x_{m\psi_m}(\varphi_m) = \iota x_{m\psi_m}(\varphi_m) \,\&\, \alpha) \quad \text{CtxtU}$$

$$\vdots$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \Xi \vdash_\iota \neg(\iota x_{1\psi_1}(\varphi_1) = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots \\ \& \iota x_{m\psi_m}(\varphi_m) = \iota x_{m\psi_m}(\varphi_m) \,\&\, \alpha) \quad \text{CtxtU}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \Xi, \alpha \vdash_\iota \neg\Xi \quad \text{contra}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \alpha \vdash_\iota \neg\Xi \quad \text{SeDe}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \alpha \vdash_\iota \neg(\forall u_1 \neg\neg \forall u_2 \ldots \forall u_m(\\ \neg(u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots \\ \& u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q))) \quad \text{Corr. 26, Prop. 37}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \alpha \vdash_\iota \neg(\forall u_1 \neg\neg \forall u_2 \neg\neg \forall u_3 \ldots \forall u_m(\\ \neg(u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \ldots \\ \& u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q))) \quad \text{Corr. 26, Prop. 37}$$

$$\vdots$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \ldots, \psi_m; \alpha \vdash_\iota \exists u_1 \exists u_2 \ldots \exists u_m(u_1 = \iota x_{1\psi_1}(\varphi_1) \\ \& \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q) \quad \text{Corr. 26, Prop. 37}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m \vdash_\iota \psi_1 \quad \&\text{-elim}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m \vdash_\iota \psi_2 \,\&\, \psi_3 \,\&\, \cdots \,\&\, \psi_m \quad \&\text{-elim}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \alpha \vdash_\iota \exists u_1 \exists u_2 \ldots \exists u_m([u_1/x_1]\varphi_1 \,\&\, u_2 = \iota x_{2\psi_2}(\varphi_2) \\ \& \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q) \quad \text{Corr. 24, Prop. 46}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m \vdash_\iota \psi_2 \quad \&\text{-elim}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m \vdash_\iota \psi_3 \,\&\, \psi_4 \,\&\, \cdots \,\&\, \psi_m \quad \&\text{-elim}$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \alpha \vdash_\iota \exists u_1 \exists u_2 \ldots \exists u_m([u_1/x_1]\varphi_1 \,\&\, [u_2/x_2]\varphi_2 \,\&\, u_3 = \iota x_{3\psi_3}(\varphi_3) \\ \& \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q) \quad \text{Corr. 24, Prop. 46}$$

$$\vdots$$

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \alpha \vdash_\iota \exists u_1 \exists u_2 \ldots \exists u_m([u_1/x_1]\varphi_1 \,\&\, \ldots \\ \& [u_m/x_m]\varphi_m \,\&\, q) \quad \text{Corr. 24, Prop. 46}$$

By induction, we can now use lemma 47 on the formulae $[u_i/x_i]\varphi_i$, which we can replace with their reductions using corollary 24 to get

$$\psi_1 \& \psi_2 \& \cdots \& \psi_m; \alpha \vdash_\iota \exists u_1 \ldots \exists u_m(\mathcal{R}([u_1/x_1]\varphi_1) \& \cdots \& \mathcal{R}([u_m/x_m]\varphi_m) \& q)$$

We can combine this with the already derived sequent

$$\psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m \vdash_\iota \mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) \,\&\, \cdots \,\&\, \mathcal{R}(\psi_2)$$

to easily obtain the required sequent.

Finally, we have to derive the other direction, i.e.,

$$\begin{aligned}&\mathcal{R}(\psi_1)\,\&\cdots\&\,\mathcal{R}(\psi_m)\\&\quad\&\,\exists u_1\exists u_2\ldots\exists u_m\big(\mathcal{R}([u_1/x_1]\,\varphi_1)\,\&\cdots\&\,\mathcal{R}([u_m/x_m]\,\varphi_m)\,\&\,q\big)\\&\qquad\qquad\vdash_\iota \psi_1\,\&\cdots\&\,\psi_m\,\&\,\alpha\end{aligned}$$

We already obtained $\mathcal{R}(\psi_1)\&\mathcal{R}(\psi_2)\&\cdots\&\mathcal{R}(\psi_m)\vdash_\iota \psi_1\&\psi_2\&\cdots\&\psi_m$ so it is sufficient to derive

$$\begin{aligned}&\mathcal{R}(\psi_1)\,\&\cdots\&\,\mathcal{R}(\psi_1)\\&\quad\&\,\exists u_1\exists u_2\ldots\exists u_m(\mathcal{R}([u_1/x_1]\,\varphi_1)\,\&\cdots\&\,\mathcal{R}([u_m/x_m]\,\varphi_m)\,\&\,q)\vdash_\iota \alpha\end{aligned}$$

Noting that q does not contain any ι-terms, we derive

$$\dfrac{}{q\vdash_\iota q}\ \text{ass}$$

$$\dfrac{}{\psi_1; q, u_1=\iota x_{1\psi_1}(\varphi_1)\vdash_\iota \left[\iota x_{1\psi_1}(\varphi_1)/u_1\right] q}\ \text{eqSubst}$$

$$\dfrac{}{\begin{array}{c}\psi_1,\psi_2;\\ q, u_1=\iota x_{1\psi_1}(\varphi_1), u_2=\iota x_{2\psi_2}(\varphi_2)\vdash_\iota \left[\iota x_{1\psi_1}(\varphi_1)/u_1\right]\left[\iota x_{2\psi_2}(\varphi_2)/u_2\right] q\end{array}}\ \text{eqSubst}$$

$$\vdots$$

$$\dfrac{}{\begin{array}{c}\psi_1,\psi_2,\ldots,\psi_m;\\ q, u_1=\iota x_{1\psi_1}(\varphi_1),\ldots,u_m=\iota x_{m\psi_m}(\varphi_m)\vdash_\iota \left[\iota x_{1\psi_1}(\varphi_1)/u_1\right]\cdots\left[\iota x_{m\psi_m}(\varphi_m)/u_m\right] q\end{array}}\ \text{eqSubst}$$

$$\dfrac{}{\begin{array}{c}\psi_1,\psi_2,\ldots,\psi_m;\\ q, u_1=\iota x_{1\psi_1}(\varphi_1)\,\&\,u_2=\iota x_{2\psi_2}(\varphi_2),\\ u_3=\iota x_{3\psi_3}(\varphi_3),\ldots,u_m=\iota x_{m\psi_m}(\varphi_m)\vdash_\iota \alpha\end{array}}\ \text{AnU}$$

$$\vdots$$

$$\dfrac{}{\begin{array}{c}\psi_1,\psi_2,\ldots,\psi_m;\\ q, u_1=\iota x_{1\psi_1}(\varphi_1)\,\&\cdots\&\,u_m=\iota x_{m\psi_m}(\varphi_m)\vdash_\iota \alpha\end{array}}\ \text{AnU}$$

$$\dfrac{}{\begin{array}{c}\psi_1,\psi_2,\ldots,\psi_m;\\ u_1=\iota x_{1\psi_1}(\varphi_1)\,\&\cdots\&\,u_m=\iota x_{m\psi_m}(\varphi_m)\,\&\,q\vdash_\iota \alpha\end{array}}\ \text{AnU}$$

$$\dfrac{}{\begin{array}{c}\psi_1,\psi_2,\ldots,\psi_m;\\ \exists u_m(u_1=\iota x_{1\psi_1}(\varphi_1)\,\&\cdots\&\,u_m=\iota x_{m\psi_m}(\varphi_m)\,\&\,q)\vdash_\iota \alpha\end{array}}\ \text{PartAnt}$$

$$\dfrac{}{\begin{array}{c}\psi_1,\psi_2,\ldots,\psi_m;\\ \exists u_{m-1}\exists u_m(u_1=\iota x_{1\psi_1}(\varphi_1)\,\&\cdots\&\,u_m=\iota x_{m\psi_m}(\varphi_m)\,\&\,q)\vdash_\iota \alpha\end{array}}\ \text{PartAnt}$$

$$\vdots$$

$$\dfrac{}{\begin{array}{c}\psi_1,\psi_2,\ldots,\psi_m;\\ \exists u_1\ldots\exists u_m(u_1=\iota x_{1\psi_1}(\varphi_1)\,\&\cdots\&\,u_m=\iota x_{m\psi_m}(\varphi_m)\,\&\,q)\vdash_\iota \alpha\end{array}}\ \text{PartAnt}$$

$$\dfrac{}{\begin{array}{c}\psi_1\,\&\,\psi_2,\psi_3,\ldots,\psi_m;\\ \exists u_1\ldots\exists u_m(u_1=\iota x_{1\psi_1}(\varphi_1)\,\&\cdots\&\,u_m=\iota x_{m\psi_m}(\varphi_m)\,\&\,q)\vdash_\iota \alpha\end{array}}\ \text{CtxtU}$$

$$\vdots$$

$$\begin{array}{rl} & \psi_1 \,\&\, \psi_2 \,\&\, \cdots \,\&\, \psi_m; \\ \exists u_1 \ldots \exists u_m (u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q) \vdash_\iota \alpha & \text{CtxtU} \\ & \mathcal{R}(\psi_1) \,\&\, \mathcal{R}(\psi_2) \,\&\, \cdots \,\&\, \mathcal{R}(\psi_m)\,; \\ \exists u_1 \ldots \exists u_m (u_1 = \iota x_{1\psi_1}(\varphi_1) \,\&\, \cdots \,\&\, u_m = \iota x_{m\psi_m}(\varphi_m) \,\&\, q) \vdash_\iota \alpha & \text{CutCtxt} \end{array}$$

Using property 46, we can replace the $\iota x_{i\psi_i}(\varphi_i)$ with $[u_i/x_i]\varphi_i$. We can now transform the obtained sequent into the required one using again lemma 47 on the formulae $[u_i/x_i]\varphi_i$, to replace them with their reductions using corollary 24.

- $\alpha \equiv \neg\beta$ From the induction hypothesis

$$\begin{cases} \boldsymbol{\Delta}(\beta) \,\&\, \beta \dashv\vdash_\iota \mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) \\ \boldsymbol{\Delta}(\beta) \dashv\vdash_\iota \mathcal{D}(\beta) \end{cases}$$

we have to derive

$$\begin{cases} \boldsymbol{\Delta}(\beta) \,\&\, \neg\beta \dashv\vdash_\iota \mathcal{D}(\beta) \,\&\, \mathcal{R}(\neg\beta) \\ \boldsymbol{\Delta}(\beta) \dashv\vdash_\iota \mathcal{D}(\beta) \end{cases}$$

so we already have the last equivalence. The first one is not difficult:

$$\begin{array}{rr} \boldsymbol{\Delta}(\beta) \,\&\, \beta \vdash_\iota \mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) & \text{induction} \\ \boldsymbol{\Delta}(\beta) \,\&\, \beta \vdash_\iota \mathcal{R}(\beta) & \&\text{-elim} \\ \boldsymbol{\Delta}(\beta)\,;\beta \vdash_\iota \mathcal{R}(\beta) & \text{toCtxt} \\ \boldsymbol{\Delta}(\beta)\,;\neg\mathcal{R}(\beta) \vdash_\iota \neg\beta & \text{CoPo1} \\ \mathcal{D}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{induction} \\ \mathcal{D}(\beta)\,;\neg\mathcal{R}(\beta) \vdash_\iota \neg\beta & \text{CutCtxt} \\ \mathcal{D}(\beta)\,;\vdash_\iota \boldsymbol{\Delta}(\beta) & \text{toCtxt} \\ \mathcal{D}(\beta)\,;\neg\mathcal{R}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) \,\&\, \neg\beta & \&\text{-intro} \\ \mathcal{D}(\beta) \,\&\, \neg\mathcal{R}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) \,\&\, \neg\beta & \text{fromCtxt} \end{array}$$

$$\begin{array}{rr} \mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) \,\&\, \beta & \text{induction} \\ \mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) \vdash_\iota \beta & \&\text{-elim} \\ \mathcal{D}(\beta)\,;\mathcal{R}(\beta) \vdash_\iota \beta & \text{toCtxt} \\ \mathcal{D}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{induction} \\ \mathcal{D}(\beta)\,;\vdash_\iota \boldsymbol{\Delta}(\beta) & \text{toCtxt} \\ \mathcal{D}(\beta)\,;\neg\beta \vdash_\iota \neg\mathcal{R}(\beta) & \text{CoPo1} \\ \boldsymbol{\Delta}(\beta) \vdash_\iota \mathcal{D}(\beta) & \text{induction} \\ \boldsymbol{\Delta}(\beta)\,;\neg\beta \vdash_\iota \neg\mathcal{R}(\beta) & \text{CutCtxt} \\ \boldsymbol{\Delta}(\beta)\,;\vdash_\iota \mathcal{D}(\beta) & \text{toCtxt} \\ \boldsymbol{\Delta}(\beta)\,;\neg\beta \vdash_\iota \mathcal{D}(\beta) \,\&\, \neg\mathcal{R}(\beta) & \&\text{-intro} \\ \boldsymbol{\Delta}(\beta) \,\&\, \neg\beta \vdash_\iota \mathcal{D}(\beta) \,\&\, \neg\mathcal{R}(\beta) & \text{fromCtxt} \end{array}$$

- $\alpha \equiv \beta \,\&\, \gamma$ From the induction hypotheses

$$\begin{cases} \boldsymbol{\Delta}(\beta) \,\&\, \beta \dashv\vdash_\iota \mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) \\ \boldsymbol{\Delta}(\beta) \dashv\vdash_\iota \mathcal{D}(\beta) \end{cases} \quad \begin{cases} \boldsymbol{\Delta}(\gamma) \,\&\, \gamma \dashv\vdash_\iota \mathcal{D}(\gamma) \,\&\, \mathcal{R}(\gamma) \\ \boldsymbol{\Delta}(\gamma) \dashv\vdash_\iota \mathcal{D}(\gamma) \end{cases}$$

we have to derive

$$\begin{cases} \boldsymbol{\Delta}(\beta \,\&\, \gamma) \,\&\, \beta \,\&\, \gamma \dashv\vdash_\iota \mathcal{D}(\beta \,\&\, \gamma) \,\&\, \mathcal{R}(\beta \,\&\, \gamma) \\ \boldsymbol{\Delta}(\beta \,\&\, \gamma) \dashv\vdash_\iota \mathcal{D}(\beta \,\&\, \gamma) \end{cases}$$

We start with the second equivalence:

$$\begin{array}{rr}
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\beta \,\&\, \gamma)) & \text{Ddef} \\
\boldsymbol{\Delta}(\beta) \,\&\, (\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) \vdash_\iota \boldsymbol{\Delta}(\beta) \,\&\, (\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) & \text{ass} \\
\boldsymbol{\Delta}(\beta) \,\&\, (\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) \vdash_\iota \boldsymbol{\Delta}(\beta) & \&\text{-elim} \\
\boldsymbol{\Delta}(\beta) \,\&\, (\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) \vdash_\iota \beta \Rightarrow \boldsymbol{\Delta}(\gamma) & \&\text{-elim} \\
\boldsymbol{\Delta}(\beta) \vdash_\iota \mathcal{D}(\beta) & \text{induction} \\
\boldsymbol{\Delta}(\beta) \,\&\, (\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) \vdash_\iota \mathcal{D}(\beta) & \text{Cut} \\
\boldsymbol{\Delta}(\beta)\,;(\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) \vdash_\iota \beta \Rightarrow \boldsymbol{\Delta}(\gamma) & \text{toCtxt} \\
\boldsymbol{\Delta}(\beta)\,,(\beta \Rightarrow \boldsymbol{\Delta}(\gamma));\beta \vdash_\iota \boldsymbol{\Delta}(\gamma) & \text{DdRu1} \\
\mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) \,\&\, \beta & \text{induction} \\
\mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) \vdash_\iota \beta & \&\text{-elim} \\
\mathcal{D}(\beta)\,;\mathcal{R}(\beta) \vdash_\iota \beta & \text{toCtxt} \\
\boldsymbol{\Delta}(\beta)\,;\mathcal{R}(\beta) \vdash_\iota \beta & \text{CutCtxt} \\
\boldsymbol{\Delta}(\beta)\,,(\beta \Rightarrow \boldsymbol{\Delta}(\gamma));\mathcal{R}(\beta) \vdash_\iota \boldsymbol{\Delta}(\gamma) & \text{Cut} \\
\boldsymbol{\Delta}(\gamma) \vdash_\iota \mathcal{D}(\gamma) & \text{induction} \\
\boldsymbol{\Delta}(\beta)\,,(\beta \Rightarrow \boldsymbol{\Delta}(\gamma));\mathcal{R}(\beta) \vdash_\iota \mathcal{D}(\gamma) & \text{Cut} \\
\boldsymbol{\Delta}(\beta)\,,(\beta \Rightarrow \boldsymbol{\Delta}(\gamma));\vdash_\iota \mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma) & \text{DdRu2} \\
\boldsymbol{\Delta}(\beta)\,;(\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) \vdash_\iota \mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma) & \text{fromCtxt} \\
\boldsymbol{\Delta}(\beta) \,\&\, (\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) \vdash_\iota \mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma) & \text{fromCtxt} \\
\boldsymbol{\Delta}(\beta) \,\&\, (\beta \Rightarrow \boldsymbol{\Delta}(\gamma)) \vdash_\iota \mathcal{D}(\beta) \,\&\, (\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma)) & \&\text{-intro}
\end{array}$$

$$\begin{array}{rr}
\mathcal{D}(\beta) \,\&\, (\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma)) \vdash_\iota \mathcal{D}(\beta) \,\&\, (\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma)) & \text{ass} \\
\mathcal{D}(\beta) \,\&\, (\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma)) \vdash_\iota \mathcal{D}(\beta) & \&\text{-elim} \\
\mathcal{D}(\beta) \,\&\, (\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma)) \vdash_\iota \mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma) & \&\text{-elim} \\
\mathcal{D}(\beta) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{induction} \\
\mathcal{D}(\beta) \,\&\, (\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma)) \vdash_\iota \boldsymbol{\Delta}(\beta) & \text{Cut} \\
\mathcal{D}(\beta)\,;(\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma)) \vdash_\iota \mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma) & \text{toCtxt} \\
\mathcal{D}(\beta)\,,(\mathcal{R}(\beta) \Rightarrow \mathcal{D}(\gamma));\mathcal{R}(\beta) \vdash_\iota \mathcal{D}(\gamma) & \text{DdRu1} \\
\boldsymbol{\Delta}(\beta) \,\&\, \beta \vdash_\iota \mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) & \text{induction} \\
\boldsymbol{\Delta}(\beta) \,\&\, \beta \vdash_\iota \mathcal{R}(\beta) & \&\text{-elim} \\
\boldsymbol{\Delta}(\beta)\,;\beta \vdash_\iota \mathcal{R}(\beta) & \text{toCtxt}
\end{array}$$

$$\begin{array}{rr}
\mathcal{D}(\beta)\,;\beta \vdash_\iota \mathcal{R}(\beta) & \text{CutCtxt}\\
\mathcal{D}(\beta)\,,(\mathcal{R}(\beta)\Rightarrow\mathcal{D}(\gamma));\beta \vdash_\iota \mathcal{D}(\gamma) & \text{Cut}\\
\mathcal{D}(\gamma) \vdash_\iota \mathbf{\Delta}(\gamma) & \text{induction}\\
\mathcal{D}(\beta)\,,(\mathcal{R}(\beta)\Rightarrow\mathcal{D}(\gamma));\beta \vdash_\iota \mathbf{\Delta}(\gamma) & \text{Cut}\\
\mathcal{D}(\beta)\,,(\mathcal{R}(\beta)\Rightarrow\mathcal{D}(\gamma));\vdash_\iota \beta\Rightarrow\mathbf{\Delta}(\gamma) & \text{DdRu2}\\
\mathcal{D}(\beta)\,;(\mathcal{R}(\beta)\Rightarrow\mathcal{D}(\gamma))\vdash_\iota \beta\Rightarrow\mathbf{\Delta}(\gamma) & \text{fromCtxt}\\
\mathcal{D}(\beta)\,\&\,(\mathcal{R}(\beta)\Rightarrow\mathcal{D}(\gamma))\vdash_\iota \beta\Rightarrow\mathbf{\Delta}(\gamma) & \text{fromCtxt}\\
\mathcal{D}(\beta)\,\&\,(\mathcal{R}(\beta)\Rightarrow\mathcal{D}(\gamma))\vdash_\iota \mathbf{\Delta}(\beta)\,\&\,(\beta\Rightarrow\mathbf{\Delta}(\gamma)) & \text{\&-intro}
\end{array}$$

Using this equivalence, we can also handle the first one:

$$\begin{array}{rr}
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \beta\,\&\,\gamma & \text{AssCtxt}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \beta & \text{\&-elim}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathbf{\Delta}(\beta) & \text{defCons}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathbf{\Delta}(\beta)\,\&\,\beta & \text{\&-intro}\\
\mathbf{\Delta}(\beta)\,\&\,\beta\vdash_\iota \mathcal{D}(\beta)\,\&\,\mathcal{R}(\beta) & \text{induction}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathcal{D}(\beta)\,\&\,\mathcal{R}(\beta) & \text{Cut}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathcal{R}(\beta) & \text{\&-elim}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \gamma & \text{\&-elim}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathbf{\Delta}(\gamma) & \text{defCons}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathbf{\Delta}(\gamma)\,\&\,\gamma & \text{\&-intro}\\
\mathbf{\Delta}(\gamma)\,\&\,\gamma\vdash_\iota \mathcal{D}(\gamma)\,\&\,\mathcal{R}(\gamma) & \text{induction}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathcal{D}(\gamma)\,\&\,\mathcal{R}(\gamma) & \text{Cut}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathcal{R}(\gamma) & \text{\&-elim}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathcal{R}(\beta)\,\&\,\mathcal{R}(\gamma) & \text{\&-intro}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{D}(\beta\,\&\,\gamma) & \text{2nd equiv.}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\vdash_\iota \mathcal{D}(\beta\,\&\,\gamma) & \text{toCtxt}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,;\beta\,\&\,\gamma\vdash_\iota \mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta)\,\&\,\mathcal{R}(\gamma) & \text{\&-intro}\\
\mathbf{\Delta}(\beta\,\&\,\gamma)\,\&\,\beta\,\&\,\gamma\vdash_\iota \mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta)\,\&\,\mathcal{R}(\gamma) & \text{fromCtxt}
\end{array}$$

$$\begin{array}{rr}
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma) & \text{ass}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{R}(\beta\,\&\,\gamma) & \text{\&-elim}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{R}(\beta) & \text{\&-elim}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{D}(\beta\,\&\,\gamma) & \text{\&-elim}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{D}(\beta) & \text{\&-elim}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{D}(\beta)\,\&\,\mathcal{R}(\beta) & \text{\&-intro}\\
\mathcal{D}(\beta)\,\&\,\mathcal{R}(\beta)\vdash_\iota \mathbf{\Delta}(\beta)\,\&\,\beta & \text{induction}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathbf{\Delta}(\beta)\,\&\,\beta & \text{Cut}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \beta & \text{\&-elim}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{R}(\gamma) & \text{\&-elim}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{R}(\beta)\Rightarrow\mathcal{D}(\gamma) & \text{\&-elim}\\
\mathcal{D}(\beta\,\&\,\gamma)\,\&\,\mathcal{R}(\beta\,\&\,\gamma)\vdash_\iota \mathcal{D}(\gamma) & \text{MP}
\end{array}$$

$$\begin{array}{rr}
\mathcal{D}(\beta \,\&\, \gamma) \,\&\, \mathcal{R}(\beta \,\&\, \gamma) \vdash_\iota \mathcal{D}(\gamma) \,\&\, \mathcal{R}(\gamma) & \text{\&-intro} \\
\mathcal{D}(\gamma) \,\&\, \mathcal{R}(\gamma) \vdash_\iota \mathbf{\Delta}(\gamma) \,\&\, \gamma & \text{induction} \\
\mathcal{D}(\beta \,\&\, \gamma) \,\&\, \mathcal{R}(\beta \,\&\, \gamma) \vdash_\iota \mathbf{\Delta}(\gamma) \,\&\, \gamma & \text{Cut} \\
\mathcal{D}(\beta \,\&\, \gamma) \,\&\, \mathcal{R}(\beta \,\&\, \gamma) \vdash_\iota \gamma & \text{\&-elim} \\
\mathcal{D}(\beta \,\&\, \gamma) \,\&\, \mathcal{R}(\beta \,\&\, \gamma) \vdash_\iota \beta \,\&\, \gamma & \text{\&-intro} \\
\mathcal{D}(\beta \,\&\, \gamma) \,\&\, \mathcal{R}(\beta \,\&\, \gamma) \vdash_\iota \mathbf{\Delta}(\beta \,\&\, \gamma) & \text{defCons} \\
\mathcal{D}(\beta \,\&\, \gamma) \,\&\, \mathcal{R}(\beta \,\&\, \gamma) \vdash_\iota \mathbf{\Delta}(\beta \,\&\, \gamma) \,\&\, \beta \,\&\, \gamma & \text{\&-intro}
\end{array}$$

- $\alpha \equiv \forall x(\beta)$ From the induction hypothesis

$$\left\{ \begin{array}{rcl}
\mathbf{\Delta}(\beta) \,\&\, \beta & \dashv\vdash_\iota & \mathcal{D}(\beta) \,\&\, \mathcal{R}(\beta) \\
\mathbf{\Delta}(\beta) & \dashv\vdash_\iota & \mathcal{D}(\beta)
\end{array} \right.$$

we have to derive

$$\left\{ \begin{array}{rcl}
\forall x(\mathbf{\Delta}(\beta)) \,\&\, \forall x(\beta) & \dashv\vdash_\iota & \forall x(\mathcal{D}(\beta)) \,\&\, \forall x(\mathcal{R}(\beta)) \\
\forall x(\mathbf{\Delta}(\beta)) & \dashv\vdash_\iota & \forall x(\mathcal{D}(\beta))
\end{array} \right.$$

To obtain the second equivalence, we simply apply the SimGen rule.

For the first equivalence, we also apply SimGen followed by property 38. To be able to apply this property, we need to derive

$$\mathbf{\Delta}(\forall x(\mathbf{\Delta}(\beta)) \,\&\, \forall x(\beta)) \vdash_\iota \mathbf{\Delta}(\beta) \Rightarrow \mathbf{\Delta}(\beta)$$

which is easy, and

$$\mathbf{\Delta}(\forall x(\mathcal{D}(\beta)) \,\&\, \forall x(\mathcal{R}(\beta))) \vdash_\iota \mathcal{D}(\beta) \Rightarrow \mathbf{\Delta}(\mathcal{R}(\beta))$$

which is void, since $\mathbf{\Delta}(\mathcal{R}(\beta))$ is $\top$ and hence so is the right hand side of the sequent, so it is to be dropped.

□

Theorem 49 *If the Hermes calculus is complete, then so is the* PITFOL *calculus.*

Proof.

Remember that we have reduced the problem to the following: given

$$\left\{ \begin{array}{rcl}
\mathcal{D}(\gamma_1) \,\&\, \mathcal{R}(\gamma_1), \ldots, \mathcal{D}(\gamma_m) \,\&\, \mathcal{R}(\gamma_m) & \vdash & \mathcal{D}(\alpha) \,\&\, \mathcal{R}(\alpha) \\
 & \vdash & \mathcal{D}(\gamma_1) \\
 & \vdash & \mathcal{D}(\gamma_2) \\
 & \vdash & \ldots \\
 & \vdash & \mathcal{D}(\gamma_m)
\end{array} \right.$$

we have to derive $\Gamma \vdash_\iota \alpha$.

Note that, since each Hermes proof is also a PITFOL proof, we have

$$\left\{\begin{array}{lll} \mathcal{D}(\gamma_1)\,\&\,\mathcal{R}(\gamma_1)\,,\ldots,\mathcal{D}(\gamma_m)\,\&\,\mathcal{R}(\gamma_m) & \vdash_\iota & \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha) \\ & \vdash_\iota & \mathcal{D}(\gamma_1) \\ & \vdash_\iota & \mathcal{D}(\gamma_2) \\ & \vdash_\iota & \ldots \\ & \vdash_\iota & \mathcal{D}(\gamma_m) \end{array}\right.$$

We then derive the required sequent as follows:

$$\begin{array}{rll} \boldsymbol{\Delta}(\gamma_1)\,\&\,\gamma_1 \vdash_\iota \mathcal{D}(\gamma_1)\,\&\,\mathcal{R}(\gamma_1) & \text{Th. 48} \\ \boldsymbol{\Delta}(\gamma_1)\,\&\,\gamma_1, \mathcal{D}(\gamma_2)\,\&\,\mathcal{R}(\gamma_2)\,,\ldots,\mathcal{D}(\gamma_m)\,\&\,\mathcal{R}(\gamma_m) \vdash_\iota \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha) & \text{Cut} \\ \boldsymbol{\Delta}(\gamma_2)\,\&\,\gamma_2 \vdash_\iota \mathcal{D}(\gamma_2)\,\&\,\mathcal{R}(\gamma_2) & \text{Th. 48} \\ \boldsymbol{\Delta}(\gamma_1)\,\&\,\gamma_1, \boldsymbol{\Delta}(\gamma_2)\,\&\,\gamma_2, & \\ \mathcal{D}(\gamma_3)\,\&\,\mathcal{R}(\gamma_3)\,,\ldots,\mathcal{D}(\gamma_m)\,\&\,\mathcal{R}(\gamma_m) \vdash_\iota \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha) & \text{Cut} \\ \vdots & \\ \boldsymbol{\Delta}(\gamma_1)\,\&\,\gamma_1,\ldots,\boldsymbol{\Delta}(\gamma_m)\,\&\,\gamma_m \vdash_\iota \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha) & \text{Cut} \\ \mathcal{D}(\alpha)\,\&\,\mathcal{R}(\alpha) \vdash_\iota \boldsymbol{\Delta}(\alpha)\,\&\,\alpha & \text{Th. 48} \\ \boldsymbol{\Delta}(\gamma_1)\,\&\,\gamma_1,\ldots,\boldsymbol{\Delta}(\gamma_m)\,\&\,\gamma_m \vdash_\iota \boldsymbol{\Delta}(\alpha)\,\&\,\alpha & \text{Cut} \\ \boldsymbol{\Delta}(\gamma_1)\,\&\,\gamma_1,\ldots,\boldsymbol{\Delta}(\gamma_m)\,\&\,\gamma_m \vdash_\iota \alpha & \&\text{-elim} \\ \mathcal{D}(\gamma_1) \vdash_\iota \boldsymbol{\Delta}(\gamma_1) & \text{Th. 48} \\ \vdash_\iota \boldsymbol{\Delta}(\gamma_1) & \text{Cut} \\ \boldsymbol{\Delta}(\gamma_1)\,,\gamma_1,\boldsymbol{\Delta}(\gamma_2)\,\&\,\gamma_2,\ldots,\boldsymbol{\Delta}(\gamma_m)\,\&\,\gamma_m \vdash_\iota \alpha & \text{AnDc} \\ \gamma_1,\boldsymbol{\Delta}(\gamma_2)\,\&\,\gamma_2,\ldots,\boldsymbol{\Delta}(\gamma_m)\,\&\,\gamma_m \vdash_\iota \alpha & \text{Cut} \\ \vdots & \\ \Gamma \vdash_\iota \alpha & \text{Cut} \end{array}$$

□

Chapter 4

Refining substitution

The calculus as presented so far has a rather 'crude' handling of substitution [Vernaeve & Hoogewijs 2007b].

For example, consider the term

$$\tau \equiv \iota z_{x=0 \vee y \neq 0}((x \neq 0 \Rightarrow x = y \cdot z) \,\&\, (x = 0 \Rightarrow z = 0))$$

Under the intended interpretation of our running example, this term denotes "$\frac{x}{y}$" where we define "$\frac{0}{y}$" (and in particular "$\frac{0}{0}$") to be 0, i.e., a kind of 'top to bottom with short circuit evaluation division'.

Now consider $[t/y]\tau$, which yields

$$\iota z_{\mathbf{\Delta}(t) \& x=0 \vee t \neq 0}((x \neq 0 \Rightarrow x = t \cdot z) \,\&\, (x = 0 \Rightarrow z = 0))$$

However, this requires t to be defined even when $x = 0$, which contradicts the short circuit evaluation idea; we would have liked to get instead the term

$$\iota z_{\mathbf{\Delta}(x=0 \vee t \neq 0) \& (x=0 \vee t \neq 0)}((x \neq 0 \Rightarrow x = t \cdot z) \,\&\, (x = 0 \Rightarrow z = 0))$$

i.e.,

$$\iota z_{(x \neq 0 \Rightarrow \mathbf{\Delta}(t)) \& (x=0 \vee t \neq 0)}((x \neq 0 \Rightarrow x = t \cdot z) \,\&\, (x = 0 \Rightarrow z = 0))$$

which obeys the short circuit evaluation.

Not only the operation of substitution can be refined, the substitution rule has a similar defect. For empty contexts and antecedents, the substitution rule reduces to

$$\frac{\boxed{\text{subst}} \quad UC(t) \quad \vdash_\iota \alpha}{\mathbf{\Delta}(t) \,;\, \vdash_\iota [t/x]\,\alpha}$$

whereas for similar reasons, we would prefer

$$\frac{\boxed{\text{subst'}} \quad UC(t) \quad \vdash_\iota \alpha}{\mathbf{\Delta}([t/x]\alpha)\,;\vdash_\iota [t/x]\alpha}$$

For example, given $\vdash_\iota y > 0 \Rightarrow x^2 \cdot y \geq 0$, the substitution rule with $t \equiv \iota x_{y\neq 0}(x \cdot y = 1)$ yields

$$y \neq 0 \vdash_\iota y > 0 \Rightarrow (\iota x_{y\neq 0}(x \cdot y = 1))^2 \cdot y \geq 0$$

whereas we would like to get

$$y > 0 \Rightarrow y \neq 0 \vdash_\iota y > 0 \Rightarrow (\iota x_{y\neq 0}(x \cdot y = 1))^2 \cdot y \geq 0$$

because here the antecedent is a validity in the theory of real numbers.

Using the deduction rules we have obtained until now, it is possible to introduce defined symbols g for which $\mathbf{\Delta}(g(t_1, t_2, \ldots, t_n)) \equiv \mathbf{\Delta}(t_1)\,\&\,\mathbf{\Delta}(t_2)\,\&\,\cdots\&\,\mathbf{\Delta}(t_n)\&G$, where G is a formula that depends on the terms $t_1, \ldots, t_n$ and the exact definition of g. We will not go into detail about this process; as an example, we could introduce the defined symbol div as $\mathrm{div}(x, y) = \iota z_{y\neq 0}(x \cdot y = 1)$ and then it would turn out that $\mathbf{\Delta}(\mathrm{div}(t_1, t_2)) \equiv \mathbf{\Delta}(t_1)\ \&\ \mathbf{\Delta}(t_2)\ \&\ [t_2/y]\mathbf{\Delta}(\iota z_{y\neq 0}(x \cdot y = 1)) \equiv \mathbf{\Delta}(t_1)\ \&\ \mathbf{\Delta}(t_2)\ \&\ t_2 \neq 0$. We observe that these defined symbols are **strict** with respect to undefined values, i.e., if one of the arguments t_i is undefined, then so is the whole expression $g(t_1, t_2, \ldots, t_n)$.

In the sequel, we will develop a refined version of substitution and show that we can use this to overcome this limitation and introduce non-strict defined symbols.

The next lemma generalises the method we used in §3.1 to 'fix' the definitions of $\sqrt{z}$ and $\frac{x}{y}$:

Lemma 50 *Given $\psi \vdash_\iota \exists!x(\varphi)$ and x is not a free variable of ψ, we can derive $\vdash_\iota \exists!x((\psi \Rightarrow \varphi)\ \&\ (\neg\psi \Rightarrow x = y))$ where $y \not\equiv x$.*

Proof.

To simplify the notation, we will set $\varphi' \equiv (\psi \Rightarrow \varphi)\ \&\ (\neg\psi \Rightarrow x = y)$. First, we derive $\vdash_\iota \exists x(\varphi')$:

$$
\begin{array}{rl}
\psi \vdash_\iota \exists x(\varphi) & \text{\&-elim} \\
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\
\psi \vdash_\iota \boldsymbol{\Delta}(\forall x(\varphi)) & \text{defCons} \\
\psi \vdash_\iota \boldsymbol{\Delta}(\varphi) & \forall\text{-elim} \\
\vdash_\iota \psi \Rightarrow \boldsymbol{\Delta}(\varphi) & \text{DdRu2} \\
\vdash_\iota \boldsymbol{\Delta}(\psi \,\&\, \varphi) & \text{\&-intro} \\
\psi \Rightarrow \varphi \vdash_\iota \boldsymbol{\Delta}(\psi) & \text{Weak} \\
\vdash_\iota (\psi \Rightarrow \varphi) \Rightarrow \boldsymbol{\Delta}(\psi) & \text{DdRu2} \\
\vdash_\iota \boldsymbol{\Delta}(\varphi') & \text{\&-intro} \\
\vdash_\iota \boldsymbol{\Delta}(\forall x(\varphi')) & \forall\text{-intro} \\
\forall x(\neg\varphi') \vdash_\iota \forall x(\neg\varphi') & \text{ass} \\
\forall x(\neg\varphi') \vdash_\iota \neg\varphi' & \forall\text{-elim} \\
\neg\psi \,\&\, \neg(x = y) \vdash_\iota \neg\psi \,\&\, \neg(x = y) & \text{ass} \\
\neg\psi \,\&\, \neg(x = y) \vdash_\iota \neg\psi & \text{\&-elim} \\
\psi \,\&\, \varphi \vdash_\iota \psi \,\&\, \varphi & \text{ass} \\
\psi \,\&\, \varphi \vdash_\iota \psi & \text{\&-elim} \\
\psi \,\&\, \varphi, \neg\psi \,\&\, \neg(x = y) \vdash_\iota \neg(\neg\psi \,\&\, \neg(x = y)) & \text{contra} \\
\psi \,\&\, \varphi \vdash_\iota \neg(\neg\psi \,\&\, \neg(x = y)) & \text{SeDe} \\
\psi \,\&\, \neg\varphi \vdash_\iota \psi \,\&\, \neg\varphi & \text{ass} \\
\psi \,\&\, \neg\varphi \vdash_\iota \neg\varphi & \text{\&-elim} \\
\psi \,\&\, \varphi \vdash_\iota \varphi & \text{\&-elim} \\
\psi \,\&\, \varphi, \psi \,\&\, \neg\varphi \vdash_\iota \neg(\psi \,\&\, \neg\varphi) & \text{contra} \\
\psi \,\&\, \varphi \vdash_\iota \neg(\psi \,\&\, \neg\varphi) & \text{SeDe} \\
\psi \,\&\, \varphi \vdash_\iota \varphi' & \text{\&-intro} \\
\psi; \varphi \vdash_\iota \varphi' & \text{toCtxt} \\
\psi; \neg\varphi' \vdash_\iota \neg\varphi & \text{CoPo1} \\
\psi \,\&\, \neg\varphi' \vdash_\iota \neg\varphi & \text{fromCtxt} \\
\psi, \neg\varphi' \vdash_\iota \neg\varphi & \text{AnDc} \\
\psi, \forall x(\neg\varphi') \vdash_\iota \neg\varphi & \text{Cut} \\
\psi, \forall x(\neg\varphi') \vdash_\iota \forall x(\neg\varphi) & \forall\text{-intro} \\
\psi, \forall x(\neg\varphi') \vdash_\iota \exists x(\varphi') & \text{contra} \\
\psi \vdash_\iota \exists x(\varphi') & \text{SeDe} \\
\psi \,\&\, \neg\varphi \vdash_\iota \psi \,\&\, \neg\varphi & \text{ass} \\
\psi \,\&\, \neg\varphi \vdash_\iota \psi & \text{\&-elim} \\
\neg\psi \vdash_\iota \neg\psi & \text{ass} \\
\neg\psi, \psi \,\&\, \neg\varphi \vdash_\iota \neg(\psi \,\&\, \neg\varphi) & \text{contra} \\
\neg\psi \vdash_\iota \neg(\psi \,\&\, \neg\varphi) & \text{SeDe} \\
\neg\psi \,\&\, \neg(x = y) \vdash_\iota \neg\psi \,\&\, \neg(x = y) & \text{ass} \\
\neg\psi \,\&\, \neg(x = y) \vdash_\iota \neg(x = y) & \text{\&-elim} \\
x = y \vdash_\iota x = y & \text{ass} \\
x = y, \neg\psi \,\&\, \neg(x = y) \vdash_\iota \neg(\neg\psi \,\&\, \neg(x = y)) & \text{contra} \\
x = y \vdash_\iota \neg(\neg\psi \,\&\, \neg(x = y)) & \text{SeDe}
\end{array}
$$

$$\begin{array}{rl}
\neg\psi, x = y \vdash_\iota \varphi' & \text{\&-intro} \\
\neg\psi, \neg\varphi' \vdash_\iota \neg(x = y) & \text{CoPo1} \\
\neg\psi, \forall x(\neg\varphi') \vdash_\iota \neg(x = y) & \text{Cut} \\
\neg\psi, \forall x(\neg\varphi') \vdash_\iota \neg(x = x) & \text{subst} \\
\vdash_\iota x = x & \text{eq} \\
\neg\psi, \forall x(\neg\varphi') \vdash_\iota \exists x(\varphi') & \text{contra} \\
\neg\psi \vdash_\iota \exists x(\varphi') & \text{SeDe} \\
\vdash_\iota \exists x(\varphi') & \text{rem}
\end{array}$$

Note that this derivation is also correct when $x \equiv y$; however, to complete the derivation of $\vdash_\iota \exists!x(\varphi')$, we next derive $\vdash_\iota \forall x\forall z((\varphi' \,\&\, [z/x]\varphi') \Rightarrow x = z)$. Here, we will use the condition that $x \not\equiv y$.

Note that in the last part of the proof as given below, w is not a free variable of φ and we choose u such that it is not a free variable of ψ and φ.

$$\begin{array}{rl}
\psi \vdash_\iota \forall x\forall w((\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w) & \text{\&-elim} \\
\psi \vdash_\iota \forall w((\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w) & \forall\text{-elim} \\
\psi \vdash_\iota \varphi \,\&\, [w/x]\varphi \Rightarrow x = w & \forall\text{-elim} \\
\mathbf{\Delta}(\dots); \forall x\forall w((\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w) \vdash_\iota \forall x\forall w((\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w) & \text{AssCtxt} \\
\mathbf{\Delta}(\dots); \forall x\forall w((\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w) \vdash_\iota \forall w((\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w) & \forall\text{-elim} \\
\mathbf{\Delta}(\dots); \forall x\forall w((\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w) \vdash_\iota (\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w & \forall\text{-elim} \\
\mathbf{\Delta}(\dots); \forall x\forall w((\varphi \,\&\, [w/x]\varphi) \Rightarrow x = w) \vdash_\iota (\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z & \text{subst} \\
\psi \vdash_\iota (\varphi \,\&\, [z/x]\varphi) \Rightarrow x = z & \text{Cut3} \\
\psi; \varphi \,\&\, [z/x]\varphi \vdash_\iota x = z & \text{DdRu1} \\
\psi \,\&\, \varphi \,\&\, [z/x]\varphi \vdash_\iota x = z & \text{fromCtxt} \\
\psi, \varphi \,\&\, [z/x]\varphi \vdash_\iota x = z & \text{AnDC} \\
\psi \vdash_\iota \psi & \text{ass} \\
\psi \Rightarrow \varphi \vdash_\iota \psi \Rightarrow \varphi & \text{ass} \\
\psi, \psi \Rightarrow \varphi \vdash_\iota \varphi & \text{MP} \\
\psi \Rightarrow [z/x]\varphi \vdash_\iota \psi \Rightarrow [z/x]\varphi & \text{subst} \\
\psi, \psi \Rightarrow [z/x]\varphi \vdash_\iota [z/x]\varphi & \text{MP} \\
\psi, \psi \Rightarrow \varphi, \psi \Rightarrow [z/x]\varphi \vdash_\iota \varphi \,\&\, [z/x]\varphi & \text{\&-intro} \\
\psi, \psi \Rightarrow \varphi, \psi \Rightarrow [z/x]\varphi \vdash_\iota x = z & \text{Cut} \\
\neg\psi \vdash_\iota \neg\psi & \text{ass} \\
\neg\psi \Rightarrow (x = y) \vdash_\iota \neg\psi \Rightarrow (x = y) & \text{ass} \\
\neg\psi, \neg\psi \Rightarrow (x = y) \vdash_\iota x = y & \text{MP} \\
\neg\psi \Rightarrow (z = y) \vdash_\iota \neg\psi \Rightarrow (z = y) & \text{ass} \\
\neg\psi, \neg\psi \Rightarrow (z = y) \vdash_\iota z = y & \text{MP} \\
\neg\psi, \neg\psi \Rightarrow (z = y) \vdash_\iota y = z & \text{ESy2} \\
\neg\psi, \neg\psi \Rightarrow (x = y), \neg\psi \Rightarrow (z = y) \vdash_\iota x = z & \text{ET2} \\
\psi \Rightarrow \varphi, \psi \Rightarrow [z/x]\varphi, & \\
\neg\psi \Rightarrow (x = y), \neg\psi \Rightarrow (z = y) \vdash_\iota x = z & \text{rem} \\
\varphi', \psi \Rightarrow [z/x]\varphi, \neg\psi \Rightarrow (z = y) \vdash_\iota x = z & \text{AnU} \\
\varphi', [z/x]\varphi' \vdash_\iota x = z & \text{AnU} \\
\varphi' \,\&\, [z/x]\varphi' \vdash_\iota x = z & \text{AnU} \\
\vdash_\iota (\varphi' \,\&\, [z/x]\varphi') \Rightarrow x = z & \text{DdRu2}
\end{array}$$

$$\begin{array}{ll}
\vdash_\iota \forall z((\varphi' \,\&\, [z/x]\varphi') \Rightarrow x = z) & \forall\text{-intro} \\
\vdash_\iota \forall x \forall z((\varphi' \,\&\, [z/x]\varphi') \Rightarrow x = z) & \forall\text{-intro} \\
\vdash_\iota \exists! x(\varphi') & \&\text{-intro}
\end{array}$$

□

Note that in general, the lemma does not hold if x is a free variable of ψ. For example, we have $\neg(x = z) \vdash_\iota \exists! x(x = z)$, whereas

$$\vdash_\iota \exists! x((\neg(x = z) \Rightarrow x = z) \,\&\, (\neg\neg(x = z) \Rightarrow x = y))$$

is clearly not derivable, for it is equivalent with

$$\vdash_\iota \exists! x(x = z \,\&\, x = y)$$

which is in turn equivalent with $\vdash_\iota y = z$ which is not derivable (semantically,it expresses that all models have only one single element).

Theorem 51 *For each formula α and term t of the* PITFOL *calculus for which the uniqueness conditions are derivable,*

$$\forall x(\boldsymbol{\Delta}(\alpha)); \boldsymbol{\Delta}([t/x]\,\alpha)\,, \forall x(\alpha) \vdash_\iota [t/x]\,\alpha$$

if the substitutions are defined.

Proof.

We first consider the case where $t \equiv \iota y_\psi(\varphi)$ and y is not a free variable of ψ. Define the term

$$t' \equiv \iota y((\psi \Rightarrow \varphi) \,\&\, (\neg\psi \Rightarrow y = a))$$

with a a variable symbol different from y. The previous lemma yields the uniqueness condition of t'.

We will show shortly that $\boldsymbol{\Delta}([t/x]\,\alpha)\,; [t'/x]\,\alpha \vdash_\iota [t/x]\,\alpha$, which we can use in the following proof

$$\begin{array}{rl}
\forall x(\boldsymbol{\Delta}(\alpha)); \forall x(\alpha) \vdash_\iota \forall x(\alpha) & \text{assCtxt} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \forall x(\alpha) \vdash_\iota \alpha & \forall\text{-elim} \\
\forall x(x = x), \forall x(\boldsymbol{\Delta}(\alpha)); \forall x(\alpha) \vdash_\iota [t'/x]\,\alpha & \text{subst} \\
\vdash_\iota x = x & \text{eq} \\
\vdash_\iota \forall x(x = x) & \forall\text{-intro} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \forall x(\alpha) \vdash_\iota [t'/x]\,\alpha & \text{CutCtxt} \\
\forall x(\boldsymbol{\Delta}(\alpha)), \boldsymbol{\Delta}([t/x]\,\alpha)\,; \forall x(\alpha) \vdash_\iota [t/x]\,\alpha & \text{Cut} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \boldsymbol{\Delta}([t/x]\,\alpha) \,\&\, \forall x(\alpha) \vdash_\iota [t/x]\,\alpha & \text{fromCtxt} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \vdash_\iota \forall x(\boldsymbol{\Delta}(\alpha)) & \text{defAnt} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \boldsymbol{\Delta}([t/x]\,\alpha)\,, \forall x(\alpha) \vdash_\iota [t/x]\,\alpha & \text{AnDc}
\end{array}$$

We have yet to show that $\mathbf{\Delta}([t/x]\alpha)\,;[t'/x]\alpha \vdash_\iota [t/x]\alpha$; actually we will prove that for each formula α of the PITFOL calculus for which the uniqueness conditions are derivable,

$$\mathbf{\Delta}([t/x]\alpha)\,;[t'/x]\alpha \vdash_\iota [t/x]\alpha$$
$$\mathbf{\Delta}([t/x]\alpha)\,;[t/x]\alpha \vdash_\iota [t'/x]\alpha$$

if the substitutions are defined, and for each term τ of the PITFOL calculus for which the uniqueness conditions are derivable,

$$\mathbf{\Delta}([t/x]\tau) \vdash_\iota [t'/x]\tau = [t/x]\tau$$

if the substitutions are defined. We will prove this by structural induction on α and τ.

- $\tau \equiv x$ We have to show that $\psi \vdash_\iota t' = t$. The Eq-ι rule yields $\forall x(x = x); \psi \vdash_\iota t' = t$ from which the desired sequent is easily obtained.

- $\tau \equiv z$ with $z \not\equiv x$ We have to show that $\vdash_\iota z = z$, which is trivial.

- $\tau \equiv f(t_1, t_2, \ldots, t_n)$ We have to show that

  $$\mathbf{\Delta}([t/x]t_1)\&\cdots\&\mathbf{\Delta}([t/x]t_n) \vdash_\iota f([t'/x]t_1, \ldots, [t'/x]t_n) = f([t/x]t_1, \ldots, [t/x]t_n)$$

  Induction yields $\mathbf{\Delta}([t/x]t_1) \vdash_\iota [t'/x]t_1 = [t/x]t_1$ etc.; applying the ERf rule one easily gets the desired sequent.

- $\tau \equiv \iota z_\Psi(\Phi)$ with $x \equiv z$ or x not a free variable of Φ.

  - x is not a free variable of Ψ We have to prove $\Psi \vdash_\iota \tau = \tau$, which is trivial.

  - x is a free variable of Ψ We have to prove

    $$\psi \,\&\, [t/x]\,\Psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\Psi}(\Phi) = \iota z_{\psi\&[t/x]\Psi}(\Phi)$$

$$\begin{array}{rl}
\Psi \vdash_\iota \iota z_\Psi(\Phi) = \iota z_\Psi(\Phi) & \text{eq} \\
\psi; [t/x]\Psi \vdash_\iota \iota z_{\psi\&[t/x]\Psi}(\Phi) = \iota z_{\psi\&[t/x]\Psi}(\Phi) & \text{subst} \\
\psi \,\&\, [t/x]\Psi \vdash_\iota \exists! z(\Phi) & \text{UC} \\
\forall x(x=x); [t'/x]\Psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\Psi}(\Phi) = \iota z_{\forall x(x=x)\&[t'/x]\Psi}(\Phi) & \text{subst} \\
\forall x(x=x) \,\&\, [t'/x]\Psi \vdash_\iota \exists! z(\Phi) & \text{UC} \\
\forall x(x=x) \,\&\, [t'/x]\Psi, & \\
\quad \psi \,\&\, [t/x]\Psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\Psi}(\Phi) = \iota z_{\psi\&[t/x]\Psi}(\Phi) & \text{Eq-}\iota \\
\quad \vdash_\iota \boldsymbol{\Delta}(\Psi) & \text{defAnt} \\
\quad \Psi \vdash_\iota \Psi & \text{ass} \\
\quad \psi; [t/x]\Psi \vdash_\iota [t/x]\Psi & \text{subst} \\
\quad \psi; \vdash_\iota \boldsymbol{\Delta}([t/x]\Psi) & \text{defAnt} \\
\quad \boldsymbol{\Delta}([t/x]\Psi); [t/x]\Psi \vdash_\iota [t'/x]\Psi & \text{induction} \\
\quad \psi; [t/x]\Psi \vdash_\iota [t'/x]\Psi & \text{CutCtxt} \\
\quad \psi \,\&\, [t/x]\Psi \vdash_\iota [t'/x]\Psi & \text{fromCtxt} \\
\quad \vdash_\iota x = x & \text{eq} \\
\quad \vdash_\iota \forall x(x=x) & \forall\text{-intro} \\
\quad \psi \,\&\, [t/x]\Psi \vdash_\iota \forall x(x=x) \,\&\, [t'/x]\Psi & \&\text{-intro} \\
\quad \psi \,\&\, [t/x]\Psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\Psi}(\Phi) = \iota z_{\psi\&[t/x]\Psi}(\Phi) & \text{Cut}
\end{array}$$

- $\tau \equiv \iota z_\Psi(\Phi)$ with $x \not\equiv z$ and x a free variable of Φ We have to derive

$$\psi \,\&\, [t/x]\Psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\Psi}([t'/x]\Phi) = \iota z_{\psi\&[t/x]\Psi}([t/x]\Phi)$$

Using induction on Φ, we easily derive the interchangeability conditions to show that $[t'/x]\Phi$ and $[t/x]\Phi$ are interchangeable under the context $\forall z(\boldsymbol{\Delta}([t/x]\Phi))$:

$$\begin{cases}
\forall z(\boldsymbol{\Delta}([t/x]\Phi)), \boldsymbol{\Delta}([t'/x]\Phi); [t'/x]\Phi \vdash_\iota [t/x]\Phi \\
\forall z(\boldsymbol{\Delta}([t/x]\Phi)), \boldsymbol{\Delta}([t/x]\Phi); [t/x]\Phi \vdash_\iota [t'/x]\Phi \\
\qquad \forall z(\boldsymbol{\Delta}([t/x]\Phi)); \boldsymbol{\Delta}([t'/x]\Phi) \vdash_\iota \boldsymbol{\Delta}([t/x]\Phi) \\
\qquad \forall z(\boldsymbol{\Delta}([t/x]\Phi)); \boldsymbol{\Delta}([t/x]\Phi) \vdash_\iota \boldsymbol{\Delta}([t'/x]\Phi)
\end{cases}$$

which we can use in the following proof:

$$\begin{array}{rl}
\Psi \vdash_\iota \iota z_\Psi(\Phi) = \iota z_\Psi(\Phi) & \text{eq} \\
\forall x(x=x); [t'/x]\Psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\Psi}([t'/x]\Phi) & \\
\qquad = \iota z_{\forall x(x=x)\&[t'/x]\Psi}([t'/x]\Phi) & \text{subst} \\
\forall x(x=x) \,\&\, [t'/x]\Psi \vdash_\iota \exists! z([t'/x]\Phi) & \text{UC} \\
\psi; [t/x]\Psi \vdash_\iota \iota z_{\psi\&[t/x]\Psi}([t/x]\Phi) = \iota z_{\psi\&[t/x]\Psi}([t/x]\Phi) & \text{subst} \\
\psi \,\&\, [t/x]\Psi \vdash_\iota \exists! z([t/x]\Phi) & \text{UC}
\end{array}$$

$$\begin{array}{rl}
\psi \,\&\, [t/x]\Psi \vdash_\iota \forall z(\boldsymbol{\Delta}([t/x]\Phi)) \,\&\, (\exists z([t/x]\Phi) \Rightarrow \boldsymbol{\Delta}(\dots)) & \text{defCons}\\
\psi \,\&\, [t/x]\Psi \vdash_\iota \forall z(\boldsymbol{\Delta}([t/x]\Phi)) & \text{\&-elim}\\
\boldsymbol{\Delta}([t/x]\Phi)\,;[t/x]\Phi \vdash_\iota [t'/x]\Phi & \text{induction}\\
\boldsymbol{\Delta}([t/x]\Phi)\,;\vdash_\iota [t/x]\Phi \Rightarrow [t'/x]\Phi & \text{DdRu2}\\
\boldsymbol{\Delta}([t/x]\Phi)\,;[t'/x]\Phi \vdash_\iota [t/x]\Phi & \text{induction}\\
\boldsymbol{\Delta}([t/x]\Phi)\,;\vdash_\iota [t'/x]\Phi \Rightarrow [t/x]\Phi & \text{DdRu2}\\
\boldsymbol{\Delta}([t/x]\Phi)\,;\vdash_\iota [t'/x]\Phi \Leftrightarrow [t/x]\Phi & \text{\&-intro}\\
\boldsymbol{\Delta}([t/x]\Phi) \vdash_\iota [t'/x]\Phi \Leftrightarrow [t/x]\Phi & \text{fromCtxt}\\
\forall z(\boldsymbol{\Delta}([t/x]\Phi)) \vdash_\iota \forall z([t'/x]\Phi \Leftrightarrow [t/x]\Phi) & \text{SimGen}\\
\psi \,\&\, [t/x]\Psi \vdash_\iota \forall z([t'/x]\Phi \Leftrightarrow [t/x]\Phi) & \text{Cut}\\
\forall x(x=x) \,\&\, [t'/x]\Psi, \psi \,\&\, [t/x]\Psi \vdash_\iota \forall z([t'/x]\Phi \Leftrightarrow [t/x]\Phi) & \text{Weak*}\\
\forall x(x=x) \,\&\, [t'/x]\Psi, \psi \,\&\, [t/x]\Psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\Psi}([t'/x]\Phi) & \\
\qquad = \iota z_{\psi\&[t/x]\Psi}([t/x]\Phi) & \text{Eq-}\iota\\
\psi \,\&\, [t/x]\Psi \vdash_\iota \forall x(x=x) \,\&\, [t'/x]\Psi & \text{see above}\\
\psi \,\&\, [t/x]\Psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\Psi}([t'/x]\Phi) & \\
\qquad = \iota z_{\psi\&[t/x]\Psi}([t/x]\Phi) & \text{Cut}
\end{array}$$

- $\alpha \equiv p(t_1, t_2, \dots, t_n)$ We have to derive

$$\begin{cases} \boldsymbol{\Delta}([t/x]t_1) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}([t/x]t_n)\,; p([t'/x]t_1, \dots, [t'/x]t_n) \vdash_\iota p([t/x]t_1, \dots, [t/x]t_1) \\ \boldsymbol{\Delta}([t/x]t_1) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}([t/x]t_n)\,; p([t/x]t_1, \dots, [t/x]t_1) \vdash_\iota p([t'/x]t_1, \dots, [t'/x]t_n) \end{cases}$$

  Induction yields $\boldsymbol{\Delta}([t/x]t_1) \vdash_\iota [t'/x]t_1 = [t/x]t_1$ etc. Using ERp2 we get

  $$\boldsymbol{\Delta}([t/x]t_1), \dots, \boldsymbol{\Delta}([t/x]t_n)\,; p([t'/x]t_1, \dots, [t'/x]t_n) \vdash_\iota p([t/x]t_1, \dots, [t/x]t_n)$$

  from which the first sequent is easily obtained. We get the second desired sequent by first using the ESy rule before applying ERp2.

- $\alpha \equiv \neg\beta$ Induction yields

  $$\begin{cases} \boldsymbol{\Delta}([t/x]\beta)\,;[t'/x]\beta \vdash_\iota [t/x]\beta \\ \boldsymbol{\Delta}([t/x]\beta)\,;[t/x]\beta \vdash_\iota [t'/x]\beta \end{cases}$$

  and Theorem 23.1 immediately yields the required sequents.

- $\alpha \equiv \beta \,\&\, \gamma$ We have to derive

  $$\begin{cases} \boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma))\,;[t'/x](\beta \,\&\, \gamma) \vdash_\iota [t/x](\beta \,\&\, \gamma) \\ \boldsymbol{\Delta}([t/x](\beta \,\&\, \gamma))\,;[t/x](\beta \,\&\, \gamma) \vdash_\iota [t'/x](\beta \,\&\, \gamma) \end{cases}$$

  Using induction on γ, Theorem 23.1 yields

  $$\begin{cases} \boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;[t/x]\beta \,\&\, [t'/x]\gamma \vdash_\iota [t/x]\beta \,\&\, [t/x]\gamma & (1) \\ \boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;[t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [t/x]\beta \,\&\, [t'/x]\gamma & (2) \end{cases}$$

Using induction on β, the same theorem yields

$$\begin{cases} \mathbf{\Delta}([t/x]\beta \,\&\, [t'/x]\gamma)\,;[t'/x]\beta \,\&\, [t'/x]\gamma \vdash_\iota [t/x]\beta \,\&\, [t'/x]\gamma & (3) \\ \mathbf{\Delta}([t/x]\beta \,\&\, [t'/x]\gamma)\,;[t/x]\beta \,\&\, [t'/x]\gamma \vdash_\iota [t'/x]\beta \,\&\, [t'/x]\gamma & (4) \end{cases}$$

using which we have the following proofs:

$$\begin{array}{rl} \mathbf{\Delta}([t/x]\beta \,\&\, [t'/x]\gamma)\,,\mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; & \\ \qquad [t'/x]\beta \,\&\, [t'/x]\gamma \vdash_\iota [t/x]\beta \,\&\, [t/x]\gamma & \text{Cut(1)(3)} \\ \qquad \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\vdash_\iota \mathbf{\Delta}([t/x]\beta \,\&\, [t'/x]\gamma) & \text{defAnt(1)} \\ \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;[t'/x]\beta \,\&\, [t'/x]\gamma \vdash_\iota [t/x]\beta \,\&\, [t/x]\gamma & \text{CutCtxt} \end{array}$$

$$\begin{array}{rl} \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,,\mathbf{\Delta}([t/x]\beta \,\&\, [t'/x]\gamma)\,; & \\ \qquad [t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [t'/x]\beta \,\&\, [t'/x]\gamma & \text{Cut(2)(4)} \\ \qquad \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\vdash_\iota \mathbf{\Delta}([t/x]\beta \,\&\, [t'/x]\gamma) & \text{defAnt(1)} \\ \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;[t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [t'/x]\beta \,\&\, [t'/x]\gamma & \text{CutCtxt} \end{array}$$

- $\alpha \equiv \forall x(\beta)$ We only have to derive the single sequent

$$\mathbf{\Delta}(\alpha)\,;\alpha \vdash_\iota \alpha$$

which we immediately get using the AssCtxt rule.

- $\alpha \equiv \forall y(\beta)$ with $x \not\equiv y$ We have to derive the two sequents

$$\begin{cases} \forall y(\mathbf{\Delta}([t/x]\beta));\forall y([t'/x]\beta) \vdash_\iota \forall y([t/x]\beta) \\ \forall y(\mathbf{\Delta}([t/x]\beta));\forall y([t/x]\beta) \vdash_\iota \forall y([t'/x]\beta) \end{cases}$$

which we easily obtain using induction on β and theorem 23.1.

Next, we consider the case where $t \equiv \iota y_\psi(\varphi)$ and y is a free variable of ψ. Applying the previous case with $t = \iota z_\psi([z/y]\varphi)$ where z not a free variable of ψ and φ, yields

$$\forall x(\mathbf{\Delta}(\alpha));\mathbf{\Delta}([\iota z_\psi([z/y]\varphi)/x]\alpha)\,,\forall x(\alpha) \vdash_\iota [\iota z_\psi([z/y]\varphi)/x]\alpha$$

Using corollary 43 and theorem 44, we get the desired sequent.

Now we are left to handle the general case where t is not a ι-term. We can apply the previous case to the ι-term $\iota y_{\mathbf{\Delta}(t)}(y = t)$ where y is not a free variable of t. In order to do this, we need to derive its uniqueness condition:

$$\begin{array}{rr}
\boldsymbol{\Delta}(t) \vdash_\iota \exists y(y = t) & \text{Existence} \\
\boldsymbol{\Delta}(t)\,; y = t, t = z \vdash_\iota y = z & \text{ET} \\
\boldsymbol{\Delta}(t)\,; z = t \vdash_\iota t = z & \text{ESy} \\
\boldsymbol{\Delta}(t)\,; y = t, z = t \vdash_\iota y = z & \text{Cut} \\
\boldsymbol{\Delta}(t)\,; y = t \,\&\, z = t \vdash_\iota y = z & \text{AnU} \\
\boldsymbol{\Delta}(t)\,; \vdash_\iota (y = t \,\&\, z = t) \Rightarrow y = z & \text{DdRu2} \\
\boldsymbol{\Delta}(t) \vdash_\iota (y = t \,\&\, z = t) \Rightarrow y = z & \text{fromCtxt} \\
\boldsymbol{\Delta}(t) \vdash_\iota \forall z((y = t \,\&\, z = t) \Rightarrow y = z) & \forall\text{-intro} \\
\boldsymbol{\Delta}(t) \vdash_\iota \forall y \forall z((y = t \,\&\, z = t) \Rightarrow y = z) & \forall\text{-intro} \\
\boldsymbol{\Delta}(t) \vdash_\iota \exists! y(y = t) & \&\text{-intro}
\end{array}$$

This yields

$$\forall x(\boldsymbol{\Delta}(\alpha)); \boldsymbol{\Delta}\left(\left[\iota y_{\boldsymbol{\Delta}(t)}(y = t)/x\right]\alpha\right), \forall x(\alpha) \vdash_\iota \left[\iota y_{\boldsymbol{\Delta}(t)}(y = t)/x\right]\alpha$$

If we can show that $\iota y_{\boldsymbol{\Delta}(t)}(y = t)$ is interchangeable with t, an application of theorem 44 proves this case. The required sequent is derived as follows:

$$\begin{array}{rr}
\boldsymbol{\Delta}(t) \vdash_\iota \iota y_{\boldsymbol{\Delta}(t)}(y = t) = \widetilde{t} & \text{iota} \\
\boldsymbol{\Delta}(t) \vdash_\iota t = \widetilde{t} & \text{Th. 25.3} \\
\boldsymbol{\Delta}(t) \vdash_\iota \widetilde{t} = t & \text{ESy2} \\
\boldsymbol{\Delta}(t) \vdash_\iota \iota y_{\boldsymbol{\Delta}(t)}(y = t) = t & \text{ET2}
\end{array}$$

□

We are now easily able to present a more refined version of the substitution rule:

$$\begin{array}{rr}
\boxed{\text{Subst2}} & \\
UC(t) & \text{prem} \\
\vdash_\iota \alpha & \text{prem} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \boldsymbol{\Delta}([t/x]\alpha), \forall x(\alpha) \vdash_\iota [t/x]\alpha & \text{Th. 51} \\
\vdash_\iota \forall x(\alpha) & \forall\text{-intro} \\
\forall x(\boldsymbol{\Delta}(\alpha)); \boldsymbol{\Delta}([t/x]\alpha) \vdash_\iota [t/x]\alpha & \text{Cut} \\
\vdash_\iota \boldsymbol{\Delta}(\forall x(\alpha)) & \text{defCons} \\
\boldsymbol{\Delta}([t/x]\alpha) \vdash_\iota [t/x]\alpha & \text{CutCtxt}
\end{array}$$

However, we can generalise this. With the same definition of t' as in theorem 51, we get

$$
\begin{array}{rl}
\boxed{\text{Subst2}} & \\
UC(t) & \text{prem} \\
\sigma_1, \sigma_2, \ldots, \sigma_n; \gamma_1, \gamma_2, \ldots, \gamma_m \vdash_\iota \alpha & \text{prem} \\
[t'/x]\sigma_1, [t'/x]\sigma_2, \ldots, [t'/x]\sigma_n; [t'/x]\gamma_1, [t'/x]\gamma_2, \ldots, [t'/x]\gamma_m \vdash_\iota [t'/x]\alpha & \text{subst} \\
\mathbf{\Delta}([t/x]\gamma_m)\,; [t/x]\gamma_m \vdash_\iota [t'/x]\gamma_m & \text{Th. 51} \\
\mathbf{\Delta}([t/x]\gamma_m)\,, [t'/x]\sigma_1, \ldots, [t'/x]\sigma_n; [t'/x]\gamma_1, [t'/x]\gamma_2, \ldots, [t/x]\gamma_m \vdash_\iota [t'/x]\alpha & \text{Cut} \\
\vdots & \\
\mathbf{\Delta}([t/x]\gamma_2)\,, \ldots, \mathbf{\Delta}([t/x]\gamma_m)\,, & \\
[t'/x]\sigma_1, \ldots, [t'/x]\sigma_n; [t'/x]\gamma_1, [t/x]\gamma_2, \ldots, [t/x]\gamma_m \vdash_\iota [t'/x]\alpha & \text{Cut} \\
\mathbf{\Delta}([t/x]\gamma_1)\,; [t/x]\gamma_1 \vdash_\iota [t'/x]\gamma_1 & \text{Th. 51} \\
\mathbf{\Delta}([t/x]\gamma_1)\,, \ldots, \mathbf{\Delta}([t/x]\gamma_m)\,, & \\
[t'/x]\sigma_1, [t'/x]\sigma_2, \ldots, [t'/x]\sigma_n; [t/x]\gamma_1, [t/x]\gamma_2, \ldots, [t/x]\gamma_m \vdash_\iota [t'/x]\alpha & \text{Cut} \\
\mathbf{\Delta}([t/x]\sigma_n)\,; [t/x]\sigma_n \vdash_\iota [t'/x]\sigma_n & \text{Th. 51} \\
\mathbf{\Delta}([t/x]\sigma_n)\,, \mathbf{\Delta}([t/x]\gamma_1)\,, \ldots, \mathbf{\Delta}([t/x]\gamma_m)\,, & \\
[t'/x]\sigma_1, [t'/x]\sigma_2, \ldots, [t/x]\sigma_n; [t/x]\gamma_1, \ldots, [t/x]\gamma_m \vdash_\iota [t'/x]\alpha & \text{CutCtxt} \\
\vdots & \\
\mathbf{\Delta}([t/x]\sigma_2)\,, \ldots \mathbf{\Delta}([t/x]\sigma_n)\,, \mathbf{\Delta}([t/x]\gamma_1)\,, \ldots, \mathbf{\Delta}([t/x]\gamma_m)\,, & \\
[t'/x]\sigma_1, [t/x]\sigma_2, \ldots, [t/x]\sigma_n; [t/x]\gamma_1, \ldots, [t/x]\gamma_m \vdash_\iota [t'/x]\alpha & \text{CutCtxt} \\
\mathbf{\Delta}([t/x]\sigma_1)\,; [t/x]\sigma_1 \vdash_\iota [t'/x]\sigma_1 & \text{Th. 51} \\
\mathbf{\Delta}([t/x]\sigma_1)\,, \ldots \mathbf{\Delta}([t/x]\sigma_n)\,, \mathbf{\Delta}([t/x]\gamma_1)\,, \ldots, \mathbf{\Delta}([t/x]\gamma_m)\,, & \\
[t/x]\sigma_1, [t/x]\sigma_2, \ldots, [t/x]\sigma_n; [t/x]\gamma_1, \ldots, [t/x]\gamma_m \vdash_\iota [t'/x]\alpha & \text{CutCtxt} \\
\mathbf{\Delta}([t/x]\alpha)\,; [t'/x]\alpha \vdash_\iota [t/x]\alpha & \text{Th. 51} \\
\mathbf{\Delta}([t/x]\sigma_1)\,, \ldots \mathbf{\Delta}([t/x]\sigma_n)\,, \mathbf{\Delta}([t/x]\gamma_1)\,, \ldots, \mathbf{\Delta}([t/x]\gamma_m)\,, & \\
[t/x]\sigma_1, [t/x]\sigma_2, \ldots, [t/x]\sigma_n, \mathbf{\Delta}([t/x]\alpha)\,; [t/x]\gamma_1, \ldots, [t/x]\gamma_m \vdash_\iota [t/x]\alpha & \text{CutCtxt}
\end{array}
$$

Note that in general, we cannot have a simpler Subst2 rule. For example, in the case $n = 0, m = 1$, naively, one could be inclined to expect that

$$
\frac{\begin{array}{c}\boxed{\text{Subst3?}} \\ UC(t) \\ \gamma \vdash_\iota \alpha\end{array}}{\mathbf{\Delta}([t/x]\gamma)\,; [t/x]\gamma \vdash_\iota [t/x]\alpha}
$$

be a valid rule. However, this expectation is proven wrong: when $\gamma \vdash_\iota \alpha$, it is still possible that $\mathbf{\Delta}([t/x]\gamma)\,; [t/x]\gamma \nvdash_\iota \mathbf{\Delta}([t/x]\alpha)$. For example, if $\gamma \equiv \forall x(x = x)$ and $\alpha \equiv x = x$, then the rule breaks down, since in general,

$$\forall x(x = x) \nvdash_\iota \mathbf{\Delta}(t = t)$$

Hence, the simplest form of the rule is indeed

$$\frac{\begin{array}{r}\boxed{\text{Subst2}}\\ UC(t)\\ \gamma \vdash_\iota \alpha\end{array}}{\boldsymbol{\Delta}([t/x]\gamma)\,,\boldsymbol{\Delta}([t/x]\alpha)\,;[t/x]\gamma \vdash_\iota [t/x]\alpha}$$

We have gained semantical accuracy but we have to pay with extra syntactical complexity.

4.1 Refined substitution

We will now define a kind of substitution which is better suited to our purposes, which we will call **refined substitution** and denote as $[\![t/x]\!]\alpha$ and $[\![t/x]\!]\tau$, where as usual, t and τ are terms of the PITFOL calculus and α is a formula of the PITFOL calculus.

The only way in which it differs from the 'crude' substitution we used up to now is the case $\tau \equiv \iota y_\psi(\varphi)$:

- $x \equiv y$ or x is not a free variable of φ
  - x is not a free variable of ψ $\qquad [\![t/x]\!]\,\iota y_\psi(\varphi) \equiv \iota y_\psi(\varphi)$
  - x is a free variable of ψ
    $$[\![t/x]\!]\,\iota y_\psi(\varphi) \equiv \begin{cases} \iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) & \text{when } \boldsymbol{\Delta}(t) \not\equiv \top \\ \iota y_{[\![t/x]\!]\psi}(\varphi) & \text{when } \boldsymbol{\Delta}(t) \equiv \top \end{cases}$$
- $x \not\equiv y$ and x is a free variable of φ and y is not a free variable of t
  - x is not a free variable of ψ
    $$[\![t/x]\!]\,\iota y_\psi(\varphi) \equiv \begin{cases} \iota y_{\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\,\varphi) & \text{when } \boldsymbol{\Delta}(t) \not\equiv \top \\ \iota y_\psi([\![t/x]\!]\,\varphi) & \text{when } \boldsymbol{\Delta}(t) \equiv \top \end{cases}$$
  - x is a free variable of ψ
    $$[\![t/x]\!]\,\iota y_\psi(\varphi) \equiv \begin{cases} \iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\,\varphi) & \text{when } \boldsymbol{\Delta}(t) \not\equiv \top \\ \iota y_{[\![t/x]\!]\psi}([\![t/x]\!]\,\varphi) & \text{when } \boldsymbol{\Delta}(t) \equiv \top \end{cases}$$
- If $x \not\equiv y$ and x is a free variable of φ and y is a free variable of t, then $[\![t/x]\!]\,\iota y_\psi(\varphi)$ is undefined; we say that the substitution would capture the free variable y of t.

Note that the inclusion of the conjunct $\forall y(\boldsymbol{\Delta}([\![t/x]\!]\,\varphi))$ in the domain formula of certain cases is necessary to guarantee that when the uniqueness condition for $\iota y_\psi(\varphi)$ is derivable, then so is the uniqueness condition for

$[\![t/x]\!]\,\iota y_\psi(\varphi)$. For example, consider the term $\iota x_{y\neq 0}(x\cdot y = z)$ (intuitively, "$\frac{z}{y}$") and the substitution $[\![t/z]\!]\,\iota x_{y\neq 0}(x\cdot y = z)$ where t is a term of the PITFOL calculus with $\mathbf{\Delta}(t) \not\equiv \top$. If this substitution were defined as $\iota x_{y\neq 0}(x\cdot y = t)$, then its uniqueness condition, $y \neq 0 \vdash_\iota \exists! x(x\cdot y = t)$ would not be derivable, since we then easily could derive $y \neq 0 \vdash_\iota \forall x(\mathbf{\Delta}(t))$, and it is clear that in general, this is not a derivable sequent.

One easily proves the corresponding versions of properties 1–5. It is also easy to see that the substitution $[t/x]\,\alpha$ is defined if and only if the substitution $[\![t/x]\!]\,\alpha$ is defined (and analogously for $[t/x]\,\tau$ and $[\![t/x]\!]\,\tau$).

Note that there are a lot of different cases in the definition of refined substitution. The main reason for distinguishing whether $\mathbf{\Delta}(t) \equiv \top$ or not is to make sure that substituting a variable symbol x for another y consists only in changing all free occurrences of y into x.

The next theorem will reduce the number of cases, which will simplify proofs later on. Moreover, we show that theorem 51 also holds for refined substitution.

Theorem 52 *1. For all terms t and τ of the* PITFOL *calculus of which the uniqueness conditions are derivable, if $\tau \equiv \iota y_\psi(\varphi)$ and the substitution $[\![t/x]\!]\,\tau$ is defined, then $[\![t/x]\!]\,\tau$ is interchangeable with*

$$\begin{cases} \iota y_{\mathbf{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) & \text{if } x \equiv y \text{ or } x \text{ is not a free variable of } \varphi \\ \iota y_{\mathbf{\Delta}([\![t/x]\!]\psi)\&\forall y(\mathbf{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\,\varphi) & \text{if } x \not\equiv y \text{ and } x \text{ is a free variable of } \varphi \end{cases}$$

(and hence, in particular, the uniqueness condition for $[\![t/x]\!]\,\tau$ is derivable).

2. For each term t and formula α of the PITFOL *calculus of which the uniqueness conditions are derivable,*

$$\begin{cases} \mathbf{\Delta}([t/x]\,\alpha) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\,\alpha) \\ \mathbf{\Delta}(t)\,, \mathbf{\Delta}([\![t/x]\!]\,\alpha) \vdash_\iota \mathbf{\Delta}([t/x]\,\alpha) \\ \mathbf{\Delta}([t/x]\,\alpha)\,; [\![t/x]\!]\,\alpha \vdash_\iota [t/x]\,\alpha \\ \mathbf{\Delta}([t/x]\,\alpha)\,; [t/x]\,\alpha \vdash_\iota [\![t/x]\!]\,\alpha \end{cases}$$

if the substitutions are defined. For all terms t and τ of the PITFOL *calculus of which the uniqueness conditions are derivable,*

$$\begin{cases} \mathbf{\Delta}([t/x]\,\tau) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\,\tau) \\ \mathbf{\Delta}(t)\,, \mathbf{\Delta}([\![t/x]\!]\,\tau) \vdash_\iota \mathbf{\Delta}([t/x]\,\tau) \\ \mathbf{\Delta}([t/x]\,\tau) \vdash_\iota [\![t/x]\!]\,\tau = [t/x]\,\tau \end{cases}$$

3. For each formula α and each term t of the PITFOL *calculus for which the uniqueness conditions are derivable,*

$$\forall x(\boldsymbol{\Delta}(\alpha)); \boldsymbol{\Delta}([\![t/x]\!]\,\alpha)\,, \forall x(\alpha) \vdash_\iota [\![t/x]\!]\,\alpha$$

if the substitutions are defined.

Proof.

We prove all three parts simultaneously by induction on the nesting depths of the ι-terms of α and τ. This in turn we prove by induction on the complexity of α and τ.

First, we handle all possible cases for the first part.

- $x \equiv y$ or x is not a free variable of φ
  - x is not a free variable of ψ We have to show that $\iota y_\psi(\varphi)$ and $\iota y_{\boldsymbol{\Delta}(\psi)\&\psi}(\varphi)$ are interchangeable. First, we derive the uniqueness condition for $\iota y_{\boldsymbol{\Delta}(\psi)\&\psi}(\varphi)$ from the given uniqueness condition $\psi \vdash_\iota \exists!y(\varphi)$:

$$\begin{array}{rr} \boldsymbol{\Delta}(\psi)\,;\psi \vdash_\iota \psi & \text{AssCtxt} \\ \boldsymbol{\Delta}(\psi)\,\&\,\psi \vdash_\iota \psi & \text{fromCtxt} \\ \boldsymbol{\Delta}(\psi)\,\&\,\psi \vdash_\iota \exists!y(\varphi) & \text{Cut} \end{array}$$

    To establish interchangeability, we derive

$$\begin{array}{rr} \boldsymbol{\Delta}(\psi)\,\&\,\psi, \psi \vdash_\iota \iota y_\psi(\varphi) = \iota y_{\boldsymbol{\Delta}(\psi)\&\psi}(\varphi) & \text{Eq-}\iota \\ \vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\ \psi \vdash_\iota \psi & \text{ass} \\ \psi \vdash_\iota \boldsymbol{\Delta}(\psi)\,\&\,\psi & \&\text{-intro} \\ \psi \vdash_\iota \iota y_\psi(\varphi) = \iota y_{\boldsymbol{\Delta}(\psi)\&\psi}(\varphi) & \text{Cut} \end{array}$$

$$\begin{array}{rr} \boldsymbol{\Delta}(\psi)\,\&\,\psi, \psi \vdash_\iota \iota y_\psi(\varphi) = \iota y_{\boldsymbol{\Delta}(\psi)\&\psi}(\varphi) & \text{Eq-}\iota \\ \boldsymbol{\Delta}(\psi)\,;\psi \vdash_\iota \psi & \text{AssCtxt} \\ \boldsymbol{\Delta}(\psi)\,\&\,\psi \vdash_\iota \psi & \text{fromCtxt} \\ \boldsymbol{\Delta}(\psi)\,\&\,\psi \vdash_\iota \iota y_\psi(\varphi) = \iota y_{\boldsymbol{\Delta}(\psi)\&\psi}(\varphi) & \text{Cut} \end{array}$$

    The remaining two sequents are easy.

  - x is a free variable of ψ We first derive

$$\begin{array}{rl}
\vdash_\iota \psi \Rightarrow \exists! y(\varphi) & \text{DdRu2}\\
\forall x(\boldsymbol{\Delta}(\psi \Rightarrow \exists! y(\varphi)));\\
\boldsymbol{\Delta}([\![t/x]\!]\psi \Rightarrow \exists! y(\varphi)), \forall x(\psi \Rightarrow \exists! y(\varphi)) \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists! y(\varphi) & \text{induction}\\
\vdash_\iota \forall x(\psi \Rightarrow \exists! y(\varphi)) & \forall\text{-intro}\\
\vdash_\iota \boldsymbol{\Delta}(\forall x(\psi \Rightarrow \exists! y(\varphi))) & \text{defCons}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi \Rightarrow \exists! y(\varphi)), \forall x(\psi \Rightarrow \exists! y(\varphi)) \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists! y(\varphi) & \text{CutCtxt}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi \Rightarrow \exists! y(\varphi)) \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists! y(\varphi) & \text{Cut}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi); [\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi)) \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists! y(\varphi) & \text{toCtxt}\\
\psi \vdash_\iota \boldsymbol{\Delta}(\exists! y(\varphi)) & \text{defCons}\\
\vdash_\iota \psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi)) & \text{DdRu2}\\
\forall x(\boldsymbol{\Delta}(\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi))));\\
\boldsymbol{\Delta}([\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi))),\\
\forall x(\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi))) \vdash_\iota [\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi)) & \text{induction}\\
\vdash_\iota \forall x(\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi))) & \forall\text{-intro}\\
\vdash_\iota \boldsymbol{\Delta}(\forall x(\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi)))) & \text{defCons}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi))),\\
\forall x(\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi))) \vdash_\iota [\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi)) & \text{CutCtxt}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi))) \vdash_\iota [\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi)) & \text{Cut}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi); [\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\boldsymbol{\Delta}(\exists! y(\varphi))) \vdash_\iota [\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi)) & \text{toCtxt}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\exists! y(\varphi))) & \text{Ddef}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([\![t/x]\!]\psi)) & \text{Ddef}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{ass}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi); \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{toCtxt}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi); [\![t/x]\!]\psi \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\exists! y(\varphi))) & \text{Weak}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi); \vdash_\iota [\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\boldsymbol{\Delta}(\exists! y(\varphi))) & \text{DdRu2}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi); \vdash_\iota [\![t/x]\!]\psi \Rightarrow \boldsymbol{\Delta}(\exists! y(\varphi)) & \text{Cut}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi); \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists! y(\varphi) & \text{Cut}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi); [\![t/x]\!]\psi \vdash_\iota \exists! y(\varphi) & \text{DdRu1}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi) \,\&\, [\![t/x]\!]\psi \vdash_\iota \exists! y(\varphi) & \text{fromCtxt}
\end{array}$$

- $\boldsymbol{\Delta}(t) \not\equiv \top$ Both terms are $\iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi)$ and we already have derived the uniqueness condition.
- $\boldsymbol{\Delta}(t) \equiv \top$ We have to show that $\iota y_{[\![t/x]\!]\psi}(\varphi)$ and $\iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi)$ are interchangeable. First, we derive the uniqueness conditions. The derivation of $\boldsymbol{\Delta}([\![t/x]\!]\psi)$ & $[\![t/x]\!]\psi \vdash_\iota \exists! y(\varphi)$ is identical to the previous case. From it, we derive the other uniqueness condition as follows:

$$\begin{array}{rl}
\boldsymbol{\Delta}([t/x]\psi) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{induction}\\
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt}\\
\psi \vdash_\iota \psi & \text{ass}\\
[t/x]\psi \vdash_\iota [t/x]\psi & \text{subst}
\end{array}$$

$$\begin{array}{rll}
 & \vdash_\iota \mathbf{\Delta}([t/x]\psi) & \text{defAnt} \\
 & \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\psi) & \text{Cut} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,;[\![t/x]\!]\psi & \vdash_\iota \exists!y(\varphi) & \text{toCtxt} \\
[\![t/x]\!]\psi & \vdash_\iota \exists!y(\varphi) & \text{CutCtxt}
\end{array}$$

The interchangeability conditions are now easy to derive.

- $x \not\equiv y$ and x is a free variable of φ and y is not a free variable of t We first derive

$$\begin{array}{rll}
 & \vdash_\iota \psi \Rightarrow \exists!y(\varphi) & \text{DdRu2} \\
\forall x(\mathbf{\Delta}(\psi \Rightarrow \exists!y(\varphi)))\,; & & \\
\mathbf{\Delta}([\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi))\,,\forall x(\psi \Rightarrow \exists!y(\varphi)) & \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi) & \text{induction} \\
 & \vdash_\iota \forall x(\psi \Rightarrow \exists!y(\varphi)) & \forall\text{-intro} \\
 & \vdash_\iota \mathbf{\Delta}(\forall x(\psi \Rightarrow \exists!y(\varphi))) & \text{defCons} \\
\mathbf{\Delta}([\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi))\,,\forall x(\psi \Rightarrow \exists!y(\varphi)) & \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi) & \text{CutCtxt} \\
\mathbf{\Delta}([\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi)) & \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi) & \text{Cut} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,;[\![t/x]\!]\psi \Rightarrow \mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi) & \text{toCtxt} \\
 & \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}([\![t/x]\!]\psi)) & \text{Ddef} \\
 & \vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi))) & \text{Ddef} \\
\mathbf{\Delta}([\![t/x]\!]\psi) & \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\psi) & \text{ass} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,; & \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\psi) & \text{toCtxt} \\
\mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \vdash_\iota \mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \text{ass} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,;\mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi))\,,[\![t/x]\!]\psi & \vdash_\iota \mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \text{Weak} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,;\mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \vdash_\iota [\![t/x]\!]\psi \Rightarrow \mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \text{DdRu2} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,;\mathbf{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi) & \text{Cut} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,;\mathbf{\Delta}(\forall y([\![t/x]\!]\varphi)) & \vdash_\iota [\![t/x]\!]\psi \Rightarrow \exists!y([\![t/x]\!]\varphi) & \text{Prop. 42} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,,\mathbf{\Delta}(\forall y([\![t/x]\!]\varphi))\,;[\![t/x]\!]\psi & \vdash_\iota \exists!y([\![t/x]\!]\varphi) & \text{DdRu1} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,;\mathbf{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi & \vdash_\iota \exists!y([\![t/x]\!]\varphi) & \text{fromCtxt} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,\&\,\mathbf{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi & \vdash_\iota \exists!y([\![t/x]\!]\varphi) & \text{fromCtxt}
\end{array}$$

  - x is not a free variable of ψ

    * $\mathbf{\Delta}(t) \not\equiv \top$ We have to show that $\iota y_{\forall y(\mathbf{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi)$ and $\iota y_{\mathbf{\Delta}(\psi)\&\forall y(\mathbf{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi)$ are interchangeable. We already have obtained $\mathbf{\Delta}(\psi)\,\&\,\mathbf{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,\psi \vdash_\iota \exists!y([\![t/x]\!]\varphi)$ Continuing the derivation above, we also obtain the second uniqueness condition:

$$\begin{array}{rlll}
\mathbf{\Delta}(\psi)\,;\mathbf{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,\psi & \vdash_\iota & \exists!y([\![t/x]\!]\varphi) & \text{see above} \\
 & \vdash_\iota & \mathbf{\Delta}(\psi) & \text{defAnt} \\
\mathbf{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,\psi & \vdash_\iota & \exists!y([\![t/x]\!]\varphi) & \text{CutCtxt}
\end{array}$$

    Next, we establish interchangeability:

$$
\begin{array}{rl}
\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi, & \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \iota y_{\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \text{Eq-}\iota \\
\qquad \vdash_\iota \boldsymbol{\Delta}(\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi) & \text{defAnt} \\
\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi & \text{ass} \\
\qquad \vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\
\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi & \text{\&-intro} \\
\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \iota y_{\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \text{Cut}
\end{array}
$$

$$
\begin{array}{rl}
\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi, & \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \iota y_{\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \text{Eq-}\iota \\
\qquad \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi) & \text{defAnt} \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi & \text{ass} \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi & \text{\&-elim} \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \iota y_{\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \text{Cut}
\end{array}
$$

The other two required sequents are easily derived.

* $\boldsymbol{\Delta}(t) \equiv \top$ We have to show that $\iota y_\psi([\![t/x]\!]\varphi)$ and $\iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi)$ are interchangeable. First, we derive the remaining uniqueness condition from the one we already obtained:

$$
\begin{array}{rl}
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\
\psi \vdash_\iota \exists! y([t/x]\varphi) & \text{subst} \\
\psi \vdash_\iota \boldsymbol{\Delta}(\exists! y([t/x]\varphi)) & \text{defCons} \\
\boldsymbol{\Delta}(\exists! y([t/x]\varphi)) \vdash_\iota \boldsymbol{\Delta}(\exists! y([\![t/x]\!]\varphi)) & \text{induction} \\
\psi \vdash_\iota \boldsymbol{\Delta}(\exists! y([\![t/x]\!]\varphi)) & \text{Cut} \\
\psi \vdash_\iota \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) & \text{Prop. 42} \\
\psi \vdash_\iota \psi & \text{ass} \\
\psi \vdash_\iota \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi & \text{\&-intro} \\
\psi \vdash_\iota \boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi & \text{\&-intro} \\
\psi \vdash_\iota \exists! y([\![t/x]\!]\varphi) & \text{Cut}
\end{array}
$$

Now, interchangeability is easy to show:

$$
\begin{array}{rl}
\psi, \boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \iota y_\psi([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \text{Eq-}\iota \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \iota y_\psi([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \text{Cut}
\end{array}
$$

$$\begin{array}{rl}
\psi, \boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \iota y_\psi([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \text{Eq-}\iota \\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi) & \text{defAnt} \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi & \text{ass} \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \psi & \&\text{-elim} \\
\boldsymbol{\Delta}(\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, \psi \vdash_\iota \iota y_\psi([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&\psi}([\![t/x]\!]\varphi) & \text{Cut}
\end{array}$$

The remaining two sequents are easy to derive.

– x is a free variable of ψ

* $\boldsymbol{\Delta}(t) \not\equiv \top$
Both terms are $\iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\varphi)$ and we already derived the uniqueness condition.

* $\boldsymbol{\Delta}(t) \equiv \top$ We have to show that $\iota y_{[\![t/x]\!]\psi}([\![t/x]\!]\varphi)$ and $\iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\varphi)$ are interchangeable. First, we derive the remaining uniqueness condition:

$$\begin{array}{rl}
\boldsymbol{\Delta}([t/x]\psi) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{induction} \\
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt} \\
\psi \vdash_\iota \psi & \text{ass} \\
[t/x]\psi \vdash_\iota [t/x]\psi & \text{subst} \\
\vdash_\iota \boldsymbol{\Delta}([t/x]\psi) & \text{defAnt} \\
\vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{Cut} \\
[t/x]\psi \vdash_\iota \exists!y([t/x]\varphi) & \text{subst} \\
[t/x]\psi \vdash_\iota \boldsymbol{\Delta}(\exists!y([t/x]\varphi)) & \text{defCons} \\
\boldsymbol{\Delta}(\exists!y([t/x]\varphi)) \vdash_\iota \boldsymbol{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \text{induction} \\
[t/x]\psi \vdash_\iota \boldsymbol{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \text{Cut} \\
\boldsymbol{\Delta}([t/x]\psi)\,; [\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{induction} \\
[\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{CutCtxt} \\
[\![t/x]\!]\psi \vdash_\iota \boldsymbol{\Delta}(\exists!y([\![t/x]\!]\varphi)) & \text{Cut} \\
[\![t/x]\!]\psi \vdash_\iota \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) & \text{Prop. 42} \\
[\![t/x]\!]\psi \vdash_\iota [\![t/x]\!]\psi & \text{ass} \\
[\![t/x]\!]\psi \vdash_\iota \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, [\![t/x]\!]\psi & \&\text{-intro} \\
[\![t/x]\!]\psi \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) \,\&\, \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) \,\&\, [\![t/x]\!]\psi & \&\text{-intro} \\
[\![t/x]\!]\psi \vdash_\iota \exists!y([\![t/x]\!]\varphi) & \text{Cut}
\end{array}$$

The four sequents required to show interchangeability are derived analogously to the previous cases.

Next, we give the cases for the second part:

- $\tau \equiv x$ We have to derive $\mathbf{\Delta}(t) \vdash_\iota \mathbf{\Delta}(t)$ and $\mathbf{\Delta}(t) \vdash_\iota t = t$, which is easy
- $\tau \equiv y$ with $y \not\equiv x$ We only have to derive $\mathbf{\Delta}(t) \vdash_\iota y = y$, which is easy.
- $\tau \equiv f(t_1, t_2, \ldots, t_n)$ We have to derive

$$\left\{\begin{array}{r} \mathbf{\Delta}([t/x]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([t/x]t_n) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([\![t/x]\!]t_n) \\ \mathbf{\Delta}(t)\,, \mathbf{\Delta}([\![t/x]\!]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([\![t/x]\!]t_n) \vdash_\iota \mathbf{\Delta}([t/x]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([t/x]t_n) \\ \mathbf{\Delta}([t/x]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([t/x]t_n) \vdash_\iota f([\![t/x]\!]t_1, \ldots, [\![t/x]\!]t_n) \\ = f([t/x]t_1, \ldots, [t/x]t_n) \end{array}\right.$$

  which is easy by applying induction on the terms $t_1, t_2, \ldots, t_n$.
- $\tau \equiv \iota y_\psi(\varphi)$
  - $x \equiv y$ or x is not a free variable of φ
    - x is not a free variable of ψ We have to derive

$$\left\{\begin{array}{r} \psi \vdash_\iota \psi \\ \mathbf{\Delta}(t)\,, \psi \vdash_\iota \psi \\ \psi \vdash_\iota \iota y_\psi(\varphi) = \iota y_\psi(\varphi) \end{array}\right.$$

      which is trivial.
    - x is a free variable of ψ Using the result of the first part, whether $\mathbf{\Delta}(t) \equiv \top$ or not, we have to derive

$$\left\{\begin{array}{r} \mathbf{\Delta}(t) \,\&\, [t/x]\psi \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\psi) \,\&\, [\![t/x]\!]\psi \\ \mathbf{\Delta}(t)\,, \mathbf{\Delta}([\![t/x]\!]\psi) \,\&\, [\![t/x]\!]\psi \vdash_\iota \mathbf{\Delta}(t) \,\&\, [t/x]\psi \\ \mathbf{\Delta}(t) \,\&\, [t/x]\psi \vdash_\iota \iota x_{\mathbf{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) = \iota x_{\mathbf{\Delta}(t)\&[t/x]\psi}(\varphi) \end{array}\right.$$

      For the first sequent, we have

$$\begin{array}{rr} \mathbf{\Delta}([t/x]\psi)\,; [t/x]\psi \vdash_\iota [\![t/x]\!]\psi & \text{induction} \\ \vdash_\iota \mathbf{\Delta}(\psi) & \text{defAnt} \\ \psi \vdash_\iota \psi & \text{ass} \\ \mathbf{\Delta}(t)\,, [t/x]\psi \vdash_\iota [t/x]\psi & \text{subst} \\ \mathbf{\Delta}(t) \vdash_\iota \mathbf{\Delta}([t/x]\psi) & \text{defAnt} \\ \mathbf{\Delta}(t)\,; [t/x]\psi \vdash_\iota [\![t/x]\!]\psi & \text{CutCtxt} \\ \mathbf{\Delta}(t) \,\&\, [t/x]\psi \vdash_\iota [\![t/x]\!]\psi & \text{fromCtxt} \\ \mathbf{\Delta}(t) \,\&\, [t/x]\psi \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\psi) & \text{defCons} \\ \mathbf{\Delta}(t) \,\&\, [t/x]\psi \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\psi) \,\&\, [\![t/x]\!]\psi & \&\text{-intro} \end{array}$$

and for the second sequent,

$$\begin{array}{rl}
\boldsymbol{\Delta}([t/x]\psi)\,;[\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{induction}\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi) \vdash_\iota \boldsymbol{\Delta}([t/x]\psi) & \text{induction}\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi)\,;[\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{CutCtxt}\\
\boldsymbol{\Delta}(t)\,;\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{fromCtxt}\\
\boldsymbol{\Delta}(t)\ \&\ \boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{fromCtxt}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi)\,;[\![t/x]\!]\psi \vdash_\iota [\![t/x]\!]\psi & \text{AssCtxt}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota [\![t/x]\!]\psi & \text{fromCtxt}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi) & \text{defAnt}\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{AnDc}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(t)) & \text{Ddef}\\
\boldsymbol{\Delta}(t) \vdash_\iota \boldsymbol{\Delta}(t) & \text{ass}\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota \boldsymbol{\Delta}(t)\ \&\ [t/x]\psi & \text{\&-intro}
\end{array}$$

The third sequent is derived as follows:

$$\begin{array}{rl}
\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota \exists!y(\varphi) & \text{part 1}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota \exists!y(\varphi) & \text{subst}\\
\boldsymbol{\Delta}(t)\ \&\ [t/x]\psi \vdash_\iota \exists!y(\varphi) & \text{toCtxt}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi, \boldsymbol{\Delta}(t)\ \&\ [t/x]\psi \vdash_\iota \iota x_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) & \\
\qquad = \iota x_{\boldsymbol{\Delta}(t)\&[t/x]\psi}(\varphi) & \text{Eq-}\iota\\
\boldsymbol{\Delta}(t)\ \&\ [t/x]\psi \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi & \text{first sequent}\\
\boldsymbol{\Delta}(t)\ \&\ [t/x]\psi \vdash_\iota \iota x_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) & \\
\qquad = \iota x_{\boldsymbol{\Delta}(t)\&[t/x]\psi}(\varphi) & \text{Cut}
\end{array}$$

– $x \not\equiv y$ and x is a free variable of φ and y is not a free variable of t Using part 1, whether $\boldsymbol{\Delta}(t) \equiv \top$ or not, whether x is a free variable of ψ or not, it suffices to derive

$$\begin{cases}
\boldsymbol{\Delta}(t)\ \&\ [t/x]\psi \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\ \&\ [\![t/x]\!]\psi\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ \forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\ \&\ [\![t/x]\!]\psi \vdash_\iota \boldsymbol{\Delta}(t)\ \&\ [t/x]\psi\\
\boldsymbol{\Delta}(t)\ \&\ [t/x]\psi \vdash_\iota \iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\varphi)\\
\qquad = \iota y_{\boldsymbol{\Delta}(t)\&\psi}([t/x]\varphi)
\end{cases}$$

For the first sequent, we have

$$\begin{array}{rl}
\psi \vdash_\iota \exists y(\varphi) & \text{\&-elim}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota \exists y([t/x]\varphi) & \text{subst}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota \boldsymbol{\Delta}(\forall y([t/x]\varphi)) & \text{defCons}\\
\boldsymbol{\Delta}([t/x]\forall y(\varphi)) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\forall y(\varphi)) & \text{induction}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota \boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi)) & \text{defCons}
\end{array}$$

$$\begin{array}{rl}
\vdash_\iota \boldsymbol{\Delta}(\psi) & \text{defAnt}\\
\psi \vdash_\iota \psi & \text{ass}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota [t/x]\psi & \text{subst}\\
\boldsymbol{\Delta}([t/x]\psi)\,,[t/x]\psi \vdash_\iota [\![t/x]\!]\psi & \text{induction}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota [\![t/x]\!]\psi & \text{Cut3}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{defCons}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi)) & \&\text{-intro}\\
\boldsymbol{\Delta}(t)\,;[t/x]\psi \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi & \&\text{-intro}\\
\boldsymbol{\Delta}(t)\,\&\,[t/x]\psi \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi & \text{fromCtxt}
\end{array}$$

and for the second sequent

$$\begin{array}{rl}
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi) \vdash_\iota \boldsymbol{\Delta}([t/x]\psi) & \text{induction}\\
\boldsymbol{\Delta}([t/x]\psi)\,;[\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{induction}\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi)\,;[\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{CutCtxt}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))) & \text{Ddef}\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi)\,;\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,,[\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{Weak}\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi)\,;\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \&\text{-intro}\\
\boldsymbol{\Delta}(t)\,;\boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi \vdash_\iota [t/x]\psi & \text{fromCtxt}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(t)) & \text{Ddef}\\
\boldsymbol{\Delta}(t) \vdash_\iota \boldsymbol{\Delta}(t) & \text{ass}\\
\boldsymbol{\Delta}(t)\,; \vdash_\iota \boldsymbol{\Delta}(t) & \text{toCtxt}\\
\boldsymbol{\Delta}(t)\,;\boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi \vdash_\iota \boldsymbol{\Delta}(t)\,\&\,[t/x]\psi & \&\text{-intro}\\
\boldsymbol{\Delta}(t)\,\&\,\boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi \vdash_\iota \boldsymbol{\Delta}(t)\,\&\,[t/x]\psi & \text{fromCtxt}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([\![t/x]\!]\psi)) & \text{Ddef}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{ass}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi)\,,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi)) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{Weak}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi) \vdash_\iota \boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi)) \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\psi) & \text{DdRu2}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi) \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))) & \text{Weak}\\
\boldsymbol{\Delta}([\![t/x]\!]\psi) \vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi) & \&\text{-intro}\\
\vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi) \Rightarrow \boldsymbol{\Delta}(\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi) & \text{DdRu2}\\
\vdash_\iota \boldsymbol{\Delta}\big(\boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\, \boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi\big) & \&\text{-intro}\\
\boldsymbol{\Delta}(t)\,,\boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi \vdash_\iota \boldsymbol{\Delta}(t)\,\&\,[t/x]\psi & \text{AnDc}
\end{array}$$

For the third one, part 1 gives us the uniqueness condition for $\iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\varphi)$ and the other one is easy to derive. This enables us to derive

$$\begin{array}{rl}
\boldsymbol{\Delta}(t)\,\&\,[t/x]\psi, \boldsymbol{\Delta}([\![t/x]\!]\psi) & \\
\quad\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi \vdash_\iota \iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(t)\&\psi}([t/x]\varphi) & \text{Eq-}\iota\\
\boldsymbol{\Delta}(t)\,\&\,[t/x]\psi \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\psi)\,\&\,\boldsymbol{\Delta}(\forall y([\![t/x]\!]\varphi))\,\&\,[\![t/x]\!]\psi & \text{first sequent}\\
\boldsymbol{\Delta}(t)\,\&\,[t/x]\psi \vdash_\iota \iota y_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&\forall y(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\varphi) & \\
\qquad = \iota y_{\boldsymbol{\Delta}(t)\&\psi}([t/x]\varphi) & \text{Cut}
\end{array}$$

- $\alpha \equiv p(t_1, t_2, \ldots, t_n)$ We have to derive

$$\begin{cases} \mathbf{\Delta}([t/x]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([t/x]t_n) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]t_1) \,\&\, \mathbf{\Delta}([\![t/x]\!]t_2) \\ \mathbf{\Delta}(t)\,, \mathbf{\Delta}([\![t/x]\!]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([\![t/x]\!]t_n) \vdash_\iota \mathbf{\Delta}([t/x]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([t/x]t_n) \\ \mathbf{\Delta}([\![t/x]\!]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([\![t/x]\!]t_n)\,; \\ \qquad p([\![t/x]\!]t_1, \ldots, [\![t/x]\!]t_n) \vdash_\iota p([t/x]t_1, \ldots, [t/x]t_n) \\ \mathbf{\Delta}([\![t/x]\!]t_1) \,\&\, \cdots \,\&\, \mathbf{\Delta}([\![t/x]\!]t_n)\,; \\ \qquad p([t/x]t_1, \ldots, [t/x]t_n) \vdash_\iota p([\![t/x]\!]t_1, \ldots, [\![t/x]\!]t_n) \end{cases}$$

  Using induction on $t_1, t_2, \ldots, t_n$, the first two sequents are easily derived. The last two are not difficult, too, using ERp2 (and ESy for the last one).

- $\alpha \equiv \beta \,\&\, \gamma$ We have to derive

$$\begin{cases} \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \\ \mathbf{\Delta}(t)\,, \mathbf{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \vdash_\iota \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \\ \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [t/x]\beta \,\&\, [t/x]\gamma \\ \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \end{cases}$$

$$\begin{array}{rr}
\vdash_\iota \mathbf{\Delta}(\mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)) & \text{Ddef} \\
\mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \vdash_\iota \mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) & \text{ass} \\
\mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \vdash_\iota \mathbf{\Delta}([t/x]\beta) & \text{\&-elim} \\
\mathbf{\Delta}([t/x]\beta) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\beta) & \text{induction} \\
\mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\beta) & \text{Cut} \\
\mathbf{\Delta}([t/x]\beta)\,; [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\beta) & \text{toCtxt} \\
\mathbf{\Delta}([t/x]\beta)\,; [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma) \vdash_\iota [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma) & \text{Cons} \\
\mathbf{\Delta}([t/x]\beta)\,, [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma)\,; [t/x]\beta \vdash_\iota \mathbf{\Delta}([t/x]\gamma) & \text{DdRu1} \\
\mathbf{\Delta}([t/x]\gamma) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\gamma) & \text{induction} \\
\mathbf{\Delta}([t/x]\beta)\,, [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma)\,; [t/x]\beta \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\gamma) & \text{Cut} \\
\mathbf{\Delta}([t/x]\beta)\,; [\![t/x]\!]\beta \vdash_\iota [t/x]\beta & \text{induction} \\
\mathbf{\Delta}([t/x]\beta)\,, [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma)\,; [\![t/x]\!]\beta \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\gamma) & \text{Cut} \\
\mathbf{\Delta}([t/x]\beta)\,, [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma)\,; \vdash_\iota [\![t/x]\!]\beta \Rightarrow \mathbf{\Delta}([\![t/x]\!]\gamma) & \text{DdRu2} \\
\mathbf{\Delta}([t/x]\beta)\,; [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma) \vdash_\iota [\![t/x]\!]\beta \Rightarrow \mathbf{\Delta}([\![t/x]\!]\gamma) & \text{fromCtxt} \\
\mathbf{\Delta}([t/x]\beta)\,; [t/x]\beta \Rightarrow \mathbf{\Delta}([t/x]\gamma) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) & \text{\&-intro} \\
\mathbf{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \vdash_\iota \mathbf{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) & \text{fromCtxt}
\end{array}$$

$$\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma)) \quad \text{Ddef}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \quad \text{ass}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\beta) \quad \&\text{-elim}$$
$$\boldsymbol{\Delta}(t)\,, \boldsymbol{\Delta}([\![t/x]\!]\beta) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta) \quad \text{induction}$$
$$\boldsymbol{\Delta}(t)\,, \boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta) \quad \text{Cut}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \vdash_\iota [\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma) \quad \&\text{-elim}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta)\,; [\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma) \vdash_\iota [\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma) \quad \text{toCtxt}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta)\,, [\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma)\,; [\![t/x]\!]\beta \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\gamma) \quad \text{DdRu1}$$
$$\boldsymbol{\Delta}([t/x]\beta)\,; [t/x]\beta \vdash_\iota [\![t/x]\!]\beta \quad \text{induction}$$
$$\boldsymbol{\Delta}(t)\,, \boldsymbol{\Delta}([\![t/x]\!]\beta) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta) \quad \text{induction}$$
$$\boldsymbol{\Delta}(t)\,, \boldsymbol{\Delta}([\![t/x]\!]\beta)\,; [t/x]\beta \vdash_\iota [\![t/x]\!]\beta \quad \text{CutCtxt}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta)\,, [\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma)\,, \boldsymbol{\Delta}(t)\,; [t/x]\beta \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\gamma) \quad \text{Cut}$$
$$\boldsymbol{\Delta}(t)\,, \boldsymbol{\Delta}([\![t/x]\!]\gamma) \vdash_\iota \boldsymbol{\Delta}([t/x]\gamma) \quad \text{induction}$$
$$\boldsymbol{\Delta}(t)\,; \boldsymbol{\Delta}([\![t/x]\!]\gamma) \vdash_\iota \boldsymbol{\Delta}([t/x]\gamma) \quad \text{toCtxt}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta)\,, [\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma)\,, \boldsymbol{\Delta}(t)\,; [t/x]\beta \vdash_\iota \boldsymbol{\Delta}([t/x]\gamma) \quad \text{Cut}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta)\,, [\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma)\,, \boldsymbol{\Delta}(t)\,; \vdash_\iota [t/x]\beta \Rightarrow \boldsymbol{\Delta}([t/x]\gamma) \quad \text{DdRu2}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta)\,, [\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma)\,; \boldsymbol{\Delta}(t) \vdash_\iota [t/x]\beta \Rightarrow \boldsymbol{\Delta}([t/x]\gamma) \quad \text{fromCtxt}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta)\,; ([\![t/x]\!]\beta \Rightarrow \boldsymbol{\Delta}([\![t/x]\!]\gamma)) \,\&\, \boldsymbol{\Delta}(t) \vdash_\iota [t/x]\beta \Rightarrow \boldsymbol{\Delta}([t/x]\gamma) \quad \text{fromCtxt}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \,\&\, \boldsymbol{\Delta}(t) \vdash_\iota [t/x]\beta \Rightarrow \boldsymbol{\Delta}([t/x]\gamma) \quad \text{fromCtxt}$$
$$\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(t)) \quad \text{Ddef}$$
$$\boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma)\,, \boldsymbol{\Delta}(t) \vdash_\iota [t/x]\beta \Rightarrow \boldsymbol{\Delta}([t/x]\gamma) \quad \text{AnDc}$$
$$\boldsymbol{\Delta}(t)\,, \boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \quad \&\text{-intro}$$

$$\boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \quad \text{AssCtxt}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma) \quad \text{first sequent}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \quad \text{CutCtxt}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [\![t/x]\!]\beta \quad \&\text{-elim}$$
$$\boldsymbol{\Delta}([t/x]\beta)\,; [\![t/x]\!]\beta \vdash_\iota [t/x]\beta \quad \text{induction}$$
$$\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)) \quad \text{Ddef}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \quad \text{ass}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta) \quad \&\text{-elim}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \vdash_\iota [t/x]\beta \quad \text{CutCtxt}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [t/x]\beta \quad \text{Cut}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota \boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma) \quad \text{ConsCtxt}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [t/x]\beta \Rightarrow \boldsymbol{\Delta}([t/x]\gamma) \quad \&\text{-elim}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota \boldsymbol{\Delta}([t/x]\gamma) \quad \text{MP}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [\![t/x]\!]\gamma \quad \&\text{-elim}$$
$$\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,; [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota \boldsymbol{\Delta}([t/x]\gamma) \,\&\, [\![t/x]\!]\gamma \quad \&\text{-intro}$$
$$\boldsymbol{\Delta}([t/x]\gamma)\,; [\![t/x]\!]\gamma \vdash_\iota [t/x]\gamma \quad \text{induction}$$
$$\boldsymbol{\Delta}([t/x]\gamma) \,\&\, [\![t/x]\!]\gamma \vdash_\iota [t/x]\gamma \quad \text{fromCtxt}$$

$$\begin{array}{rr}
\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\, [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [t/x]\gamma & \text{Cut}\\
\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\, [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma \vdash_\iota [t/x]\gamma \,\&\, [t/x]\gamma & \&\text{-intro}
\end{array}$$

$$\begin{array}{rr}
\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\, [t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [t/x]\beta \,\&\, [t/x]\gamma & \text{AssCtxt}\\
\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\, [t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [t/x]\beta & \&\text{-elim}\\
\boldsymbol{\Delta}([t/x]\beta)\,;\, [t/x]\beta \vdash_\iota [\![t/x]\!]\beta & \text{induction}\\
\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\, [t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [\![t/x]\!]\beta & \text{Cut3}\\
\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\, [t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [t/x]\gamma & \&\text{-elim}\\
\boldsymbol{\Delta}([t/x]\gamma)\,;\, [t/x]\gamma \vdash_\iota [\![t/x]\!]\gamma & \text{induction}\\
\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\, [t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [\![t/x]\!]\gamma & \text{Cut3}\\
\boldsymbol{\Delta}([t/x]\beta \,\&\, [t/x]\gamma)\,;\, [t/x]\beta \,\&\, [t/x]\gamma \vdash_\iota [\![t/x]\!]\beta \,\&\, [\![t/x]\!]\gamma & \&\text{-intro}
\end{array}$$

- $\alpha \equiv \neg\beta$ We immediately obtain the first two sequents by induction and the last two are not difficult either:

$$\begin{array}{rr}
\boldsymbol{\Delta}([t/x]\beta)\,;\, [t/x]\beta \vdash_\iota [\![t/x]\!]\beta & \text{induction}\\
\boldsymbol{\Delta}([t/x]\beta) \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\beta) & \text{induction}\\
\boldsymbol{\Delta}([t/x]\beta)\,;\, \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\beta) & \text{toCtxt}\\
\boldsymbol{\Delta}([t/x]\beta)\,;\, \neg\,[\![t/x]\!]\beta \vdash_\iota \neg\,[t/x]\beta & \text{CoPo1}
\end{array}$$

$$\begin{array}{rr}
\boldsymbol{\Delta}([t/x]\beta)\,;\, [\![t/x]\!]\beta \vdash_\iota [t/x]\beta & \text{induction}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([t/x]\beta)) & \text{Ddef}\\
\boldsymbol{\Delta}([t/x]\beta) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta) & \text{ass}\\
\boldsymbol{\Delta}([t/x]\beta)\,;\, \vdash_\iota \boldsymbol{\Delta}([t/x]\beta) & \text{toCtxt}\\
\boldsymbol{\Delta}([t/x]\beta)\,;\, \neg\,[t/x]\beta \vdash_\iota \neg\,[\![t/x]\!]\beta & \text{CoPo1}
\end{array}$$

- $\alpha \equiv \forall y(\beta)$ with $x \equiv y$ or x not a free variable of β In this case, $[t/x]\alpha \equiv \forall x(\beta) \equiv [\![t/x]\!]\alpha$ and the required sequents are easy to derive.

- $\alpha \equiv \forall y(\beta)$ with $x \not\equiv y$ and x a free variable of β The first two required sequents are obtained by applying induction on β and invoking the SimGen rule (remember that for the substitution to be defined, y must not be a free variable of t).

  The third sequent is derived as follows:

$$\begin{array}{rr}
\boldsymbol{\Delta}([t/x]\beta)\,;\, [\![t/x]\!]\beta \vdash_\iota [t/x]\beta & \text{induction}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}([t/x]\beta)) & \text{Ddef}\\
\vdash_\iota \forall y(\boldsymbol{\Delta}(\boldsymbol{\Delta}([t/x]\beta))) & \forall\text{-intro}\\
\forall y(\boldsymbol{\Delta}([t/x]\beta)) \vdash_\iota \forall y(\boldsymbol{\Delta}([t/x]\beta)) & \text{ass}\\
\forall y(\boldsymbol{\Delta}([t/x]\beta)) \vdash_\iota \boldsymbol{\Delta}([t/x]\beta) & \forall\text{-elim}\\
\forall y(\boldsymbol{\Delta}([t/x]\beta));\, [\![t/x]\!]\beta \vdash_\iota [t/x]\beta & \text{CutCtxt}\\
\forall y(\boldsymbol{\Delta}([t/x]\beta));\, \forall y([\![t/x]\!]\beta) \vdash_\iota \forall y([t/x]\beta) & \text{SimGen}
\end{array}$$

  and the last one likewise.

Finally, we prove the third part. We proceed analogously as in theorem 51 and first consider the case that $t \equiv \iota y_\Psi(\Phi)$ and y is not a free variable of Ψ. We again define the term

$$t' \equiv \iota y((\Psi \Rightarrow \Phi) \,\&\, (\neg\Psi \Rightarrow y = a))$$

with a a variable symbol different from y. Lemma 50 yields its uniqueness condition.

The proof is analogous to theorem 51, but in this case we will derive for each formula α of the PITFOL calculus for which the uniqueness conditions are derivable,

$$\begin{cases} \mathbf{\Delta}([\![t/x]\!]\,\alpha)\,;[t'/x]\,\alpha & \vdash_\iota & [\![t/x]\!]\,\alpha \\ \mathbf{\Delta}([\![t/x]\!]\,\alpha)\,;[\![t/x]\!]\,\alpha & \vdash_\iota & [t'/x]\,\alpha \end{cases}$$

if the substitutions are defined, and for each term τ of the PITFOL calculus for which the uniqueness conditions are derivable,

$$\mathbf{\Delta}([\![t/x]\!]\,\tau) \vdash_\iota [t'/x]\,\tau = [\![t/x]\!]\,\tau$$

if the substitutions are defined.

Most of the cases are identical to theorem 51; we only need to mention

- $\tau \equiv \iota z_\psi(\varphi)$ with $x \equiv z$ or x not a free variable of φ Using part 1, what we have to derive is

  $$\mathbf{\Delta}([\![t/x]\!]\,\psi) \,\&\, [\![t/x]\!]\,\psi \vdash_\iota \iota z_\psi(\varphi) = \iota z_{\mathbf{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\Psi}(\varphi)$$

  when x is not a free variable of ψ or

  $$\mathbf{\Delta}([\![t/x]\!]\,\psi) \,\&\, [\![t/x]\!]\,\psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\psi}(\varphi) = \iota z_{\mathbf{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi)$$

  when x is a free variable of ψ.

  We will only handle the last case explicitly; the case where x is not free in ψ is even easier.

  We have the uniqueness condition of τ at our disposal, from which we get

  $$\begin{array}{rcll} \forall x(x = x);[t'/x]\,\psi & \vdash_\iota & \exists!z(\varphi) & \text{subst} \\ \forall x(x = x) \,\&\, [t'/x]\,\psi & \vdash_\iota & \exists!z(\varphi) & \text{fromCtxt} \end{array}$$

  and part 1 yields the uniqueness condition of $\iota z_{\mathbf{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi)$.

  Combining these sequents, we finally get

$$
\begin{array}{rl}
\forall x(x=x)\ \&\ [t'/x]\psi, & \\
\quad \boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\psi}(\varphi) = \iota z_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) & \text{Eq-}\iota \\
\quad \boldsymbol{\Delta}([\![t/x]\!]\psi)\,;[\![t/x]\!]\psi \vdash_\iota [t'/x]\psi & \text{induction} \\
\quad \boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota [t'/x]\psi & \text{fromCtxt} \\
\quad \vdash_\iota x=x & \text{eq} \\
\quad \vdash_\iota \forall x(x=x) & \forall\text{-intro} \\
\quad \boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota \forall x(x=x)\ \&\ [t'/x]\psi & \&\text{-intro} \\
\quad \boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ [\![t/x]\!]\psi \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\psi}(\varphi) = \iota z_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) & \text{Cut}
\end{array}
$$

- $\tau \equiv \iota z_\psi(\varphi)$ with $x \not\equiv z$ and x a free variable of φ Again invoking part 1, it is sufficient to derive

$$
\begin{array}{l}
\boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ \forall z(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\ \&\ [\![t/x]\!]\psi \\
\quad \vdash_\iota \iota z_{\forall x(x=x)\&[t'/x]\psi}([t'/x]\varphi) = \iota z_{\boldsymbol{\Delta}([\![t/x]\!]\psi)\&\forall z(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\varphi)
\end{array}
$$

From the uniqueness condition of τ, we derive the first uniqueness condition:

$$
\begin{array}{rcll}
\forall x(x=x); [t'/x]\psi & \vdash_\iota & \exists! z([t'/x]\varphi) & \text{subst} \\
\forall x(x=x)\ \&\ [t'/x]\psi & \vdash_\iota & \exists! z([t'/x]\varphi) & \text{fromCtxt}
\end{array}
$$

and again, part 1 gives us the second uniqueness condition.

As in the previous case, we finally combine both uniqueness conditions. To ease the notation, we will abbreviate $\psi_1 \equiv \forall x(x=x)\ \&\ [t'/x]\psi$ and $\psi_2 \equiv \boldsymbol{\Delta}([\![t/x]\!]\psi)\ \&\ \forall z(\boldsymbol{\Delta}([\![t/x]\!]\varphi))\ \&\ [\![t/x]\!]\psi$.

$$
\begin{array}{rl}
\boldsymbol{\Delta}([\![t/x]\!]\varphi)\,;[t'/x]\varphi \vdash_\iota [\![t/x]\!]\varphi & \text{induction} \\
\boldsymbol{\Delta}([\![t/x]\!]\varphi)\,;\vdash_\iota [t'/x]\varphi \Rightarrow [\![t/x]\!]\varphi & \text{DdRu2} \\
\boldsymbol{\Delta}([\![t/x]\!]\varphi)\,;[\![t/x]\!]\varphi \vdash_\iota [t'/x]\varphi & \text{induction} \\
\boldsymbol{\Delta}([\![t/x]\!]\varphi)\,;\vdash_\iota [\![t/x]\!]\varphi \Rightarrow [t'/x]\varphi & \text{DdRu2} \\
\boldsymbol{\Delta}([\![t/x]\!]\varphi)\,;\vdash_\iota [t'/x]\varphi \Leftrightarrow [\![t/x]\!]\varphi & \&\text{-intro} \\
\boldsymbol{\Delta}([\![t/x]\!]\varphi) \vdash_\iota [t'/x]\varphi \Leftrightarrow [\![t/x]\!]\varphi & \text{fromCtxt} \\
\vdash_\iota \boldsymbol{\Delta}(\psi_2) & \text{defAnt} \\
\psi_2 \vdash_\iota \psi_2 & \text{ass} \\
\psi_2 \vdash_\iota \forall z(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) & \&\text{-elim} \\
[w/z]\psi_2 \vdash_\iota \forall z(\boldsymbol{\Delta}([\![t/x]\!]\varphi)) & \text{subst} \\
[w/z]\psi_2 \vdash_\iota \boldsymbol{\Delta}([\![t/x]\!]\varphi) & \forall\text{-elim} \\
[w/z]\psi_2 \vdash_\iota [t'/x]\varphi \Leftrightarrow [\![t/x]\!]\varphi & \text{Cut} \\
[w/z]\psi_2 \vdash_\iota \forall z([t'/x]\varphi \Leftrightarrow [\![t/x]\!]\varphi) & \forall\text{-intro} \\
\vdash_\iota \boldsymbol{\Delta}(\psi_1) & \text{defAnt} \\
\vdash_\iota [w/z]\boldsymbol{\Delta}(\psi_1) & \text{subst}
\end{array}
$$

$$
\begin{array}{rl}
[w/z]\psi_1, [w/z]\psi_2 \vdash_\iota \forall z([t'/x]\varphi \Leftrightarrow [\![t/x]\!]\varphi) & \text{Weak} \\
[w/z]\psi_1, [w/z]\psi_2 \vdash_\iota \iota z_{[w/z]\psi_1}([t'/x]\varphi) = \iota z_{[w/z]\psi_2}([\![t/x]\!]\varphi) & \text{Eq-}\iota \\
\psi_1, \psi_2 \vdash_\iota \iota z_{\psi_1}([t'/x]\varphi) = \iota z_{\psi_2}([\![t/x]\!]\varphi) & \text{subst} \\
\psi_2 \vdash_\iota \mathbf{\Delta}([t/x]\psi) & \&\text{-elim} \\
\mathbf{\Delta}([\![t/x]\!]\psi)\,; [\![t/x]\!]\psi \vdash_\iota [t'/x]\psi & \text{induction} \\
\psi_2 \vdash_\iota [t/x]\psi & \text{Cut3} \\
\vdash_\iota x = x & \text{eq} \\
\vdash_\iota \forall x(x = x) & \forall\text{-intro} \\
\psi_2 \vdash_\iota \forall x(x = x)\ \&\ [t'/x]\psi & \&\text{-intro} \\
\psi_2 \vdash_\iota \iota z_{\psi_1}([t'/x]\varphi) = \iota z_{\psi_2}([\![t/x]\!]\varphi) & \text{Cut}
\end{array}
$$

where w is a variable symbol different from z and not occurring in any of ψ_1, ψ_2, φ, t and t'.

□

We can now derive a substitution rule analogous to Subst2: With the same definition of t' as in theorem 51, we get

$$
\begin{array}{rl}
\boxed{\text{RefSubst}} & \\
UC(t) & \text{prem} \\
\sigma_1, \sigma_2, \ldots, \sigma_n; \gamma_1, \gamma_2, \ldots, \gamma_m \vdash_\iota \alpha & \text{prem} \\
[t'/x]\sigma_1, [t'/x]\sigma_2, \ldots, [t'/x]\sigma_n; [t'/x]\gamma_1, [t'/x]\gamma_2, \ldots, [t'/x]\gamma_m \vdash_\iota [t'/x]\alpha & \text{subst} \\
\mathbf{\Delta}([\![t/x]\!]\gamma_m)\,; [\![t/x]\!]\gamma_m \vdash_\iota [t'/x]\gamma_m & \text{Th. 52} \\
\mathbf{\Delta}([\![t/x]\!]\gamma_m), [t'/x]\sigma_1, \ldots, [t'/x]\sigma_n; & \\
\quad [t'/x]\gamma_1, [t'/x]\gamma_2, \ldots, [\![t/x]\!]\gamma_m \vdash_\iota [t'/x]\alpha & \text{Cut} \\
\vdots & \\
\mathbf{\Delta}([\![t/x]\!]\alpha)\,; [t'/x]\alpha \vdash_\iota [\![t/x]\!]\alpha & \text{Th. 52} \\
\mathbf{\Delta}([\![t/x]\!]\sigma_1), \ldots \mathbf{\Delta}([\![t/x]\!]\sigma_n), & \\
\mathbf{\Delta}([\![t/x]\!]\gamma_1), \ldots, \mathbf{\Delta}([\![t/x]\!]\gamma_m), & \\
[\![t/x]\!]\sigma_1, \ldots, [\![t/x]\!]\sigma_n, \mathbf{\Delta}([\![t/x]\!]\alpha)\,; [\![t/x]\!]\gamma_1, \ldots, [\![t/x]\!]\gamma_m \vdash_\iota [\![t/x]\!]\alpha & \text{CutCtxt}
\end{array}
$$

Chapter 5

Defined symbols using simultaneous substitution

5.1 Simultaneous substitution

Given a formula α, terms $t_1, \ldots, t_n$ and variable symbols $x_1, \ldots, x_n$ of the PITFOL calculus, where all x_i are different, we introduce the **simultaneous substitution** $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ of α, and likewise for a term τ. It is defined as follows when $n > 0$:

- $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] x \equiv x$ if x is not one of $x_1, x_2, \ldots, x_n$.
- $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] x_i \equiv t_i$
- $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] f(\tau_1, \tau_2, \ldots, \tau_m) \equiv f\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_1, \ldots, \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_m\right)$
- $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \iota x_\psi(\varphi) \equiv \iota x_{\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \& \forall x \left(\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \& \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)}$ if x is not one of $x_1, x_2, \ldots, x_n$. However, the substitution is not defined when there exists a j such that both x_j is a free variable of φ and x is a free variable of t_j; in that case, we say that the substitution would **capture** the free variable x of t_j.
- $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \iota x_{i\psi}(\varphi) \equiv$

  $$\iota x_i \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \& \forall x_i \left(\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & \ldots & x_{i-1} & x_{i+1} & \ldots & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \& \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & \ldots & x_{i-1} & x_{i+1} & \ldots & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)$$

  However, the substitution is not defined when there exists a j different from i such that both x_j is a free variable of φ and x_i is a free variable

of t_j; in that case, we say that the substitution would **capture** the free variable x_i of t_j.

Note that if $n = 1$, this reduces to

$$\left[\!\!\left[\begin{matrix} t_1 \\ x_1 \end{matrix} \right]\!\!\right] \iota x_1 {}_\psi(\varphi) \equiv \iota x_1 {}_{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix} \right]\!\!\right]\psi\right)\&\forall x_1(\mathbf{\Delta}(\varphi))\&\left[\!\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix} \right]\!\!\right]\psi}(\varphi)$$

since we will define $[\![]\!]\alpha \equiv \alpha$.

- $\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \neg\alpha \equiv \neg \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \alpha$
- $\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] (\alpha \,\&\, \beta) \equiv \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \alpha \,\&\, \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \beta$
- $\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] p(\tau_1, \tau_2, \ldots, \tau_m) \equiv p(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \tau_1, \ldots, \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \tau_m)$
- $\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \forall x(\alpha) \equiv \forall x \left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \alpha \right)$ if x is not one of $x_1, x_2, \ldots, x_n$. However, the substitution is not defined when there exists a j such that both x_j is a free variable of α and x is a free variable of t_j; in that case, we say that the substitution would **capture** the free variable x of t_j.
- $\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \forall x_i(\alpha) \equiv \forall x_i \left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & \cdots & x_{i-1} & x_{i+1} & \cdots & x_n \end{matrix} \right]\!\!\right] \alpha \right)$. However, the substitution is not defined when there exists a j different from i such that both x_j is a free variable of α and x_i is a free variable of t_j; in that case, we say that the substitution would **capture** the free variable x_i of t_j.

If $n = 0$ then we define $[\![]\!]\alpha \equiv \alpha$.

For convenience, we will again set $\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \top \equiv \top$.

Property 53 *For any formula α and terms $t_1, \ldots, t_n$ of the* PITFOL *calculus, $\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{matrix} \right]\!\!\right] \alpha$ does not contain ι-terms if and only if α does not contain ι-terms and for each $i = 1, \ldots, n$ either x_i is not free in α or t_i does not contain ι-terms.*

The analogous theorem for terms τ also holds.

Proof.

Easy by structural induction on α and τ. □

Property 54 *For any formula α and terms $t_1, \ldots, t_n$ of the* PITFOL *calculus, x is free in $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ if and only if either x is free in α and x is not one of the x_i's, or for at least one i, x_i is free in α and x is free in t_i.*

The analogous theorem for terms τ also holds.

Proof.

Easy by structural induction on α and τ. □

Property 55 *If the term t_1 is interchangeable with the term s_1, t_2 is interchangeable with s_2, ... and the uniqueness conditions of at least one of $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ and $\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ are derivable, then $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ is interchangeable with $\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$, if both of these simultaneous substitutions are defined.*

The analogous property for terms τ of the PITFOL *calculus also holds.*

Proof.

We prove this by structural induction.

- $\tau \equiv x_i$ We have to show that t_i and s_i are interchangeable, which we already assumed in the statement of the property.
- $\tau \equiv x$ where $x \notin \{x_1, x_2, \ldots, x_n\}$ Both terms are x.
- $\tau \equiv f(\tau_1, \tau_2, \ldots, \tau_m)$ Easy using induction on the τ_i and the ERf2 rule.
- $\tau \equiv \iota x_\psi(\varphi)$ Suppose $x \notin \{x_1, x_2, \ldots, x_n\}$; the other case is similar. We have to show the interchangeability of $\iota x_{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \& \forall x \left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \& \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi} \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)$ and $\iota x_{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \& \forall x \left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \& \left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi} \left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)$ which is easy using induction on ψ and φ and theorem 25.
- The other cases are straightforward.

□

Lemma 56 *For all terms t_1, t_2,... and for each formula α of the* PITFOL *calculus for which the uniqueness conditions are derivable, if the substitution $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ is defined, then*

$$\forall x_1 \forall x_2 \ldots \forall x_n(\mathbf{\Delta}(\alpha)); \mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha\right), \forall x_1 \forall x_2 \ldots \forall x_n(\alpha) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$$

Proof.
• First, suppose all t_i are of the form $\iota y_{i_{\psi_i}}(\varphi_i)$ and y_i is not a free variable of ψ_i. We define the terms t'_1, t'_2,... as before:

$$t'_i \equiv \iota y_i((\psi_i \Rightarrow \varphi_i) \,\&\, (\neg\psi_i \Rightarrow y_i = a))$$

where a is a variable symbol different from all the y_i; lemma 50 yields their uniqueness conditions.

We will shortly prove that

$$\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha\right); [t'_1/z_1]\dots[t'_n/z_n][z_1/x_1]\dots[z_n/x_n]\alpha \vdash_\iota \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha$$

where the z_i are all different, are different from the x_i and do not occur in the t_i and do not occur in α. It is easy to see that these substitutions are always defined and that the formula $[t'_1/z_1]\dots[t'_n/z_n][z_1/x_1]\dots[z_n/x_n]\alpha$ does not depend on the exact choice of the z_i; we will denote this formula as $\left[\begin{smallmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{smallmatrix}\right]\alpha$.

With this sequent at our disposal, we then can easily handle this case:

$$\begin{array}{rl}
\forall x_1\dots\forall x_n(\boldsymbol{\Delta}(\alpha)); \forall x_1\dots\forall x_n(\alpha) \vdash_\iota \forall x_1\dots\forall x_n(\alpha) & \text{assCtxt} \\
\forall x_1\dots\forall x_n(\boldsymbol{\Delta}(\alpha)); \forall x_1\dots\forall x_n(\alpha) \vdash_\iota \alpha & \forall\text{-elim*} \\
\forall x(x=x), \forall x_1\dots\forall x_n(\boldsymbol{\Delta}(\alpha)); \forall x_1\dots\forall x_n(\alpha) \vdash_\iota \left[\begin{smallmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{smallmatrix}\right]\alpha & \text{subst*} \\
\vdash_\iota x = x & \text{eq} \\
\vdash_\iota \forall x(x=x) & \forall\text{-intro} \\
\forall x_1\dots\forall x_n(\boldsymbol{\Delta}(\alpha)); \forall x_1\dots\forall x_n(\alpha) \vdash_\iota \left[\begin{smallmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{smallmatrix}\right]\alpha & \text{CutCtxt} \\
\forall x_1\dots\forall x_n(\boldsymbol{\Delta}(\alpha)), \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha\right); & \\
\qquad \forall x_1\dots\forall x_n(\alpha) \vdash_\iota \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha & \text{Cut} \\
\forall x_1\dots\forall x_n(\boldsymbol{\Delta}(\alpha)); & \\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha\right), \forall x_1\dots\forall x_n(\alpha) \vdash_\iota \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha & \text{FromCtxt2}
\end{array}$$

To derive the missing sequent, we will prove that for each formula α for which the uniqueness conditions are derivable,

$$\begin{cases}
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha\right); \left[\begin{matrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{matrix}\right]\alpha \vdash_\iota \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha \\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha\right); \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\alpha \vdash_\iota \left[\begin{matrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{matrix}\right]\alpha
\end{cases}$$

provided the substitutions are defined and for each term τ for which the uniqueness conditions are derivable,

$$\mathbf{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\tau\right) \vdash_\iota \begin{bmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{bmatrix}\tau = \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\tau$$

provided the substitutions are defined.

We prove this by induction on the number of ι symbols in α and τ. This in turn we prove this by structural induction on α and τ:

- $\tau \equiv x_i$ We have to derive $\mathbf{\Delta}(t_i) \vdash_\iota t'_i = t_i$, which we already did in the proof of theorem 51.

- $\tau \equiv x$ where $x \notin \{x_1, x_2, \ldots, x_n\}$ We have to derive $\vdash_\iota x = x$, which is trivial.

- $\tau \equiv f(\tau_1, \tau_2, \ldots, \tau_m)$ We have to derive

$$\begin{aligned}\mathbf{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\tau_1\right) \,\&\, \cdots \,\&\, \mathbf{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\tau_m\right) \\ \vdash_\iota f\left(\begin{bmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{bmatrix}\tau_1, \ldots, \begin{bmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{bmatrix}\tau_m\right) \\ = f\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\tau_1, \ldots, \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right]\tau_m\right)\end{aligned}$$

  which is not difficult using induction on the τ_i and the ERf2 rule.

- $\tau \equiv \iota y_\psi(\varphi)$ We will assume that $y \notin \{x_1, x_2, \ldots, x_n\}$ (the other case is analogous).

  It is easy to see that

$$\begin{bmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{bmatrix}\iota x_\psi(\varphi) \equiv \iota x_{\mathbf{\Delta}(t'_1)\&\cdots\&\mathbf{\Delta}(t'_n)\&\left[\begin{smallmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{smallmatrix}\right]\psi}\left(\begin{bmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{bmatrix}\varphi\right)$$

  where some $\mathbf{\Delta}(t'_i)$ may be missing (in case x_i is not free in both φ and ψ). All the $\mathbf{\Delta}(t'_i)$ are the validity $\forall x(x = x)$, so it is easily proved that $\begin{bmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{bmatrix}\iota x_\psi(\varphi)$ is interchangeable with $\iota x_{\left[\begin{smallmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{smallmatrix}\right]\psi}\left(\begin{bmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{bmatrix}\varphi\right)$.

  Next, we derive the uniqueness conditions for both ι-terms:

$$
\begin{array}{rl}
\psi \vdash_\iota \exists! x(\varphi) & \text{UC}\\
\vdash_\iota \psi \Rightarrow \exists! x(\varphi) & \text{DdRu2}\\
\forall x_1 \ldots \forall x_n(\boldsymbol{\Delta}(\psi \Rightarrow \exists! x(\varphi)));\ \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi))\right), & \\
\quad \forall x_1 \ldots \forall x_n(\psi \Rightarrow \exists! x(\varphi)) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi)) & \text{induction}\\
\quad \vdash_\iota \forall x_n(\psi \Rightarrow \exists! x(\varphi)) & \forall\text{-intro}\\
\quad \vdots & \\
\quad \vdash_\iota \forall x_1 \ldots \forall x_n(\psi \Rightarrow \exists! x(\varphi)) & \forall\text{-intro}\\
\quad \vdash_\iota \forall x_1 \ldots \forall x_n(\boldsymbol{\Delta}(\psi \Rightarrow \exists! x(\varphi))) & \text{defCons}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi))\right), & \\
\quad \forall x_1 \ldots \forall x_n(\psi \Rightarrow \exists! x(\varphi)) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi)) & \text{CutCtxt}\\
\quad \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi))\right) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi)) & \text{Cut}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right); & \\
\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi \Rightarrow \boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi)) & \text{toCtxt}\\
\quad \vdash_\iota \boldsymbol{\Delta}\left(\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right)\right) & \text{Ddef}\\
\quad \vdash_\iota \boldsymbol{\Delta}\left(\boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right)\right) & \text{Ddef}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right) \vdash_\iota \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right) & \text{ass}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right); \vdash_\iota \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right) & \text{toCtxt}\\
\boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right) \vdash_\iota \boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right) & \text{ass}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right); & \\
\boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right), \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi \vdash_\iota \boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right) & \text{Weak}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right); \boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi & \\
\quad \Rightarrow \boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right) & \text{DdRu2}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right); \boldsymbol{\Delta}\left(\exists! x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi)) & \text{Cut}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right); \boldsymbol{\Delta}\left(\forall x\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\varphi\right)\right) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right](\psi \Rightarrow \exists! x(\varphi)) & \text{Prop. 42}\\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\psi\right), &
\end{array}
$$

$$\boldsymbol{\Delta}\Big(\forall x\,\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big)\Big);\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\psi \vdash_\iota \exists!x\,\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big) \qquad \text{DdRu1}$$

Applying fromCtxt twice yields the first uniqueness condition.

The second one is derived as follows:

$$\begin{array}{rl}
\psi \vdash_\iota \exists!x(\varphi) & \text{UC}\\
[z_n/x_n]\psi \vdash_\iota \exists!x([z_n/x_n]\varphi) & \text{subst}\\
\vdots & \\
[z_1/x_1]\dots[z_n/x_n]\psi \vdash_\iota \exists!x([z_1/x_1]\dots[z_n/x_n]\varphi) & \text{subst}\\
[t_n/z_n][z_1/x_1]\dots[z_n/x_n]\psi \vdash_\iota \exists!x([t'_n/z_n][z_1/x_1]\dots[z_n/x_n]\varphi) & \text{subst}\\
\vdots & \\
[t'_1/z_1]\dots[t'_n/z_n][z_1/x_1]\dots[z_n/x_n]\psi \vdash_\iota \exists!x([t_1/z_1]\dots[t_n/z_n][z_1/x_1]\dots[z_n/x_n]\varphi) & \text{subst}
\end{array}$$

We can use these uniqueness conditions to derive the required sequent. We will abbreviate $\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\psi\Big)$ & $\boldsymbol{\Delta}\Big(\forall x\,\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big)\Big)$ & $\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\psi$ as Ψ.

$$\begin{array}{rl}
\Psi \vdash_\iota \exists x\,\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big) & \forall\text{-elim}\\
\Psi \vdash_\iota \forall x\,\Big(\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big)\Big) & \text{defCons}\\
[z/x]\Psi \vdash_\iota \forall x\,\Big(\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big)\Big) & \text{subst}\\
[z/x]\Psi \vdash_\iota \boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big) & \forall\text{-elim}\\
\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big);\Big[\!\Big[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi \vdash_\iota \Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi & \text{induction}\\
\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big);\vdash_\iota \Big[\!\Big[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi \Rightarrow \Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi & \text{DdRu2}\\
\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big);\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi \vdash_\iota \Big[\!\Big[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi & \text{induction}\\
\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big);\vdash_\iota \Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi \Rightarrow \Big[\!\Big[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi & \text{DdRu2}\\
\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big);\vdash_\iota \Big[\!\Big[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi \Leftrightarrow \Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi & \text{\&-intro}\\
\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\Big) \vdash_\iota \Big[\!\Big[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi \Leftrightarrow \Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi & \text{fromCtxt}\\
[z/x]\Psi \vdash_\iota \Big[\!\Big[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi \Leftrightarrow \Big[\!\Big[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi & \text{Cut}
\end{array}$$

$$
\begin{array}{rl}
[z/x]\,\Psi \vdash_\iota \forall x \left(\begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \varphi \Leftrightarrow \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \varphi \right) & \forall\text{-intro} \\
\Psi \vdash_\iota \forall x \left(\begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \varphi \Leftrightarrow \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \varphi \right) & \text{subst} \\
\vdash_\iota \boldsymbol{\Delta}\left(\begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \psi \right) & \text{defAnt} \\
\begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \psi, \Psi \vdash_\iota \forall x \left(\begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \varphi \Leftrightarrow \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \varphi \right) & \text{Weak} \\
\begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \psi, \Psi \vdash_\iota \iota x_{\left[\begin{smallmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{smallmatrix}\right]\psi} \left(\begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \varphi \right) & \\
\qquad = \iota x_\Psi \left(\left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \varphi \right) & \text{Eq-}\iota \\
\vdash_\iota \boldsymbol{\Delta}(\Psi) & \text{defAnt} \\
\Psi \vdash_\iota \Psi & \text{ass} \\
\Psi \vdash_\iota \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \psi & \&\text{-elim} \\
\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \psi \right); \left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \psi \vdash_\iota \begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \psi & \text{induction} \\
\Psi \vdash_\iota \begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} & \text{Cut3} \\
\Psi \vdash_\iota \iota x_{\left[\begin{smallmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{smallmatrix}\right]\psi} \left(\begin{bmatrix} t'_1 & \dots & t'_n \\ x_1 & & x_n \end{bmatrix} \varphi \right) & \\
\qquad = \iota x_\Psi \left(\left[\!\!\left[\begin{matrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \varphi \right) & \text{Cut}
\end{array}
$$

where we choose z different from x and not occurring in φ, t_1, t'_1, …, t_n and t'_n.

- The other cases are not difficult.

• If not all t_i are ι-terms, then we can proceed as follows. Choose y such that it does not occur in any t_i. If $t_i \equiv \iota y_{i\psi_i}(\varphi_i)$ and y_i is a free variable of ψ_i, then t_i is interchangeable with $\iota y_\psi([y/y_i]\varphi)$. If t_i is not a ι-term, then it is interchangeable with $\iota y_{\boldsymbol{\Delta}(t_i)}(y = t_i)$. Hence, every t_i is interchangeable with a suitable ι-term for the previous case; using property 55 we then obtain the desired sequent. □

Corollary 57 *If the uniqueness conditions of t_1, t_2,... and α are derivable, and the substitution $\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ is defined, then the uniqueness conditions of $\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ are also derivable.*

Proof.

We can invoke the previous lemma; applying the UC rule yields the desired uniqueness conditions. □

The following lemma is analogous to theorem 33:

Lemma 58 *If the uniqueness conditions of t_1, t_2,... and α are derivable, and the substitution $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \alpha$ is defined, then so is the substitution $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}(\alpha)$ and $\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \alpha\Big) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}(\alpha)$. Likewise for terms τ.*

Proof.

We prove this by structural induction on α and τ.

- τ is a variable symbol The consequent of the sequent we have to derive is $\top$.
- $\tau \equiv f(\tau_1, \tau_2, \ldots, \tau_m)$ We have to derive

$$\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \tau_1\Big) \,\&\, \cdots \,\&\, \boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \tau_m\Big) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}(\tau_1) \,\&\, \cdots \,\&\, \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}(\tau_m)$$

  which is easily obtained using induction on τ_1, τ_2, ..., τ_m.

- $\tau \equiv \iota x_\psi(\varphi)$ where x is not one of $x_1, x_2, \ldots, x_n$ We have to derive

$$\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \psi\Big) \,\&\, \forall x \left(\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \varphi\Big)\right) \,\&\, \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \psi \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \psi$$

  which is easy.

- $\tau \equiv \iota x_{i\psi}(\varphi)$ We have to derive

$$\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \psi\Big) \& \forall x (\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\!\right] \varphi\Big)) \& \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \psi \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \psi$$

  which is easy.

- $\alpha \equiv \forall x(\beta)$ where x is not one of $x_1, x_2, \ldots, x_n$ We have to derive $\forall x \left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \beta \right) \right) \vdash_\iota \forall x \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathbf{\Delta}(\beta) \right)$ which is easy using induction on α.

- $\alpha \equiv \forall x_i(\beta)$ We have to derive

$$\forall x_i \left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \beta \right) \right) \vdash_\iota \forall x_i \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \mathbf{\Delta}(\beta) \right)$$

  which is easy using induction on α.

- The other cases are easy.

□

Lemma 59 *If the uniqueness conditions of t_1, t_2,... and α are derivable, x_i is not a free variable of α, and one of the substitutions $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ and $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t'_i & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ is defined, then both formulae are identical and interchangeable with $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$.*

The analogous property for terms τ of the PITFOL *calculus also holds.*

Proof.

We prove this by structural induction on τ and α. The only interesting case is $\tau \equiv \iota x_\psi(\varphi)$.

- If $x \notin \{x_1, x_2, \ldots\}$ then we have to show that $\iota x_{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \& \forall x \left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \& \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi} \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi \right)$ is identical to

$$\iota x_{\substack{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t'_i & t_{i+1} & \ldots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \\ \& \forall x \left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t'_i & t_{i+1} & \ldots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \\ \& \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t'_i & t_{i+1} & \ldots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \psi}} \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t'_i & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \varphi \right)$$

  and interchangeable with

$$\iota x_{\substack{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \\ \& \forall x \left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \\ \& \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \psi}} \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \varphi \right)$$

  which is easy using induction.

- If $x \equiv x_i$ then if $n > 1$ we have to show that

$$\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)\right) \atop \&\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]\psi} \left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)$$

  is identical to

$$\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t'_i & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\psi\right) \atop \&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)\right) \atop \&\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t'_i & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\psi} \left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)$$

  and interchangeable with

$$\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\psi\right) \atop \&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)\right) \atop \&\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\psi} \left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)$$

  which is easy using induction.

  If $n = 1$ then this reduces to showing that $\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x(\mathbf{\Delta}(\varphi))\&\left[\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix}\right]\!\right]\psi}(\varphi)$ is identical to $\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t'_1 \\ x_1 \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x(\mathbf{\Delta}(\varphi))\&\left[\!\left[\begin{smallmatrix} t'_1 \\ x_1 \end{smallmatrix}\right]\!\right]\psi}(\varphi)$ and interchangeable with $\iota x_\psi(\varphi)$, which is not difficult.

- If $x \equiv x_j$ and $j \not\equiv i$, where we suppose $i < j$ for the ease of notation, then if $n > 2$ we have to show that

$$\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{j-1} & t_{j+1} & \dots & t_n \\ x_1 & & x_{j-1} & x_{j+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)\right) \atop \&\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]\psi} \left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{j-1} & t_{j+1} & \dots & t_n \\ x_1 & & x_{j-1} & x_{j+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)$$

  is identical to

$$\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t'_i & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\psi\right) \atop \&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t'_i & t_{i+1} & \dots & t_{j-1} & t_{j+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_{j-1} & x_{j+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)\right) \atop \&\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t'_i & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\psi} \left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t'_i & t_{i+1} & \dots & t_{j-1} & t_{j+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_{j-1} & x_{j+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)$$

  and interchangeable with

$$\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\psi\right) \atop \&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_{j-1} & t_{j+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_{j-1} & x_{j+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)\right) \atop \&\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\psi} \left(\left[\!\left[\begin{smallmatrix} t_1 & \dots & t_{i-1} & t_{i+1} & \dots & t_{j-1} & t_{j+1} & \dots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_{j-1} & x_{j+1} & & x_n \end{smallmatrix}\right]\!\right]\varphi\right)$$

If $n = 2$ then $i = 1$ and $j = 2$ and we have to show that $\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & t_2 \\ x_1 & x_2 \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix}\right]\!\right]\varphi\right)\right)\&\left[\!\left[\begin{smallmatrix} t_1 & t_2 \\ x_1 & x_2 \end{smallmatrix}\right]\!\right]\psi}\left(\left[\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix}\right]\!\right]\varphi\right)$ is identical to $\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t'_1 & t_2 \\ x_1 & x_2 \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t'_1 \\ x_1 \end{smallmatrix}\right]\!\right]\varphi\right)\right)\&\left[\!\left[\begin{smallmatrix} t'_1 & t_2 \\ x_1 & x_2 \end{smallmatrix}\right]\!\right]\psi}\left(\left[\!\left[\begin{smallmatrix} t'_1 \\ x_1 \end{smallmatrix}\right]\!\right]\varphi\right)$ and interchangeable with $\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_2 \\ x_2 \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x(\mathbf{\Delta}(\varphi))\&\left[\!\left[\begin{smallmatrix} t_2 \\ x_2 \end{smallmatrix}\right]\!\right]\psi}(\varphi)$.

Both cases are analogous to the previous ones.

□

Note that in general, if x_i is not a free variable of α, then $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]\alpha$ is not identical to $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\right]\alpha$ but only interchangeable. For example, if x_1 is not a free variable of α, then the first formula is $\left[\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix}\right]\!\right]\iota x_\psi(\varphi) \equiv \iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x(\mathbf{\Delta}(\varphi))\&\left[\!\left[\begin{smallmatrix} t_1 \\ x_1 \end{smallmatrix}\right]\!\right]\psi}(\varphi)$ which is in general not identical to $\iota x_\psi(\varphi)$.

Lemma 60 *If the uniqueness conditions of t_1, t_2, ..., t_n, s_1, s_2, ..., s_m and α are derivable, $\{x_1, x_2, \ldots, x_n\} \cap (FV(\alpha) \setminus \{y_1, y_2, \ldots, y_m\}) = \emptyset$ and the substitutions $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]\left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\right]\alpha$ are defined, then this formula is interchangeable with $\left[\!\left[\begin{smallmatrix} \left[\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]s_1 & \cdots & \left[\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\right]\alpha$ when the uniqueness conditions of one of both are derivable.*

The analogous property for terms τ of the PITFOL *calculus also holds.*

Proof.

We prove this by structural induction on α and τ.

- If $\tau \equiv y_i$ then both terms are $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]s_i$.
- If $\tau \equiv x_i$ then the statement of the lemma requires that $x_i \in \{y_1, y_2, \ldots, y_m\}$, so we already handled that case.
- If τ is a variable symbol not in $\{x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m\}$ then both terms are τ.
- If $\tau \equiv f(\tau_1, \ldots, \tau_k)$ then induction on the τ_i and the ERf2 rule easily yield the desired result.
- If $\tau \equiv \iota x_\psi(\varphi)$ and $x \notin \{y_1, y_2, \ldots, y_m\}$ then we have to show the interchangeability of

$$\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right]\iota x_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} s_1 & \ldots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\right]\psi\right)\&\forall x\left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} s_1 & \ldots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\right]\varphi\right)\right)\&\left[\!\left[\begin{smallmatrix} s_1 & \ldots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\right]\psi}\left(\left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\right]\varphi\right)$$

and $\iota x_{\boldsymbol{\Delta}(\psi_2)\&\forall x(\boldsymbol{\Delta}(\varphi_2))\&\psi_2}(\varphi_2)$ where we abbreviate $\psi_2 :\equiv \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix} \right]\!\!\right] s_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix} \right]\!\!\right] s_m \Big]\!\!\Big] \psi$ and $\varphi_1 :\equiv \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix} \right]\!\!\right] s_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix} \right]\!\!\right] s_m \Big]\!\!\Big] \varphi$.

If also $x \notin \{x_1, x_2, \dots, x_n\}$ then the first term is $\iota x_{\boldsymbol{\Delta}(\psi_1)\&\forall x(\boldsymbol{\Delta}(\varphi_1))\&\psi_1}(\varphi_1)$ where we abbreviate

$$\psi_1 :\equiv \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \Big(\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \psi\Big) \& \forall x \Big(\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \varphi\Big)\Big) \& \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \psi\Big)$$

and $\varphi_1 :\equiv \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \varphi$.

By induction, φ_1 and φ_2 are interchangeable. We will now show that $\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1$ and $\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2$ are also interchangeable.

It is easy to derive $\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1)$ and $\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2)$, hence we only have to derive $\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \dashv\vdash_\iota \boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2$.

$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1$ ass
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1$ &-elim
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \psi_1$ &-elim
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \psi$ &-elim
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \psi_2$ induction
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \boldsymbol{\Delta}(\psi_2)$ defCons
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \forall x(\boldsymbol{\Delta}(\varphi_1))$ &-elim
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \forall x(\boldsymbol{\Delta}(\varphi_2))$ induction, Th. 32
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2))$ &-intro
$\boldsymbol{\Delta}(\psi_1) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_1)) \,\&\, \psi_1 \vdash_\iota \boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2$ &-intro

$\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2 \vdash_\iota \boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2$ ass
$\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2 \vdash_\iota \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2$ &-elim
$\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2 \vdash_\iota \psi_2$ &-elim
$\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2 \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \psi$ induction
$\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2 \vdash_\iota \boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \psi\Big)$ defCons
$\boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \psi\Big) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \psi\Big)$ Lemma 58
$\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2 \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \boldsymbol{\Delta}\Big(\left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \psi\Big)$ Cut
$\boldsymbol{\Delta}(\psi_2) \,\&\, \forall x(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \psi_2 \vdash_\iota \forall x(\boldsymbol{\Delta}(\varphi_2))$ &-elim

$$\begin{array}{rl}
\boldsymbol{\Delta}(\psi_2)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_2))\,\&\,\psi_2 \vdash_\iota \forall x(\boldsymbol{\Delta}(\varphi_1)) & \text{induction, Th. 32}\\
\boldsymbol{\Delta}(\varphi_1) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \varphi\right) & \text{Lemma 58}\\
\forall x(\boldsymbol{\Delta}(\varphi_1)) \vdash_\iota \forall x \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \varphi\right)\right) & \text{SimGen}\\
\boldsymbol{\Delta}(\psi_2)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_2))\,\&\,\psi_2 \vdash_\iota \forall x \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \varphi\right)\right) & \text{Cut}\\
\boldsymbol{\Delta}(\psi_2)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_2))\,\&\,\psi_2 \vdash_\iota \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \psi\right) & \\
\qquad\qquad \&\,\forall x \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \varphi\right)\right) & \&\text{-intro}\\
\boldsymbol{\Delta}(\psi_2)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_2))\,\&\,\psi_2 \vdash_\iota \psi_1 & \&\text{-intro}\\
\boldsymbol{\Delta}(\psi_2)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_2))\,\&\,\psi_2 \vdash_\iota \boldsymbol{\Delta}(\psi_1) & \text{defCons}\\
\boldsymbol{\Delta}(\psi_2)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_2))\,\&\,\psi_2 \vdash_\iota \boldsymbol{\Delta}(\psi_1)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_1)) & \&\text{-intro}\\
\boldsymbol{\Delta}(\psi_2)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_2))\,\&\,\psi_2 \vdash_\iota \boldsymbol{\Delta}(\psi_1)\,\&\,\forall x(\boldsymbol{\Delta}(\varphi_1))\,\&\,\psi_1 & \&\text{-intro}
\end{array}$$

If however $x \equiv x_i$ then the first term again has the form $\iota x_{\boldsymbol{\Delta}(\psi_1)\&\forall x(\boldsymbol{\Delta}(\varphi_1))\&\psi_1}(\varphi_1)$ where now $\varphi_1 \equiv \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \varphi$ and $\psi_1 \equiv \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \left(\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \psi\right) \,\&\, \forall x(\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \varphi\right)) \,\&\, \left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \psi\right)$. This case is analogous to the previous one, except that here we we cannot apply induction directly to show that φ_1 and φ_2 are interchangeable.

If $n = 1$, then $\varphi_1 \equiv \left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \varphi \equiv \varphi_2$ and we are done.

If $n > 1$, we will show for $j = 0, 1, \ldots, m$ that φ_2 is interchangeable with $\Phi_j :\equiv \left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & \ldots & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] s_1 \ldots \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & \ldots & x_n \\ & y_j & \end{smallmatrix}\right]\!\!\right] s_j \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & \ldots & x_{i-1} & x_{i+1} & \ldots & x_n \\ & & y_{j+1} & & & \end{smallmatrix}\right]\!\!\right] s_{j+1} \ldots \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & \ldots & x_{i-1} & x_{i+1} & \ldots & x_n \\ & & y_m & & & \end{smallmatrix}\right]\!\!\right] s_m \right]\!\!\right] \varphi$ where $\Phi_0 \equiv \left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & \ldots & x_{i-1} & x_{i+1} & \ldots & x_n \\ & & y_1 & & & \end{smallmatrix}\right]\!\!\right] s_1 \ldots \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & \ldots & x_{i-1} & x_{i+1} & \ldots & x_n \\ & & y_m & & & \end{smallmatrix}\right]\!\!\right] s_m \right]\!\!\right] \varphi$ and $\Phi_m \equiv \varphi_2$. We do this by showing that Φ_j is interchangeable with Φ_{j+1} for $j = 0, 1, \ldots, m-1$. The construction of these two formulas only differs in the term which is substituted for y_j. Because we assumed the substitution $\left[\!\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \iota x_{i\psi}(\varphi)$ to be defined, we have two possibilities:

- y_j is not a free variable of φ. Hence, by the previous lemma, $\Phi_j \equiv \Phi_{j+1}$.

– x_i is not a free variable of s_j. Again invoking the previous lemma, we get that $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right]$ and $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right]$ are interchangeable. Using property 55, we get the interchangeability of Φ_j and Φ_{j+1}.

We have obtained that φ_2 is interchangeable with Φ_0; induction immediately yields that Φ_0 is interchangeable with φ_1.

- If $\tau \equiv \iota x_\psi(\varphi)$ and $x \equiv y_k$ then we have to show the interchangeability of

$$\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \iota x \begin{array}{c} \boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \ldots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right]\psi\right) \\ \&\forall x\left(\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{k-1} & t_{k+1} & \ldots & t_n \\ x_1 & \ldots & x_{k-1} & x_{k+1} & \ldots & x_n \end{smallmatrix} \right]\!\!\right]\varphi\right)\right) \\ \&\left[\!\!\left[\begin{smallmatrix} s_1 & \ldots & s_m \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right]\psi \end{array} \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{k-1} & t_{k+1} & \cdots & t_n \\ x_1 & & x_{k-1} & x_{k+1} & & x_n \end{smallmatrix} \right]\!\!\right]\varphi\right)$$

and $\iota x_{\boldsymbol{\Delta}(\psi_2)\&\forall x(\boldsymbol{\Delta}(\varphi_2))\&\psi_2}(\varphi_2)$ where

$$\varphi_2 \equiv \left[\!\!\left[\begin{smallmatrix} \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right]s_1 & \ldots & \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right]s_{k-1} & \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right]s_{k+1} & \ldots & \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right]s_m \\ y_1 & & y_{k-1} & y_{k+1} & & y_m \end{smallmatrix} \right]\!\!\right]\varphi$$

and $\psi_2 \equiv \left[\!\!\left[\begin{smallmatrix} \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right]s_1 & \cdots \\ y_1 & \end{smallmatrix} \right]\!\!\right]\psi$. This case is analogous to the previous one.

- The other cases are easy.

□

5.2 Defined symbols

We will extend the PITFOL calculus with a finite number of so called **defined function symbol**s and **defined predicate symbol**s. We will denote the resulting extension as the PITFOL' calculus. Instead of $\boldsymbol{\Delta}$ (which is only defined on terms and formulae of the PITFOL calculus), this calculus will use $\boldsymbol{\Delta}'$, which is defined in the same way as $\boldsymbol{\Delta}$ except on defined symbols.

These defined symbols are supposed to be added to the calculus in a certain order. Hence we can speak of defined symbols added before a certain defined symbol, etc. We call the position of a defined symbol in the series of defined symbols its **index**; the first defined symbol has index 1 and so on. For convenience, we define the index of a function or predicate symbol that

is not a defined symbol as 0, so we always can talk about "the highest index of defined symbols in a given formula or term".

We will add a ι' subscript to the $\vdash$ symbol to indicate sequents of the PITFOL$'$ calculus.

For a defined function symbol g, we define

$$\boldsymbol{\Delta}'(g(t_1, t_2, \ldots, t_n)) := \boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*}\right)$$

where g^* is a term of the PITFOL' calculus containing only symbols defined before g. As before, $\widetilde{g^*}$ is obtained from g^* by changing the names of all bound variables, but here we require that the bound variables all be renamed such that they are different from the free variables of t_1, t_2, $\ldots$, t_n. This renaming is necessary to ascertain that the simultaneous substitution is always defined. For each defined function symbol, we require that $UC(g^*)$ be derivable. We also add a new rule to the calculus:

$$\frac{\boxed{\text{definition}} \quad UC(t_1), UC(t_2), \ldots, UC(t_n)}{\boldsymbol{\Delta}'(g(t_1, \ldots, t_n)) \vdash_{\iota'} g(t_1, \ldots, t_n) = \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*}}$$

We will call g^* the **defining term** of g.

For a defined predicate symbol q, we define

$$\boldsymbol{\Delta}'(q(t_1, t_2, \ldots, t_n)) := \boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{q^*}\right)$$

where q^* is a formula of the PITFOL' calculus containing only symbols defined before q and for which $UC(q^*)$ is derivable. We again add a new rule to the calculus:

$$\frac{\boxed{\text{definition}} \quad UC(t_1), UC(t_2), \ldots, UC(t_n)}{\boldsymbol{\Delta}'(q(t_1, \ldots, t_n)) \vdash_{\iota'} q(t_1, \ldots, t_n) \Leftrightarrow \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{q^*}}$$

We will call q^* the **defining formula** of q.

We call $x_1, \ldots, x_n$ the **argument variable symbol**s of g^* and q^* and require that $FV(g^*) \subseteq \{x_1, x_2, \ldots, x_n\}$ and likewise for q^*.

We will abbreviate "the defined function symbol g with argument symbols $x_1, x_2, \ldots, x_n$ and with defining term $g^* \equiv \varphi$" as "the definition $g(x_1, x_2, \ldots, x_n) = \varphi$", e.g. "the definition $\mathrm{frac}(x, y) = \iota z_{\neg(y=0)}(z \cdot y = x)$".

Note that in contrast to PVS, the addition of a defined symbol does not cause an extra formula to appear in the context of sequents.

5.2.1 Well-foundedness of the definition of Δ'

Remark that we define Δ' recursively; it is not immediately clear that this recursion is well-founded. Contrast this with the definition of Δ where it is clear that to calculate $\Delta(\alpha)$ we recursively need to calculate Δ of formulae or terms with smaller complexity than α. However, in the calculation Δ' of a defined symbol application $g(t_1, \ldots, t_n)$, the complexity of $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{q^*}$ can be much larger than the complexity of $g(t_1, \ldots, t_n)$: we need to search for an other measure than complexity to show that the recursion always terminates.

We define the **expansion rank** $\rho(\alpha)$ resp. $\rho(\tau)$ of a formula α or term τ of the PITFOL' calculus as

- $\rho(x) := 0$
- $\rho(f(\tau_1, \tau_2, \ldots, \tau_n)) := \max_i \rho(\tau_i)$; if f has no arguments, then $\rho(f()) := 0$.
- $\rho(\iota x_\psi(\varphi)) := \max(\rho(\psi), \rho(\varphi))$
- $\rho(g(\tau_1, \tau_2, \ldots, \tau_n)) := N^j + \max_i \rho(\tau_i)$ where g is the j-th defined function symbol; if g has no arguments, then $\rho(g()) := N^j$.
- $\rho(p(\tau_1, \tau_2, \ldots, \tau_n)) := \max_i \rho(\tau_i)$; if p has no arguments, then $\rho(p()) := 0$.
- $\rho(q(\tau_1, \tau_2, \ldots, \tau_n)) := N^j + \max_i \rho(\tau_i)$ where q is the j-th defined predicate symbol; if q has no arguments, then $\rho(q()) := N^j$.
- $\rho(\neg\alpha) := \rho(\alpha)$
- $\rho(\alpha \,\&\, \beta) := \max(\rho(\alpha), \rho(\beta))$
- $\rho(\forall x(\alpha)) := \rho(\alpha)$

where we choose N such that for all defined symbols, $\rho(g^*) < N^j$ where g is the j-th defined function or predicate symbol.

It is indeed possible to find such a N: one sees easily that $\rho(g^*)$ is a polynomial $P_j(N)$ with non-negative coefficients of degree at most $j-1$ in N where g is the j-th defined symbol. Since there are only a finite number of defined symbols, by choosing N large enough, we can indeed satisfy $P_j(N) < N^j$ for all j.

For convenience, we set $\rho(\top) \equiv 0$.

We define a sequence of metalogical operators Δ'_k where $k > 0$ which transform a formula or term with expansion rank at most k into another formula or term or the symbol $\top$:

- $\boldsymbol{\Delta}'_0(\alpha) := \boldsymbol{\Delta}(\alpha)$ and $\boldsymbol{\Delta}'(\tau) := \boldsymbol{\Delta}(\tau)$
- For $k > 0$, $\boldsymbol{\Delta}'_k$ is defined analogously to $\boldsymbol{\Delta}$, for example

  $$\boldsymbol{\Delta}'_k(\alpha \,\&\, \beta) := \boldsymbol{\Delta}'_k(\alpha) \,\&\, (\alpha \Rightarrow \boldsymbol{\Delta}'_k(\beta))$$

  with the extra rule

  $$\boldsymbol{\Delta}'_k(q(t_1, t_2, \ldots, t_n)) := \boldsymbol{\Delta}'_{k-1}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{q^*}\right)$$

  where q is a defined predicate symbol, and likewise for defined function symbols.

  We will show shortly (see corollary 62) that $\rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{q^*}\right) < k$ so $\boldsymbol{\Delta}'_{k-1}$ can indeed be applied to the formula $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{q^*}$.

We now define $\boldsymbol{\Delta}'(\alpha) := \boldsymbol{\Delta}'_{\rho(\alpha)}(\alpha)$.

Theorem 61 *Given a formula α and n terms $t_1, t_2, \ldots, t_n$ of the* PITFOL' *calculus. If the operators $\boldsymbol{\Delta}'_0, \ldots, \boldsymbol{\Delta}'_{\rho(\alpha)+\max_{i=1}^n \rho(t_i)}$ are already defined, and $\rho(\boldsymbol{\Delta}'(t_i)) \leq \rho(t_i)$ for all $i = 1 \ldots n$ then*

$$\rho\left(\boldsymbol{\Delta}'_{\rho(\alpha)+\max_{i=1}^n \rho(t_i)}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha\right)\right) \leq \rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha\right) \leq \rho(\alpha) + \max_{i=1}^n \rho(t_i)$$

if the simultaneous substitution is defined. Likewise for terms τ.

For $n = 0$, this collapses to

$$\rho(\boldsymbol{\Delta}'(\alpha)) \leq \rho(\alpha) \quad \text{if } \boldsymbol{\Delta}'_0, \ldots, \boldsymbol{\Delta}'_{\rho(\alpha)} \text{ are already defined.}$$

Proof.

We prove this by induction on the highest index of defined symbols in α and τ. This we prove by structural induction on α and τ:

- $\tau \equiv x_i$ We have to show that $\rho\left(\boldsymbol{\Delta}'_{\max_{i=1}^n \rho(t_i)}(t_i)\right) \leq \rho(t_i) \leq \max_i \rho(t_i)$, which was already assumed in the statement of the theorem.
- $\tau \equiv x$ with $x \notin \{x_1, x_2, \ldots, x_n\}$ We have to show that $0 \leq \max_i \rho(t_i)$, which is trivial.
- $\tau \equiv f(\tau_1, \ldots, \tau_m)$ We have to show that $\max_i \rho\left(\boldsymbol{\Delta}'_{\rho(\tau)+\max_{i=1}^n \rho(t_i)}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_i\right)\right) \leq \max_i \rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_i\right) \leq \max_i \rho(\tau_i) + \max_i \rho(t_i)$ which immediately follows from structural induction on the τ_i.

- $\tau \equiv g(\tau_1, \ldots, \tau_m)$ where g is the j-th defined function symbol We have to show that

$$\rho\left(\boldsymbol{\Delta}'_{\rho(\tau)+\max_{i=1}^n \rho(t_i)}\left(\left[\!\!\left[\begin{matrix}\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_1 & \cdots & \left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_m\\ y_1 & & y_m\end{matrix}\right]\!\!\right]\widetilde{g^*}\right)\right)$$
$$\leq N^j + \max_i \rho\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\right)$$
$$\leq N^j + \max_i \rho(\tau_i) + \max_i \rho(t_i)\,.$$

We can apply structural induction on all τ_i to obtain that $\rho\Big(\boldsymbol{\Delta}'_{\rho(\tau_i)+\max_{i=1}^n \rho(t_i)}\Big(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\Big)\Big) \leq \rho\Big(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\Big) \leq \rho(\tau_i) + \max_i \rho(t_i)$, from which the second inequality readily follows.

To obtain the first inequality, we apply induction on $\widetilde{g^*}$, since it contains only defined symbols with index $< j$, to obtain that if $\boldsymbol{\Delta}'_0, \ldots, \boldsymbol{\Delta}'_{\rho(\widetilde{g^*})+\max_i \rho\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\right)}$ are already defined—which is the case, because we supposed $\boldsymbol{\Delta}'_{\rho(\tau)+\max_{i=1}^n \rho(t_i)}$ to be defined and noting that by definition $\rho(g^*) < N^j$, we have $\rho\big(\widetilde{g^*}\big) + \max_i \rho\Big(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\Big) \leq N^j + \max_i \rho(\tau_i) + \max_i \rho(t_i) = \rho(\tau) + \max_{i=1}^n \rho(t_i)$—and if $\rho\Big(\boldsymbol{\Delta}'\Big(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\Big)\Big) \leq \rho\Big(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\Big)$—which we just proved—then

$$\rho\left(\boldsymbol{\Delta}'_{\rho(\widetilde{g^*})+\max_i \rho\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\right)}\left(\left[\!\!\left[\begin{matrix}\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_1 & \cdots & \left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_m\\ y_1 & & y_m\end{matrix}\right]\!\!\right]\widetilde{g^*}\right)\right)$$
$$\leq \rho\left(\left[\!\!\left[\begin{matrix}\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_1 & \cdots & \left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_m\\ y_1 & & y_m\end{matrix}\right]\!\!\right]\widetilde{g^*}\right)$$
$$\leq \rho\big(\widetilde{g^*}\big) + \max_i \rho\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i\right).$$

- $\tau \equiv \iota x_{i\psi}(\varphi)$ We have to show that

$$\max\left(\begin{array}{l}\rho\left(\boldsymbol{\Delta}'_k\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\psi\right)\right), \rho\left(\boldsymbol{\Delta}'_k\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n\\ x_1 & & x_{i-1} & x_{i+1} & & x_n\end{smallmatrix}\right]\!\!\right]\varphi\right)\right),\\ \rho\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\psi\right)\end{array}\right)$$
$$\leq \max\left(\begin{array}{l}\rho\left(\boldsymbol{\Delta}'_k\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\psi\right)\right), \rho\left(\boldsymbol{\Delta}'_k\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n\\ x_1 & & x_{i-1} & x_{i+1} & & x_n\end{smallmatrix}\right]\!\!\right]\varphi\right)\right),\\ \rho\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\psi\right), \rho\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n\\ x_1 & & x_{i-1} & x_{i+1} & & x_n\end{smallmatrix}\right]\!\!\right]\varphi\right)\end{array}\right)$$

$\leq \max(\rho(\psi), \rho(\varphi)) + \max_i \rho(t_i)$,

where $k = \rho(\tau) + \max_{i=1}^n \rho(t_i)$, which is easy using the structural induction on ψ and φ.

- $\tau \equiv \iota x_\psi(\varphi)$ where $x \notin \{x_1, x_2, \ldots, x_n\}$ Analogous.
- $\tau \equiv p(\tau_1, \ldots, \tau_m)$ Analogous to the case $f(\tau_1, \ldots, \tau_m)$.
- $\tau \equiv q(\tau_1, \ldots, \tau_m)$ where q is the j-th defined predicate symbol Analogous to the case $g(\tau_1, \ldots, \tau_m)$.
- $\alpha \equiv \neg\beta$ Replacing α by β does not change the desired inequalities, so we get them by applying structural induction on β.
- $\alpha \equiv \beta \;\&\; \gamma$ We have to show that $\max\left(\rho\left(\boldsymbol{\Delta}'_k\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \beta\right)\right), \rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \beta\right), \rho\left(\boldsymbol{\Delta}'_k\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \gamma\right)\right)\right) \leq \max\left(\rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \beta\right), \rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \gamma\right)\right) \leq \max(\rho(\alpha), \rho(\beta)) + \max_i \rho(t_i)$ where $k = \rho(\alpha) + \max_{i=1}^n \rho(t_i)$ which we obtain immediately using structural induction on α and β.
- $\alpha \equiv \forall x_i(\beta)$ We have to show that $\rho\left(\boldsymbol{\Delta}'_k\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\!\right] \beta\right)\right) \leq \rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_{i-1} & t_{i+1} & \cdots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix}\right]\!\!\right] \beta\right) \leq \rho(\beta) + \max_i \rho(t_i)$ where $k = \rho(\alpha) + \max_{i=1}^n \rho(t_i)$. By induction on β, the first two comparands are actually at most $\rho(\beta) + \max(\rho(t_1), \ldots, \rho(t_{i-1}), \rho(t_{i+1}), \ldots, \rho(t_n))$.
- $\alpha \equiv \forall x(\beta)$ where $x \notin \{x_1, x_2, \ldots, x_n\}$ Replacing α by β does not change the desired inequalities, so we get them by applying structural induction on β.

□

Now we can show that $\mathcal{D}'_k$ is well defined:

Corollary 62 *Suppose $\boldsymbol{\Delta}'_0, \ldots, \boldsymbol{\Delta}'_{k-1}$ are defined. If $\rho(q(t_1, \ldots, t_n)) \leq k$, then $\rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \widetilde{q^*}\right) < k$ where q is a defined predicate or function symbol.*

Proof.

The previous theorem yields that $\rho(\boldsymbol{\Delta}'(t_i)) \leq \rho(t_i)$ when $\boldsymbol{\Delta}'_{\rho(t_i)}$ is defined, which is the case since $\rho(t_i) < N^j + \max_{i=1}^n \rho(t_i) = \rho(q(t_1, \ldots, t_n)) \leq k$ where j is the index of q.

Hence we can again apply the previous theorem to obtain that

$$\rho\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{q^*}\right) \leq \rho\big(\widetilde{q^*}\big) + \max_{i=1}^{n} \rho(t_i)$$

provided $\boldsymbol{\Delta}'_{\rho\left(\widetilde{q^*}\right)+\max_{i=1}^{n}\rho(t_i)}$ is defined. This is again the case because $\rho\big(\widetilde{q^*}\big) + \max_{i=1}^{n} \rho(t_i) < N^j + \max_{i=1}^{n} \rho(t_i) < k$. □

Corollary 63 *Given a formula α and n terms $t_1, t_2, \ldots, t_n$ of the* PITFOL' *calculus. Then*

$$\rho\left(\boldsymbol{\Delta}'_k\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \alpha\right)\right) \;\leq\; \rho\left(\left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \alpha\right) \;\leq\; \rho(\alpha) + \max_{i=1}^{n} \rho(t_i)$$

if the simultaneous substitution is defined, and likewise for terms τ.

For $n = 0$, this again collapses to

$$\rho(\boldsymbol{\Delta}'(\alpha)) \leq \rho(\alpha)\,.$$

Corollary 64 *Given n terms $t_1, t_2, \ldots, t_n$ of the* PITFOL' *calculus. Then $\rho\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \widetilde{g^*}\right) < \rho(g(t_1, \ldots, t_n))$ where g is a defined function or predicate symbol.*

5.3 Expansion of proofs

We will prove that adding defined symbols to the logic does not modify its consistency.

To prove this, we define the **expansion** $\mathcal{E}(\alpha)$ and $\mathcal{E}(t)$ of a formula α or term t of the PITFOL' calculus. These both yield a formula resp. term of the PITFOL calculus. We will then show that when we replace in a proof all formulae by their expansions, we get a correct proof in the PITFOL calculus—just like we did before with $\mathcal{R}$ and $\mathcal{D}$.

The expansion of formulae and terms is defined as follows:

- $\mathcal{E}(x) \equiv x$
- $\mathcal{E}(f(t_1, t_2, \ldots, t_n)) \equiv f(\mathcal{E}(t_1), \mathcal{E}(t_2), \ldots, \mathcal{E}(t_n))$ if f is not a defined function symbol

- $\mathcal{E}(g(t_1,\ldots,t_n)) \equiv \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}\left(\widetilde{g^*}\right)$ if g is a defined function symbol; $x_1,\ldots,x_n$ are the argument variable symbols used in the definition of g; $\widetilde{g^*}$ is as before obtained from g^* by changing the names of all bound variables, but here we require that the bound variables all be renamed such that they are different from the free variables of t_1, t_2, …, t_n. This renaming is again necessary to ascertain that the simultaneous substitution is always defined.
- $\mathcal{E}(\iota x_\psi(\varphi)) \equiv \iota x_{\mathcal{E}(\psi)}(\mathcal{E}(\varphi))$
- $\mathcal{E}(p(t_1,t_2,\ldots,t_n)) \equiv p(\mathcal{E}(t_1),\mathcal{E}(t_2),\ldots,\mathcal{E}(t_n))$ if p is not a defined predicate symbol, and likewise $\mathcal{E}(t_1 = t_2) \equiv \mathcal{E}(t_1) = \mathcal{E}(t_2)$
- $\mathcal{E}(q(t_1,t_2,\ldots,t_n)) \equiv \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}\left(\widetilde{q^*}\right)$ if q is a defined predicate symbol; $x_1,\ldots,x_n$ are the argument variable symbols used in the definition of q; again, $\widetilde{q^*}$ is obtained from q^* by renaming all bound variables to variables different from different from the free variables of t_1, t_2, …, t_n.
- $\mathcal{E}(\neg\alpha) \equiv \neg\mathcal{E}(\alpha)$
- $\mathcal{E}(\alpha \,\&\, \beta) \equiv \mathcal{E}(\alpha) \,\&\, \mathcal{E}(\beta)$
- $\mathcal{E}(\forall x(\alpha)) \equiv \forall x(\mathcal{E}(\alpha))$
- For definedness properties, it will be handy to define $\mathcal{E}(\top) \equiv \top$.

Lemma 65 *For any formula α and terms $t_1,\ldots,t_n$ of the* PITFOL' *calculus, $\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha\right)$ does not contain ι-terms if and only if $\mathcal{E}(\alpha)$ does not contain ι-terms and for each $i = 1,\ldots,n$ either x_i is not free in $\mathcal{E}(\alpha)$ or $\mathcal{E}(t_i)$ does not contain ι-terms.*

The analogous theorem for terms τ also holds.

Proof.

We prove this by structural induction on α and τ. The interesting cases are

- $\tau \equiv g(\tau_1,\ldots,\tau_m)$ where g is a defined function symbol We have to show that $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_1\right) & \ldots & \mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_m\right) \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \mathcal{E}\left(\widetilde{g^*}\right)$ does not contain ι-terms (A) if and only if $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(\tau_1) & \ldots & \mathcal{E}(\tau_m) \\ y_1 & & y_m \end{smallmatrix} \right]\!\!\right] \mathcal{E}\left(\widetilde{g^*}\right)$ does not contain ι-terms $\&\forall i\big(x_i \notin$

$FV(\left[\!\left[\begin{smallmatrix} \mathcal{E}(\tau_1) & \cdots & \mathcal{E}(\tau_m) \\ y_1 & & y_m \end{smallmatrix} \right]\!\right] \mathcal{E}\left(\widetilde{g^*}\right)) \ \vee \ \mathcal{E}(t_i)$ does not contain ι-terms$\big)$. (B)

By property 53, (A) is equivalent with: $\mathcal{E}\left(\widetilde{g^*}\right)$ does not contain ι-terms $\&\forall j\big(y_j \notin FV(\mathcal{E}\left(\widetilde{g^*}\right)) \vee \mathcal{E}\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \tau_j\right)$ does not contain ι-terms$\big)$.

Using induction, this is in turn equivalent with: $\mathcal{E}\left(\widetilde{g^*}\right)$ does not contain ι-terms $\&\forall j\Big(y_j \notin FV(\mathcal{E}\left(\widetilde{g^*}\right)) \vee \big(\mathcal{E}(\tau_j)$ does not contain ι-terms $\&\forall i(x_i \notin FV(\mathcal{E}(\tau_j)) \vee \mathcal{E}(t_i)$ does not contain ι-terms$)\big)\Big)$. (A')

Next, consider (B). Using properties 53 and 54 this is equivalent with: $\mathcal{E}\left(\widetilde{g^*}\right)$ does not contain ι-terms $\&\forall j\big(y_j \notin FV(\mathcal{E}\left(\widetilde{g^*}\right)) \vee \mathcal{E}(\tau_j)$ does not contain ι-terms$\big)$ $\&\forall i\Big(\underbrace{(x_i \notin FV(\mathcal{E}\left(\widetilde{g^*}\right)) \vee x_i \in \{y_1, \ldots, y_m\})}_{(*)}$ $\&\forall j\big(y_j \notin FV(\mathcal{E}\left(\widetilde{g^*}\right)) \vee x_i \notin FV(\mathcal{E}(\tau_j))\big) \vee \mathcal{E}(t_i)$ does not contain ι-terms$\Big)$. (B')

Note that $FV(\mathcal{E}\left(\widetilde{g^*}\right) \subseteq FV(g) \subseteq \{y_1, \ldots, y_m\}$, so $(*)$ is a tautology. One now easily sees that (A') and (B') are equivalent.

- $\tau \equiv \iota x_\psi(\varphi)$ Since both α and $\mathcal{E}\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \alpha\right)$ are ι-terms, this case is trivial.

□

Property 66 *For all formulae α of the* PITFOL' *calculus $\mathbf{\Delta}(\mathcal{E}(\alpha)) \equiv \top$ if and only if $\mathcal{E}(\mathbf{\Delta}'(\alpha)) \equiv \top$ and likewise for terms τ.*

Proof.

We prove this by induction on $\rho(\alpha)$ and $\rho(\tau)$, which we in turn prove by structural induction α and τ. The interesting cases are

- $\tau \equiv g(\tau_1, \ldots, \tau_n)$ where g is a defined function symbol We have to show that $\mathbf{\Delta}\Big(\left[\!\left[\begin{smallmatrix} \mathcal{E}(\tau_1) & \cdots & \mathcal{E}(\tau_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \mathcal{E}\left(\widetilde{g^*}\right)\Big) \equiv \top$ if and only if $\mathcal{E}\left(\mathbf{\Delta}'\Big(\left[\!\left[\begin{smallmatrix} \tau_1 & \cdots & \tau_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\right]\Big)\right) \widetilde{g^*} \equiv \top$.

  One shows easily that for any term t of the PITFOL' calculus, $\mathbf{\Delta}'(t) \equiv \top$ is equivalent with requiring that t does not contain ι-terms, and likewise for formulae.
  Hence, $\mathbf{\Delta}\Big(\left[\!\left[\begin{smallmatrix} \mathcal{E}(\tau_1) & \cdots & \mathcal{E}(\tau_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \mathcal{E}\left(\widetilde{g^*}\right)\Big) \equiv \top$ if and only if

$\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(\tau_1) & \dots & \mathcal{E}(\tau_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}\left(\widetilde{g^*}\right)$ does not contain ι-terms.
By property 53, this is equivalent with: $\mathcal{E}\left(\widetilde{g^*}\right)$ does not contain ι-terms and for each $i = 1, \dots, n$ either x_i is not free in $\mathcal{E}\left(\widetilde{g^*}\right)$ or $\mathcal{E}(\tau_i)$ does not contain ι-terms.
Using the previous lemma, this in turn is equivalent with requiring that $\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} \tau_1 & \dots & \tau_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right)$ does not contain ι-terms.
This is again equivalent with $\mathbf{\Delta}\left(\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} \tau_1 & \dots & \tau_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right)\right) \equiv \top$. Because of corollary 64, we can apply induction on the expansion rank to obtain the required equivalence.

- $\tau \equiv \iota x_\psi(\varphi)$ Both terms are $\mathcal{E}(\varphi)$ and hence cannot be $\top$.

□

Theorem 67 *1. For all formulae α of the* PITFOL' *calculus for which the uniqueness conditions are derivable, $\mathbf{\Delta}(\mathcal{E}(\alpha))$ and $\mathcal{E}(\mathbf{\Delta}'(\alpha))$ are either interchangeable or both equal to $\top$, and likewise for terms τ.*

2. For any formula β of the PITFOL *calculus and any terms $t_1, \dots, t_n$ of the* PITFOL *calculus for which the uniqueness conditions are derivable and for which the simultaneous substitution $\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \beta$ is defined, $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}(\beta)$ is interchangeable with $\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \beta\right)$ and likewise for terms σ.*

Proof.

Remark that in the previous property, we showed already that if one of $\mathbf{\Delta}(\mathcal{E}(\alpha))$ and $\mathcal{E}(\mathbf{\Delta}'(\alpha))$ is $\top$, then so is the other, so we only have to concern us with establishing their interchangeability when they are not $\top$.

We prove the theorem by simultaneous induction on $\rho(\alpha)$ and $\rho(\beta) + \max_i \rho(t_i)$.

For the base case, $\rho(\alpha)$ is 0, i.e., α does not contain defined symbols. Hence, $\mathcal{E}(\alpha) \equiv \alpha$, so $\mathbf{\Delta}(\mathcal{E}(\alpha)) \equiv \mathbf{\Delta}(\alpha)$. Further, $\mathbf{\Delta}'(\alpha) \equiv \mathbf{\Delta}(\alpha)$, and this formula also does not contain defined symbols, so $\mathcal{E}(\mathbf{\Delta}'(\alpha)) \equiv \mathbf{\Delta}(\alpha)$ and we see that both terms are identical.

For the second part, $\rho(\beta) = \rho(t_1) = \dots = \rho(t_n) = 0$, so no defined symbols are involved and both simultaneous substitutions are $\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \beta$.

For the induction step, we perform structural induction on α and β. We first prove the first part of the theorem. The interesting cases are:

- $\tau \equiv f(\tau_1, \tau_2, \ldots, \tau_m)$ We have to show that $\mathbf{\Delta}(\mathcal{E}(\tau_1)) \,\&\, \cdots \,\&\, \mathbf{\Delta}(\mathcal{E}(\tau_m))$ is interchangeable with $\mathcal{E}(\mathbf{\Delta}'(\tau_1)) \,\&\, \cdots \,\&\, \mathcal{E}(\mathbf{\Delta}'(\tau_m))$, which is easy by induction on the τ_i.

- $\tau \equiv \iota x_\psi(\varphi)$ We have to show that $\mathbf{\Delta}(\mathcal{E}(\psi))$ is interchangeable with $\mathcal{E}(\mathbf{\Delta}'(\psi))$ which we immediately get using structural induction.

- $\tau \equiv g(\tau_1, \tau_2, \ldots, \tau_m)$ We have to show that $\mathbf{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_m) \\ x_1 & & x_m \end{smallmatrix} \Big]\!\Big] \mathcal{E}(g^*)\Big)$ is interchangeable with $\mathcal{E}\left(\mathbf{\Delta}'\Big(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_m \\ x_1 & & x_m \end{smallmatrix} \Big]\!\Big] g^*\Big)\right)$. Because of corollary 64, we can apply induction on the first part of the theorem on $\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] g^*$. This gives us the interchangeability of $\mathcal{E}\left(\mathbf{\Delta}'\Big(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_m \\ x_1 & & x_m \end{smallmatrix} \Big]\!\Big] g^*\Big)\right)$ with $\mathbf{\Delta}\Big(\mathcal{E}\left(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] g^*\right)\Big)$. What remains is an application of the second part of the theorem. We can apply it if $\rho(g^*) + \max_i \rho(t_i) < \rho(g(\tau_1, \tau_2, \ldots, \tau_m))$, i.e., if $\rho(g^*) + \max_i \rho(t_i) < N^j + \max_i \rho(t_i)$ where j is the index of g. The inequality easily follows from the way we defined N.

We now prove the second part of the theorem. We can use the first part on formulae α for which $\rho(\alpha) \leq \rho(\beta) + \max_i \rho(t_i)$. The interesting cases are:

- $\sigma \equiv f(\sigma_1, \sigma_2, \ldots, \sigma_m)$ We have to establish the interchangeability of $f\left(\Big[\!\Big[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \mathcal{E}(\sigma_1), \ldots, \Big[\!\Big[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \mathcal{E}(\sigma_m)\right)$ and $f\left(\mathcal{E}\left(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \sigma_1\right), \ldots, \mathcal{E}\left(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \sigma_m\right)\right)$, which is easy by applying the structural induction on all of the τ_i.

- $\sigma \equiv g(\sigma_1, \sigma_2, \ldots, \sigma_m)$ We have to show that $\Big[\!\Big[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \Big[\!\Big[\begin{smallmatrix} \mathcal{E}(\sigma_1) & \ldots & \mathcal{E}(\sigma_m) \\ y_1 & & y_m \end{smallmatrix} \Big]\!\Big] \mathcal{E}\left(\widetilde{g^*}\right)$ is interchangeable with $\Big[\!\Big[\begin{smallmatrix} \mathcal{E}\left(\Big[\!\Big[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \sigma_1\right) & \ldots & \mathcal{E}\left(\Big[\!\Big[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \sigma_m\right) \\ y_1 & & y_m \end{smallmatrix} \Big]\!\Big] \mathcal{E}\left(\widetilde{g^*}\right)$. On the first term, we can apply lemma 60, and show that the result is interchangeable with the second term using property 55. To apply this property however, we require the interchangeability of $\Big[\!\Big[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \mathcal{E}(\sigma_i)$ and $\mathcal{E}\left(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \sigma_i\right)$ for $i = 1, 2, \ldots, m$, which we obtain by applying structural induction.

- $\sigma \equiv \iota x_{\psi}(\varphi)$ where $x \notin \{x_1, x_2, \ldots, x_n\}$ We have to show that

$$\iota x_{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n)\\ x_1 & \ldots & x_n\end{smallmatrix}\right]\!\!\right]\mathcal{E}(\psi)\right)\&\forall x\left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n)\\ x_1 & \ldots & x_n\end{smallmatrix}\right]\!\!\right]\mathcal{E}(\varphi)\right)\right)\ \&\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n)\\ x_1 & \ldots & x_n\end{smallmatrix}\right]\!\!\right]\mathcal{E}(\psi)} \left(\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n)\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\mathcal{E}(\varphi)\right)$$

and

$$\iota x_{\mathcal{E}\left(\mathbf{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \ldots & t_n\\ x_1 & \ldots & x_n\end{smallmatrix}\right]\!\!\right]\psi\right)\right)\&\forall x\left(\mathcal{E}\left(\mathbf{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \ldots & t_n\\ x_1 & \ldots & x_n\end{smallmatrix}\right]\!\!\right]\varphi\right)\right)\right)\&\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \ldots & t_n\\ x_1 & \ldots & x_n\end{smallmatrix}\right]\!\!\right]\psi\right)} \left(\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\varphi\right)\right)$$

are interchangeable.

To establish that $\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n)\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\mathcal{E}(\psi)$ is interchangeable with $\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\psi\right)$ and likewise for φ is easy by structural induction.

The interchangeability of $\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n)\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\mathcal{E}(\psi)\right)$ and $\mathcal{E}\left(\mathbf{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\psi\right)\right)$ and likewise for φ is somewhat more involved. By structural induction, the first term is interchangeable with $\mathbf{\Delta}\left(\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\psi\right)\right)$. If we can now apply the first part of the theorem, we are done. To be able to apply induction, we need to check that $\rho\left(\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\psi\right) \leq \rho(\iota x_{\psi}(\varphi)) + \max_i \rho(t_i)$, which immediately follows from corollary 63.

□

Note that for the second part of the theorem, it is possible that the simultaneous substitution $\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n)\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\mathcal{E}(\beta)$ is defined while the simultaneous substitution $\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & \cdots & x_n\end{smallmatrix}\right]\!\!\right]\beta$ is not defined.
For example, if g is a defined function symbol, defined as $g(y,z) = y$, then the simultaneous substitution $\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(u) & \mathcal{E}(x)\\ y & z\end{smallmatrix}\right]\!\!\right]\mathcal{E}(\iota x(x = g(y,z))) \equiv \left[\!\!\left[\begin{smallmatrix}u & x\\ y & z\end{smallmatrix}\right]\!\!\right]\iota x(x = y)$ is defined, but the simultaneous substitution $\left[\!\!\left[\begin{smallmatrix}u & x\\ y & z\end{smallmatrix}\right]\!\!\right]\iota x(x = g(y,z))$ is not defined, because substituting x for z would capture x.

For the following lemmas and theorems, one can construct analogous examples.

Lemma 68 *For any formula α of the* PITFOL *calculus for which the uniqueness conditions are derivable, and any terms $t, t_1, \ldots, t_n$ of the* PITFOL' *calculus for which the uniqueness conditions are derivable, if $x \notin$*

$FV(\alpha) \setminus \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{E}([t/x]t_i)$ is interchangeable with $[\mathcal{E}(t)/x]\mathcal{E}(t_i)$ under the condition $\mathbf{\Delta}(\mathcal{E}(t))$ for all $i \in \{1, \ldots, n\}$, and the substitutions $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \ldots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ are defined, then $[\mathcal{E}(t)/x] \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ is interchangeable with $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \ldots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ under the condition $\mathbf{\Delta}(\mathcal{E}(t))$, and likewise for terms τ of the PITFOL *calculus.*

Proof.

We prove this by structural induction.

- $\tau \equiv x_i$ We have to show that $[\mathcal{E}(t)/x]\mathcal{E}(t_i)$ and $\mathcal{E}([t/x]t_i)$ are interchangeable under the condition $\mathbf{\Delta}(\mathcal{E}(t))$, which we required in the statement of the lemma.

- If τ is a variable symbol different from the x_i's, then it cannot be x. Hence both terms are just the variable symbol τ.

- $\tau \equiv f(\tau_1, \tau_2, \ldots, \tau_m)$ We have to show that $f\left([\mathcal{E}(t)/x] \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_1, \ldots, [\mathcal{E}(t)/x] \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_m\right)$ is interchangeable with $f\left(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \ldots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_1, \ldots\right)$ under the condition $\mathbf{\Delta}(\mathcal{E}(t))$, which is easy using the ERf2 rule and induction on all the τ_i.

- $\tau \equiv \iota y_\psi(\varphi)$ where $y \notin \{x_1, x_2, \ldots, x_n\}$ We have to show that $[\mathcal{E}(t)/x] \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau$, i.e.,

  $$[\mathcal{E}(t)/x]\, \iota y_{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1)\ldots \\ x_1 \end{smallmatrix} \right]\!\!\right]\psi\right)\&\forall y\left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1)\ldots \\ x_1 \end{smallmatrix} \right]\!\!\right]\varphi\right)\right)\&\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1)\ldots \\ x_1 \end{smallmatrix} \right]\!\!\right]\psi} \left(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)$$

  is interchangeable with

  $$\iota y_{\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1)\ldots \\ x_1 \end{smallmatrix} \right]\!\!\right]\psi\right)\&\forall y\left(\mathbf{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1)\ldots \\ x_1 \end{smallmatrix} \right]\!\!\right]\varphi\right)\right)\&\left[\!\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1)\ldots \\ x_1 \end{smallmatrix} \right]\!\!\right]\psi} \left(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \ldots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)$$

  We will abbreviate the latter term as $\iota x_{\psi_2}(\varphi_2)$.

  Using corollary 57, it is not difficult to derive the uniqueness conditions for both terms.

  - $x \equiv y$ or x is not a free variable of $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi$

    We first remark that $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi \equiv \left[\!\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \ldots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi$.
    Indeed, in case $x \equiv y$, remark that for the substitution $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \iota x_\psi(\varphi)$ to be defined, either x_i is not a free variable

of φ (and then Lemma 59 yields that substituting either $\mathcal{E}([t/x]t_i)$ or $\mathcal{E}(t_i)$ for x_i does not change the resulting formula) or x is not a free variable of t_i, and then $\mathcal{E}([t/x]t_i) \equiv \mathcal{E}(t_i)$.

In the other case, x is not a free variable of $\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \varphi$; it is easy to see that both terms are identical.

Next, we investigate the first term to consider, $[\mathcal{E}(t)/x] \left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \tau$. Its form depends on whether x is a free variable of $\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \psi$. In case x is not free in $\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \psi$, the first term becomes

$$\iota y_{\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots \\ x_1 & \end{smallmatrix} \right]\!\right] \psi\right) \& \forall y \left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots \\ x_1 & \end{smallmatrix} \right]\!\right] \varphi\right)\right) \& \left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots \\ x_1 & \end{smallmatrix} \right]\!\right] \psi} \left(\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \varphi \right)$$

and again it is easy to see that $\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \psi$ is identical to $\left[\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \dots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \psi$ and hence both terms to consider are identical.

If x is a free variable of $\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \psi$, the first term is

$$\iota y_{\mathbf{\Delta}(\mathcal{E}(t)) \& [\mathcal{E}(t)/x] \mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots \\ x_1 & \end{smallmatrix} \right]\!\right] \psi\right) \& \forall y \left(\mathbf{\Delta}\left(\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots \\ x_1 & \end{smallmatrix} \right]\!\right] \varphi\right)\right) \& [\mathcal{E}(t)/x] \left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots \\ x_1 & \end{smallmatrix} \right]\!\right] \psi} \left(\left[\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \dots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \varphi \right)$$

which we will abbreviate as $\iota y_{\Psi_1}(\varphi_1)$ with $\Psi_1 \equiv \mathbf{\Delta}(\mathcal{E}(t)) \,\&\, [\mathcal{E}(t)/x] \mathbf{\Delta}(\psi_1) \,\&\, \forall y(\mathbf{\Delta}(\varphi_1)) \,\&\, [\mathcal{E}(t)/x] \psi_1$.

We already established that φ_1 and φ_2 are identical. Hence, in order to apply theorem 29, we need the uniqueness conditions of both ι-terms (which we already have), that y be not free in t (see below) and that Ψ_1 and ψ_2 be interchangeable under $\mathbf{\Delta}(\mathcal{E}(t))$:

$$\begin{aligned}
&\vdash_\iota \mathbf{\Delta}(\Psi_1) && \text{defAnt} \\
\Psi_1 &\vdash_\iota \Psi_1 && \text{ass} \\
\Psi_1 &\vdash_\iota \forall y(\mathbf{\Delta}(\varphi_1)) \,\&\, [\mathcal{E}(t)/x] \psi_1 && \&\text{-elim} \\
\Psi_1 &\vdash_\iota [\mathcal{E}(t)/x] \psi_1 && \&\text{-elim} \\
\mathbf{\Delta}(\mathcal{E}(t)) \,;\, \mathbf{\Delta}([\mathcal{E}(t)/x] \psi_1) \,\&\, [\mathcal{E}(t)/x] \psi_1 &\vdash_\iota \left[\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \dots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \psi && \text{induction} \\
\mathbf{\Delta}(\mathcal{E}(t)) \,,\, \mathbf{\Delta}([\mathcal{E}(t)/x] \psi_1) \,;\, [\mathcal{E}(t)/x] \psi_1 &\vdash_\iota \left[\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \dots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \psi && \text{toCtxt} \\
\mathbf{\Delta}(\mathcal{E}(t)) \,;\, \Psi_1 &\vdash_\iota \left[\!\left[\begin{smallmatrix} \mathcal{E}([t/x]t_1) & \dots & \mathcal{E}([t/x]t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\right] \psi && \text{Cut3}
\end{aligned}$$

$$\begin{array}{rl}
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\Psi_1 \vdash_\iota \boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,([t/x]\,t_1) & \dots & \mathcal{E}\,([t/x]\,t_n)\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\psi\Big) & \text{defCons}\\
\Psi_1 \vdash_\iota \forall y(\boldsymbol{\Delta}(\varphi_1)) & \text{\&-elim}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\Psi_1 \vdash_\iota \forall y(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,([t/x]\,t_1) & \dots\\ x_1 & \end{smallmatrix}\Big]\!\Big]\,\psi & \text{\&-intro}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\Psi_1 \vdash_\iota \psi_2 & \text{\&-intro}
\end{array}$$

$$\begin{array}{rl}
\vdash_\iota \boldsymbol{\Delta}(\psi_2) & \text{defAnt}\\
\psi_2 \vdash_\iota \psi_2 & \text{ass}\\
\psi_2 \vdash_\iota \forall y(\boldsymbol{\Delta}(\varphi_2)) \,\&\, \Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,([t/x]\,t_1) & \dots\\ x_1 & \end{smallmatrix}\Big]\!\Big]\,\psi & \text{\&-elim}\\
\psi_2 \vdash_\iota \Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,([t/x]\,t_1) & \dots & \mathcal{E}\,([t/x]\,t_n)\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\psi & \text{\&-elim}
\end{array}$$

$$\begin{array}{rl}
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,([t/x]\,t_1) & \dots\\ x_1 & \end{smallmatrix}\Big]\!\Big]\,\psi\Big)\\
\quad\&\,\Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,([t/x]\,t_1) & \dots & \mathcal{E}\,([t/x]\,t_n)\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\psi \vdash_\iota [\mathcal{E}\,(t)/x]\,\psi_1 & \text{induction}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,,\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,([t/x]\,t_1) & \dots\\ x_1 & \end{smallmatrix}\Big]\!\Big]\,\psi\Big)\,;\\
\quad\Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,([t/x]\,t_1) & \dots & \mathcal{E}\,([t/x]\,t_n)\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\psi \vdash_\iota [\mathcal{E}\,(t)/x]\,\psi_1 & \text{toCtxt}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\psi_2 \vdash_\iota [\mathcal{E}\,(t)/x]\,\psi_1 & \text{Cut3}\\
\psi_2 \vdash_\iota \forall y(\boldsymbol{\Delta}(\varphi_2)) & \text{\&-elim}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\psi_2 \vdash_\iota \forall y(\boldsymbol{\Delta}(\varphi_2)) \,\&\, [\mathcal{E}\,(t)/x]\,\psi_1 & \text{\&-intro}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\psi_2 \vdash_\iota \boldsymbol{\Delta}([\mathcal{E}\,(t)/x]\,\psi_1) & \text{defCons}\\
\boldsymbol{\Delta}([\mathcal{E}\,(t)/x]\,\psi_1) \vdash_\iota [\mathcal{E}\,(t)/x]\,\boldsymbol{\Delta}(\psi_1) & \text{Th. 33}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\psi_2 \vdash_\iota [\mathcal{E}\,(t)/x]\,\boldsymbol{\Delta}(\psi_1) & \text{Cut}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\psi_2 \vdash_\iota [\mathcal{E}\,(t)/x]\,\boldsymbol{\Delta}(\psi_1) \,\&\, \forall y(\boldsymbol{\Delta}(\varphi_2))\\
\qquad\qquad \&\, [\mathcal{E}\,(t)/x]\,\psi_1 & \text{\&-intro}\\
\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\mathcal{E}\,(t))) & \text{Ddef}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t)) \vdash_\iota \boldsymbol{\Delta}(\mathcal{E}\,(t)) & \text{ass}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\vdash_\iota \boldsymbol{\Delta}(\mathcal{E}\,(t)) & \text{toCtxt}\\
\boldsymbol{\Delta}(\mathcal{E}\,(t))\,;\psi_2 \vdash_\iota \Psi_1 & \text{\&-intro}
\end{array}$$

– If $x \not\equiv y$ and x is a free variable of $\Big[\!\Big[\begin{smallmatrix}\mathcal{E}(t_1) & \dots & \mathcal{E}(t_n)\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi$, then the first term, $[\mathcal{E}\,(t)/x]\Big[\!\Big[\begin{smallmatrix}\mathcal{E}(t_1) & \dots & \mathcal{E}(t_n)\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\tau$, is

$$\iota y_{\substack{\boldsymbol{\Delta}(\mathcal{E}(t))\&[\mathcal{E}(t)/x]\boldsymbol{\Delta}\left(\left[\!\left[\begin{smallmatrix}\mathcal{E}(t_1)\dots\mathcal{E}(t_n)\\ x_1\dots x_n\end{smallmatrix}\right]\!\right]\psi\right)\\ \&[\mathcal{E}(t)/x]\forall y\left(\boldsymbol{\Delta}\left(\left[\!\left[\begin{smallmatrix}\mathcal{E}(t_1)\dots\mathcal{E}(t_n)\\ x_1\dots x_n\end{smallmatrix}\right]\!\right]\varphi\right)\right)\&[\mathcal{E}(t)/x]\left[\!\left[\begin{smallmatrix}\mathcal{E}(t_1)\dots\mathcal{E}(t_n)\\ x_1\dots x_n\end{smallmatrix}\right]\!\right]\psi}} \left([\mathcal{E}\,(t)/x]\Big[\!\Big[\begin{smallmatrix}\mathcal{E}\,(t_1) & \dots & \mathcal{E}\,(t_n)\\ x_1 & & x_n\end{smallmatrix}\Big]\!\Big]\,\varphi\right)$$

For the substitution $[\mathcal{E}(t)/x]\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n)\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau$ to be defined, y must not be free in t, and hence the proof is analogous to the first case.

- The other cases are easy.

□

Theorem 69 *1. For any formula α and term t of the* PITFOL' *calculus for which the uniqueness conditions of the expansion all of its ι-terms are derivable in the* PITFOL' *calculus, if the substitution $[\mathcal{E}(t)/x]\mathcal{E}(\alpha)$ is defined, then $\mathcal{E}([t/x]\alpha)$ is interchangeable with $[\mathcal{E}(t)/x]\mathcal{E}(\alpha)$ under the condition $\mathbf{\Delta}(\mathcal{E}(t))$, and likewise for terms τ of the* PITFOL' *calculus.*

2. For any formula β of the PITFOL *calculus for which the uniqueness conditions are derivable, and any terms $t, t_1, \ldots, t_n$ of the* PITFOL' *calculus, if $x \notin FV(\beta) \setminus \{x_1, x_2, \ldots, x_n\}$, and the substitutions $[\mathcal{E}(t)/x]\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n)\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\beta$ are defined, then $\left[\!\!\left[\begin{smallmatrix}\mathcal{E}([t/x]t_1) & \ldots & \mathcal{E}([t/x]t_n)\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\beta$ is interchangeable with $[\mathcal{E}(t)/x]\left[\!\!\left[\begin{smallmatrix}\mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n)\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\beta$ under the condition $\mathbf{\Delta}(\mathcal{E}(t))$, and likewise for terms σ of the* PITFOL *calculus.*

Proof.

We prove both parts simultaneously by induction on $\mathrm{cpl}(\alpha)$ resp. $\mathrm{cpl}(\tau)$ and $\sum_i \mathrm{cpl}(t_i)$.

For the base case, we have

- $\mathrm{cpl}(\tau) = 0$, i.e., τ is a variable symbol. Then either $\tau \equiv x$ and both terms are $\mathcal{E}(t)$ or $\tau \not\equiv x$ and both terms are τ.

- $\mathrm{cpl}(t_i) = 0$ for $i = 1, 2, \ldots, n$. We can apply the previous lemma if the first part of this theorem holds for $\tau \equiv t_i$, which is what we just proved.

Next we handle the induction step.

For the first part, the possible shapes α and τ can take are

- $\tau \equiv f(\tau_1, \tau_2, \ldots, \tau_m)$ Easy using induction on the τ_i and the ERf2 rule.

- $\tau \equiv \iota y_\psi(\varphi)$

– If $x \equiv y$ or x is not a free variable of φ, then we have the following situation. Either x is not a free variable of ψ, and both terms are are $\iota y_{\mathcal{E}(\psi)}(\mathcal{E}(\varphi))$. Else, we have to show the interchangeability of $\iota y_{\mathcal{E}(\boldsymbol{\Delta}(t))\&\mathcal{E}([t/x]\psi)}(\mathcal{E}(\varphi))$ and $\iota y_{\boldsymbol{\Delta}(\mathcal{E}(t))\&[\mathcal{E}(t)/x]\mathcal{E}(\psi)}(\mathcal{E}(\varphi))$ under the condition $\boldsymbol{\Delta}(\mathcal{E}(t))$. Using theorem 67.1 and induction, it is not difficult to show that the domain formulae of both terms are interchangeable.

– If $x \not\equiv y$ and x is a free variable of φ, then we have to show the interchangeability of $\iota y_{\mathcal{E}(\boldsymbol{\Delta}(t))\&\mathcal{E}([t/x]\psi)}(\mathcal{E}([t/x]\varphi))$ and $\iota y_{\boldsymbol{\Delta}(\mathcal{E}(t))\&[\mathcal{E}(t)/x]\mathcal{E}(\psi)}([\mathcal{E}(t)/x]\mathcal{E}(\varphi))$ under the condition $\boldsymbol{\Delta}(\mathcal{E}(t))$. It is again easy to show that the domain formulae of both terms are interchangeable, but the definiens are only interchangeable under the condition $\boldsymbol{\Delta}(\mathcal{E}(t))$. Note that for the substitution to be defined, y must not be free in t. We choose the variable symbol z different from y and not occurring in t or ψ:

$$\begin{aligned}
\mathcal{E}(\psi) &\vdash_\iota \exists!y(\mathcal{E}(\varphi)) && \text{UC}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [\mathcal{E}(t)/x]\mathcal{E}(\psi) &\vdash_\iota \exists!y([\mathcal{E}(t)/x]\mathcal{E}(\varphi)) && \text{subst}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; \mathcal{E}([t/x]\psi) &\vdash_\iota \exists!y(\mathcal{E}([t/x]\varphi)) && \text{Th. 29}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [\mathcal{E}(t)/x]\mathcal{E}(\psi) &\vdash_\iota \exists y([\mathcal{E}(t)/x]\mathcal{E}(\varphi)) && \&\text{-elim}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [z/y][\mathcal{E}(t)/x]\mathcal{E}(\psi) &\vdash_\iota \exists y([\mathcal{E}(t)/x]\mathcal{E}(\varphi)) && \text{subst}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [z/y][\mathcal{E}(t)/x]\mathcal{E}(\psi) &\vdash_\iota \forall y(\boldsymbol{\Delta}([\mathcal{E}(t)/x]\mathcal{E}(\varphi))) && \text{defCons}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [z/y][\mathcal{E}(t)/x]\mathcal{E}(\psi) &\vdash_\iota \boldsymbol{\Delta}([\mathcal{E}(t)/x]\mathcal{E}(\varphi)) && \forall\text{-elim}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; \mathcal{E}([t/x]\varphi) &\vdash_\iota \exists y(\mathcal{E}([t/x]\varphi)) && \&\text{-elim}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [z/y]\mathcal{E}([t/x]\varphi) &\vdash_\iota \exists y(\mathcal{E}([t/x]\varphi)) && \text{subst}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [z/y]\mathcal{E}([t/x]\varphi) &\vdash_\iota \forall y(\boldsymbol{\Delta}(\mathcal{E}([t/x]\varphi))) && \text{defCons}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [z/y]\mathcal{E}([t/x]\varphi) &\vdash_\iota \boldsymbol{\Delta}(\mathcal{E}([t/x]\varphi)) && \forall\text{-elim}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; \boldsymbol{\Delta}([\mathcal{E}(t)/x]\mathcal{E}(\varphi)) \,\&\, [\mathcal{E}(t)/x]\mathcal{E}(\varphi) &\vdash_\iota \mathcal{E}([t/x]\varphi) && \text{induction}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) , \boldsymbol{\Delta}([\mathcal{E}(t)/x]\mathcal{E}(\varphi)) ; [\mathcal{E}(t)/x]\mathcal{E}(\varphi) &\vdash_\iota \mathcal{E}([t/x]\varphi) && \text{toCtxt}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) , [z/y][\mathcal{E}(t)/x]\mathcal{E}(\psi) ; [\mathcal{E}(t)/x]\mathcal{E}(\varphi) &\vdash_\iota \mathcal{E}([t/x]\varphi) && \text{CutCtxt}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) , [z/y][\mathcal{E}(t)/x]\mathcal{E}(\psi) ; &\vdash_\iota [\mathcal{E}(t)/x]\mathcal{E}(\varphi) \Rightarrow \mathcal{E}([t/x]\varphi) && \text{DdRu2}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [z/y][\mathcal{E}(t)/x]\mathcal{E}(\psi) &\vdash_\iota [\mathcal{E}(t)/x]\mathcal{E}(\varphi) \Rightarrow \mathcal{E}([t/x]\varphi) && \text{fromCtxt}\\
&\vdash_\iota \boldsymbol{\Delta}(\boldsymbol{\Delta}(\mathcal{E}(t))) && \text{Ddef}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; \boldsymbol{\Delta}(\mathcal{E}(t)) , [z/y][\mathcal{E}(t)/x]\mathcal{E}(\psi) &\vdash_\iota [\mathcal{E}(t)/x]\mathcal{E}(\varphi) \Rightarrow \mathcal{E}([t/x]\varphi) && \text{Weak}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; \boldsymbol{\Delta}(\mathcal{E}(t)) \,\&\, [z/y][\mathcal{E}(t)/x]\mathcal{E}(\psi) &\vdash_\iota [\mathcal{E}(t)/x]\mathcal{E}(\varphi) \Rightarrow \mathcal{E}([t/x]\varphi) && \text{AnU}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; \boldsymbol{\Delta}(\mathcal{E}([t/x]\varphi)) \,\&\, \mathcal{E}([t/x]\varphi) &\vdash_\iota [\mathcal{E}(t)/x]\mathcal{E}(\varphi) && \text{induction}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) , \boldsymbol{\Delta}(\mathcal{E}([t/x]\varphi)) ; \mathcal{E}([t/x]\varphi) &\vdash_\iota [\mathcal{E}(t)/x]\mathcal{E}(\varphi) && \text{toCtxt}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) , [z/y]\mathcal{E}([t/x]\psi) ; \mathcal{E}([t/x]\varphi) &\vdash_\iota [\mathcal{E}(t)/x]\mathcal{E}(\varphi) && \text{CutCtxt}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) , [z/y]\mathcal{E}([t/x]\psi) ; &\vdash_\iota \mathcal{E}([t/x]\varphi) \Rightarrow [\mathcal{E}(t)/x]\mathcal{E}(\varphi) && \text{DdRu2}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; [z/y]\mathcal{E}([t/x]\psi) &\vdash_\iota \mathcal{E}([t/x]\varphi) \Rightarrow [\mathcal{E}(t)/x]\mathcal{E}(\varphi) && \text{fromCtxt}\\
\boldsymbol{\Delta}(\boldsymbol{\Delta}(\mathcal{E}(t))) &\vdash_\iota \boldsymbol{\Delta}\big(\mathcal{E}\big(\boldsymbol{\Delta}'(t)\big)\big) && \text{Th. 67.1}\\
&\vdash_\iota \boldsymbol{\Delta}\big(\mathcal{E}\big(\boldsymbol{\Delta}'(t)\big)\big) && \text{Cut}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; \mathcal{E}\big(\boldsymbol{\Delta}'(t)\big) , [z/y]\mathcal{E}([t/x]\psi) &\vdash_\iota \mathcal{E}([t/x]\varphi) \Rightarrow [\mathcal{E}(t)/x]\mathcal{E}(\varphi) && \text{Weak}\\
\boldsymbol{\Delta}(\mathcal{E}(t)) ; \mathcal{E}\big(\boldsymbol{\Delta}'(t)\big) \,\&\, [z/y]\mathcal{E}([t/x]\psi) &\vdash_\iota \mathcal{E}([t/x]\varphi) \Rightarrow [\mathcal{E}(t)/x]\mathcal{E}(\varphi) && \text{AnU}
\end{aligned}$$

$$\begin{array}{rl}
\mathbf{\Delta}(\mathcal{E}(t)); [z/y]\psi_1, [z/y]\psi_2 \vdash_\iota \mathcal{E}([t/x]\varphi) \Leftrightarrow [\mathcal{E}(t)/x]\mathcal{E}(\varphi) & \&\text{-intro} \\
\mathbf{\Delta}(\mathcal{E}(t)); [z/y]\psi_1, [z/y]\psi_2 \vdash_\iota \forall y(\mathcal{E}([t/x]\varphi) \Leftrightarrow [\mathcal{E}(t)/x]\mathcal{E}(\varphi)) & \forall\text{-intro} \\
\mathbf{\Delta}(\mathcal{E}(t)); [z/y]\psi_1, [z/y]\psi_2 \vdash_\iota \iota y_{[z/y]\psi_1}(\mathcal{E}([t/x]\varphi)) & \\
\qquad = \iota y_{[z/y]\psi_2}([\mathcal{E}(t)/x]\mathcal{E}(\varphi)) & \text{CtxtEq-}\iota \\
\mathbf{\Delta}(\mathcal{E}(t)); \psi_1, \psi_2 \vdash_\iota \iota y_{\psi_1}(\mathcal{E}([t/x]\varphi)) & \\
\qquad = \iota y_{\psi_2}([\mathcal{E}(t)/x]\mathcal{E}(\varphi)) & \text{subst}
\end{array}$$

where we used the abbreviations $\psi_1 \equiv \mathcal{E}(\mathbf{\Delta}'(t)) \,\&\, \mathcal{E}([t/x]\psi)$ and $\psi_2 \equiv \mathbf{\Delta}(\mathcal{E}(t)) \,\&\, [\mathcal{E}(t)/x]\mathcal{E}(\psi)$

Using the interchangeability of ψ_1 and ψ_2, the required sequents easily follow.

- $\tau \equiv g(\tau_1, \tau_2, \ldots, \tau_m)$ where g is a defined function symbol We have to show that $\mathcal{E}(g([t/x]\tau_1, \ldots, [t/x]\tau_m))$, i.e., $\left[\!\!\left[{}^{\mathcal{E}([t/x]\tau_1)}_{x_1} \ldots {}^{\mathcal{E}([t/x]\tau_m)}_{x_m} \right]\!\!\right] \mathcal{E}\left(\widetilde{g^*}\right)$, is interchangeable with $[\mathcal{E}(t)/x]\left[\!\!\left[{}^{\mathcal{E}(\tau_1)}_{x_1} \ldots {}^{\mathcal{E}(\tau_m)}_{x_m} \right]\!\!\right] \mathcal{E}\left(\widetilde{g^*}\right)$ under the condition $\mathbf{\Delta}(\mathcal{E}(t))$, which we can obtain by induction, since indeed $\sum_i(\mathrm{cpl}(\tau_i)) < \mathrm{cpl}(\tau) = 1 + \sum_i \mathrm{cpl}(\tau_i)$.
- The other cases are easy.

For the second part, we again apply the previous lemma, for which we need the first part to hold for all t_i. We have obviously $\mathrm{cpl}(t_i) \leq \sum_i \mathrm{cpl}(t_i)$. If the inequality is strict, we can apply induction; if both values are equal, we can use the first part which we just proved for $\mathrm{cpl}(\tau) = \sum_i \mathrm{cpl}(t_i)$. □

5.4 Equiconsistency

We define the expansion $\mathcal{E}(L)$ of a list of formulae $L := \alpha_1, \alpha_2, \ldots, \alpha_n$ as the list of its expansions $\mathcal{E}(\alpha_1), \mathcal{E}(\alpha_2), \ldots, \mathcal{E}(\alpha_n)$. We also define the expansion of a PITFOL' sequent $\Sigma; \Gamma \vdash_{\iota'} \alpha$ as the PITFOL sequent $\mathcal{E}(\Sigma); \mathcal{E}(\Gamma) \vdash_\iota \mathcal{E}(\alpha)$.

As already announced, we will show that for each proof of $\Sigma; \Gamma \vdash_{\iota'} \alpha$, we can derive $\mathcal{E}(\Sigma); \mathcal{E}(\Gamma) \vdash_\iota \mathcal{E}(\alpha)$. We prove this by induction on the length of the proof.

For the base case, a proof of length 1 is either an application of the ass rule with $\mathbf{\Delta}'(\alpha) \equiv \top$, the eq rule with $\mathbf{\Delta}'(t) \equiv \top$ or a definition. These cases are handled analogously to the corresponding cases of the induction step below.

For the induction step, we examine the rule used to obtain the last sequent of the proof.

5.4.1 Expansion of a definition

We will only consider definition of function symbols explicitly; definition of predicate symbols is handled analogously. By induction, we have $\mathcal{E}(UC(t_1))$, $\mathcal{E}(UC(t_2))$ and so on, and we required also $\mathcal{E}(UC(g^*))$ to be derivable. It is easy to see that $\mathcal{E}(\psi \vdash_\iota \exists!x(\varphi)) \equiv \mathcal{E}(\psi) \vdash_\iota \exists!x(\mathcal{E}(\varphi))$, so we actually have $\mathcal{E}(UC(\alpha)) \equiv UC(\mathcal{E}(\alpha))$. From $UC(\mathcal{E}(g^*))$ we easily obtain $UC(\mathcal{E}(\widetilde{g^*}))$. Hence, by applying corollary 57, we have $UC(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}(\widetilde{g^*}))$ and we can apply the eq rule, yielding

$$\boldsymbol{\Delta}\left(\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}(\widetilde{g^*})\right) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}(\widetilde{g^*}) = \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}(\widetilde{g^*})$$

Using theorem 67.2 and theorem 67.1, we get

$$\mathcal{E}\left(\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right)\right) \vdash_\iota \left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \cdots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}(\widetilde{g^*}) = \mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right)$$

which is by definition

$$\mathcal{E}(\boldsymbol{\Delta}'(g(t_1, t_2, \ldots, t_n))) \vdash_\iota \mathcal{E}(g(t_1, t_2, \ldots, t_n)) = \mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right),$$

the required sequent.

5.4.2 Expansion of the proposition rules

Since $\mathcal{E}(\alpha \,\&\, \beta) \equiv \mathcal{E}(\alpha) \& \mathcal{E}(\beta)$ and $\mathcal{E}(\neg\alpha) \equiv \neg\mathcal{E}(\alpha)$, the proof for these rules is trivial.

Only the assumption rule is somewhat more complicated and we will handle it explicitly. By induction, we have $\mathcal{E}(UC(\alpha))$ and $\mathcal{E}(\Sigma) ; \vdash_\iota \mathcal{E}(\boldsymbol{\Delta}'(\alpha))$. Using theorem 67.1, we obtain $\mathcal{E}(\Sigma) ; \vdash_\iota \boldsymbol{\Delta}(\mathcal{E}(\alpha))$ and we are able to apply the assumption rule of the PITFOL calculus, yielding the required sequent.

5.4.3 Expansion of the predicate rules

Since $\mathcal{E}(\forall x(\alpha)) \equiv \forall x(\mathcal{E}(\alpha))$, the proof for the $\forall$-intro and $\forall$-elim rules is trivial.

For the equality and substitution rules, we only handle the subst rule explicitly. By induction, we have $\mathcal{E}(UC(t))$ and $\mathcal{E}(\Sigma) ; \mathcal{E}(\Gamma) \vdash_\iota \mathcal{E}(\alpha)$. Applying the subst rule yields

$$\boldsymbol{\Delta}(\mathcal{E}(t)) , [\mathcal{E}(t)/x]\mathcal{E}(\Sigma) ; [\mathcal{E}(t)/x]\mathcal{E}(\Gamma) \vdash_\iota [\mathcal{E}(t)/x]\mathcal{E}(\alpha) .$$

Using theorems 67.1 and 69.1, we obtain the required sequent

$$\mathcal{E}\left(\mathbf{\Delta}'(t)\right), \mathcal{E}\left([t/x]\,\Sigma\right); \mathcal{E}\left([t/x]\,\Gamma\right) \vdash_\iota \mathcal{E}\left([t/x]\,\alpha\right).$$

5.4.4 Expansion of the other rules

For the iota rule, we are given $\mathcal{E}(\psi \vdash_\iota \exists!x(\varphi))$. As noted above, this is identical to $\mathcal{E}(\psi) \vdash_\iota \exists!x(\mathcal{E}(\varphi))$ and the iota rule of the PITFOL calculus yields $\mathcal{E}(\psi) \vdash_\iota [\mathcal{E}(\iota x_\psi(\varphi))/x]\widetilde{\mathcal{E}(\varphi)}$. Using theorem 69.1, we easily get the required sequent $\mathcal{E}(\psi) \vdash_\iota \mathcal{E}([\iota x_\psi(\varphi)/x]\,\widetilde{\varphi})$.

The other rules are analogous.

5.5 Derived rules

Most derived rules of the PITFOL calculus still work unmodified in the PITFOL' calculus; only those rules that depend on the exact form of $\mathbf{\Delta}(\alpha)$ or $\mathbf{\Delta}(t)$ need to be reconsidered.

5.5.1 Ddef rule

We prove this by induction on $\rho(\alpha)$ and $\rho(t)$. For the base case, α resp. t do not contain defined symbols and we can use the old Ddef rule unmodified.

We handle the induction step by structural induction. The only new cases are

- $t \equiv g(t_1, t_2, \ldots, t_n)$ where g is the j-th defined function symbol. We have to derive $\vdash_{\iota'} \mathbf{\Delta}'(\mathbf{\Delta}'(g(t_1, t_2, \ldots, t_n)))$, i.e, $\vdash_{\iota'} \mathbf{\Delta}'\Big(\mathbf{\Delta}'\Big(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big]\, g^*\Big)\Big)$. From corollary 64, we see that we can apply induction on $\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big]\, g^*$ to get the required sequent.
- $t \equiv q(t_1, t_2, \ldots, t_n)$ where q is the j-th defined predicate symbol. This case is analogous.

All further derived rules up to ERf2 can be copied unmodified from the PITFOL' calculus. For ERp and ERp2, p is allowed to be a defined predicate symbol; for ERf and ERf2, f is allowed to be a defined function symbol.

Theorem 23 also applies to the PITFOL' calculus, but since the definition of $\mathbf{\Delta}'$ also uses simultaneous substitution, we have to prove a variant of property 55 and also lemmas 60, 58 and 59 at the same time, so what we will prove is

Theorem 70 *1. Given*

$$\begin{cases} \boldsymbol{\Delta}'(\alpha)\,;\alpha \vdash_{\iota'} \beta \\ \boldsymbol{\Delta}'(\alpha)\,;\beta \vdash_{\iota'} \alpha \end{cases}$$

then

$$\begin{cases} \boldsymbol{\Delta}'(A(\alpha))\,;A(\alpha) \vdash_{\iota'} A(\beta) \\ \boldsymbol{\Delta}'(A(\alpha))\,;A(\beta) \vdash_{\iota'} A(\alpha) \end{cases}$$

where $A(\alpha)$ is a formula possibly containing α and $A(\beta)$ is the same formula where a number of instances of α are replaced by β, and

$$\boldsymbol{\Delta}'(t(\alpha)) \vdash_{\iota'} t(\alpha) = t(\beta)$$

where $t(\alpha)$ is a term possibly containing α and $t(\beta)$ is the same term where a number of instances of α are replaced by β.

These results only hold if the uniqueness conditions for $A(\alpha)$, resp. $t(\alpha)$ can be derived.

2. Given

$$\boldsymbol{\Delta}'(t_1) \vdash_{\iota'} t_1 = t_2$$

then analogously,

$$\begin{cases} \boldsymbol{\Delta}'(A(t_1))\,;A(t_1) \vdash_{\iota'} A(t_2) \\ \boldsymbol{\Delta}'(A(t_1))\,;A(t_2) \vdash_{\iota'} A(t_1) \\ \qquad\qquad \boldsymbol{\Delta}'(t(t_1)) \vdash_{\iota'} t(t_1) = t(t_2) \end{cases}$$

These results only hold if the uniqueness conditions for $A(t_1)$, resp. $t(t_1)$ can be derived.

3. Given a formula α and the formula $\widetilde{\alpha}$ obtained from α by renaming all its bound variables (as in the statement of the ι-rule). Then

$$\begin{cases} \boldsymbol{\Delta}'(\alpha)\,;\alpha \vdash_{\iota'} \widetilde{\alpha} \\ \boldsymbol{\Delta}'(\alpha)\,;\widetilde{\alpha} \vdash_{\iota'} \alpha \end{cases}$$

Given a term t and the term $\widetilde{t}$ obtained from t by renaming all its bound variables. Then

$$\boldsymbol{\Delta}'(t) \vdash_{\iota'} t = \widetilde{t}$$

These results only hold if the uniqueness conditions for α, resp. t can be derived.

4. *If* $\begin{cases} \boldsymbol{\Delta}'(t_1) \vdash_{\iota'} t_1 = s_1 \\ \boldsymbol{\Delta}'(t_2) \vdash_{\iota'} t_2 = s_2 \\ \qquad \cdots \\ \boldsymbol{\Delta}'(t_n) \vdash_{\iota'} t_n = s_n \end{cases}$
and the uniqueness conditions of $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha$ *and* $\left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha$, *resp.* $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \tau$ *and* $\left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \tau$ *are derivable, then for formulae* α *resp. terms* τ, *we have*

$$\begin{cases} \boldsymbol{\Delta}'\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha\right); \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha \vdash_{\iota'} \left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha \\ \boldsymbol{\Delta}'\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha\right); \left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha \vdash_{\iota'} \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha \end{cases}$$

resp.

$$\boldsymbol{\Delta}'\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \tau\right) \vdash_{\iota'} \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \tau = \left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \tau$$

if the simultaneous substitutions are defined.

If moreover t_i *is interchangeable with* s_i *for all* i, *then it is not necessary that the uniqueness conditions of* $\left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha$ *resp.* $\left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \tau$ *be given.*

5. *If the uniqueness conditions of* t_1, t_2, $\ldots$, t_n, s_1, s_2, $\ldots$, s_m *and* α *are derivable,* $\{x_1, x_2, \ldots, x_n\} \cap (FV(\alpha) \setminus \{y_1, y_2, \ldots, y_m\}) = \emptyset$ *and the substitutions* $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\right] \alpha$ *are defined, then this formula is interchangeable with* $\left[\!\left[\begin{smallmatrix} \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] s_1 & \cdots & \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\right] \alpha$ *when the uniqueness conditions of one of both are derivable.*

The analogous property for terms τ *of the* PITFOL' *calculus also holds.*

6. *If the uniqueness conditions of* t_1, t_2, $\ldots$, t_n *and* α *are derivable, and the substitution* $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha$ *is defined, then so is the substitution* $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \boldsymbol{\Delta}(\alpha)$ *and* $\boldsymbol{\Delta}\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha\right) \vdash_{\iota} \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \boldsymbol{\Delta}(\alpha)$. *Likewise for terms* τ.

7. *If the uniqueness conditions of* t_1, t_2, $\ldots$, t_n *and* α *are derivable,* x_i *is not a free variable of* α, *and one of the substitutions* $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\right] \alpha$ *and*

$\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t'_i & t_{i+1} & \ldots & t_n \\ x_1 & & x_{i-1} & x_i & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$ *is defined, then both formulae are identical and interchangeable with* $\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_{i-1} & t_{i+1} & \ldots & t_n \\ x_1 & & x_{i-1} & x_{i+1} & & x_n \end{smallmatrix} \right]\!\!\right] \alpha$.

The analogous property for terms τ *of the* PITFOL' *calculus also holds.*

Proof.

We prove the theorem by induction on $\rho(A(\alpha))$ and $\rho(t(\alpha))$ for the first part, on $\rho(A(t_1))$ and $\rho(t(t_1))$ for the second part, and on $\rho(\alpha)$ and $\rho(\tau)$ for the other parts.

For the base case, no defined symbols are present in $A(\alpha)$, $t(\alpha)$, ... and we can prove the first three parts in the same way as theorem 23, the fifth part in the same way as lemma 60, the sixth part in the same way as lemma 58 and the seventh part in the same way as lemma 59. (Note that β in part 1, t_2 in part 2, $t_1, \ldots, t_n, s_1, \ldots, s_m$ in part 5 and $t_1, \ldots, t_n$ in part 6 and 7 still may contain defined symbols but this does not affect the proofs.) The fourth part is proved by structural induction. Most cases are easy; we only mention

- $\tau \equiv \iota x_\psi(\varphi)$ Suppose that x is not one of $x_1, x_2, \ldots, x_n$ (the other case is analogous). Then we have to derive

$$\begin{aligned} &\boldsymbol{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \,\&\, \forall x(\boldsymbol{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)) \,\&\, \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi \\ &\quad \vdash_{\iota'} \iota x_{\boldsymbol{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \,\&\, \forall x\left(\boldsymbol{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \,\&\, \left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi} \left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right) \\ &\quad = \iota x_{\boldsymbol{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi\right) \,\&\, \forall x\left(\boldsymbol{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} s_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right)\right) \,\&\, \left[\!\!\left[\begin{smallmatrix} s_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi} \left(\left[\!\!\left[\begin{smallmatrix} s_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \varphi\right) \end{aligned}$$

which we will abbreviate as

$$\begin{aligned} \boldsymbol{\Delta'}(\Psi_1) \,\&\, \forall x(\boldsymbol{\Delta'}(\Phi_1)) \,\&\, \Psi_1 \vdash_{\iota'} \iota x_{\boldsymbol{\Delta'}(\Psi_1)\&\forall x(\boldsymbol{\Delta'}(\Phi_1))\&\Psi_1}(\Phi_1) \\ = \iota x_{\boldsymbol{\Delta'}(\Psi_2)\&\forall x(\boldsymbol{\Delta'}(\Phi_2))\&\Psi_2}(\Phi_2). \end{aligned}$$

The derivation is as follows, where we further abbreviate $\boldsymbol{\Delta'}(\Psi_1) \,\&\, \forall x(\boldsymbol{\Delta'}(\Phi_1)) \,\&\, \Psi_1$ as Ψ and where z is a variable symbol different from x and not occurring free in Φ_1 and Ψ:

$$
\begin{array}{rl}
\Psi \vdash_{\iota'} \exists!x(\Phi_1) & \text{given} \\
\Psi \vdash_{\iota'} \exists x(\Phi_1) & \&\text{-elim} \\
\Psi \vdash_{\iota'} \boldsymbol{\Delta}'(\forall x(\Phi_1)) & \text{defCons} \\
\boldsymbol{\Delta}'(\Psi_1)\,;\Psi_1 \vdash_{\iota'} \Psi_2 & \text{induction} \\
\boldsymbol{\Delta}'(\Psi_1)\,;\vdash_{\iota'} \Psi_1 \Rightarrow \Psi_2 & \text{DdRu2} \\
\boldsymbol{\Delta}'(\Psi_1)\,;\Psi_2 \vdash_{\iota'} \Psi_1 & \text{induction} \\
\boldsymbol{\Delta}'(\Psi_1)\,;\vdash_{\iota'} \Psi_2 \Rightarrow \Psi_1 & \text{DdRu2} \\
\boldsymbol{\Delta}'(\Psi_1)\,;\vdash_{\iota'} \Psi_1 \Leftrightarrow \Psi_2 & \&\text{-intro} \\
\boldsymbol{\Delta}'(\forall x(\Psi_1))\,;\vdash_{\iota'} \forall x(\Psi_1 \Leftrightarrow \Psi_2) & \text{SimGen} \\
\boldsymbol{\Delta}'(\forall x(\Psi_1)) \vdash_{\iota'} \forall x(\Psi_1 \Leftrightarrow \Psi_2) & \text{fromCtxt} \\
\Psi \vdash_{\iota'} \forall x(\Psi_1 \Leftrightarrow \Psi_2) & \text{Cut} \\
\boldsymbol{\Delta}'(\Psi_2) \,\&\, \forall x(\boldsymbol{\Delta}'(\Phi_2)) \,\&\, \Psi_2 \vdash_{\iota'} \exists!x(\Phi_2) & \text{given} \\
\Psi, \boldsymbol{\Delta}'(\Psi_2) \,\&\, \forall x(\boldsymbol{\Delta}'(\Phi_2)) \,\&\, \Psi_2 \vdash_{\iota'} \forall x(\Psi_1 \Leftrightarrow \Psi_2) \,\&\, \exists!x(\Phi_2) & \&\text{-intro} \\
\Psi, \boldsymbol{\Delta}'(\Psi_2) \,\&\, \forall x(\boldsymbol{\Delta}'(\Phi_2)) \,\&\, \Psi_2 \vdash_{\iota'} \forall x(\Psi_1 \Leftrightarrow \Psi_2) & \&\text{-elim} \\
\Psi, \boldsymbol{\Delta}'(\Psi_2) \,\&\, \forall x(\boldsymbol{\Delta}'(\Phi_2)) \,\&\, \Psi_2 \vdash_{\iota'} \iota x_{\Psi}(\Phi_1) = \iota x_{\boldsymbol{\Delta}'(\Psi_2)\&\forall x(\boldsymbol{\Delta}'(\Phi_2))}(\Phi_2) & \text{iota} \\
\vdash_{\iota'} \boldsymbol{\Delta}(\Psi) & \text{defAnt} \\
\Psi \vdash_{\iota'} \boldsymbol{\Delta}'(\Psi_1) \,\&\, \forall x(\boldsymbol{\Delta}'(\Phi_1)) \,\&\, \Psi_1 & \text{ass} \\
\Psi \vdash_{\iota'} \Psi_1 & \&\text{-elim} \\
\boldsymbol{\Delta}'(\Psi_1)\,;\Psi_1 \vdash_{\iota'} \Psi_2 & \text{induction} \\
\Psi \vdash_{\iota'} \Psi_2 & \text{Cut3} \\
\Psi \vdash_{\iota'} \boldsymbol{\Delta}'(\Psi_2) & \text{defCons} \\
\Psi \vdash_{\iota'} \forall x(\boldsymbol{\Delta}'(\Phi_1)) \,\&\, \Psi_1 & \&\text{-elim} \\
\Psi \vdash_{\iota'} \forall x(\boldsymbol{\Delta}'(\Phi_1)) & \&\text{-elim} \\
\boldsymbol{\Delta}'(\Phi_1)\,;\Phi_2 \vdash_{\iota'} \Phi_1 & \text{induction} \\
\boldsymbol{\Delta}'(\Phi_1)\,;\vdash_{\iota'} \boldsymbol{\Delta}(\Phi_2) & \text{defAnt} \\
\boldsymbol{\Delta}'(\Phi_1) \vdash_{\iota'} \boldsymbol{\Delta}(\Phi_2) & \text{fromCtxt} \\
\forall x(\boldsymbol{\Delta}'(\Phi_1)) \vdash_{\iota'} \forall x(\boldsymbol{\Delta}(\Phi_2)) & \text{SimGen} \\
\Psi \vdash_{\iota'} \forall x(\boldsymbol{\Delta}'(\Phi_2)) & \text{Cut} \\
\Psi \vdash_{\iota'} \boldsymbol{\Delta}'(\Psi_2) \,\&\, \forall x(\boldsymbol{\Delta}'(\Phi_2)) & \&\text{-intro} \\
\Psi \vdash_{\iota'} \boldsymbol{\Delta}'(\Psi_2) \,\&\, \forall x(\boldsymbol{\Delta}'(\Phi_2)) \,\&\, \Psi_2 & \&\text{-intro} \\
\Psi \vdash_{\iota'} \iota x_{\Psi}(\Phi_1) = \iota x_{\boldsymbol{\Delta}'(\Psi_2)\&\forall x(\boldsymbol{\Delta}'(\Phi_2))}(\Phi_2) & \text{Cut}
\end{array}
$$

In case the t_i are interchangeable with the corresponding s_i, we can apply the eq rule, yielding

$$\Psi \vdash_{\iota} \iota x_{\Psi}(\Phi_1) = \iota x_{\boldsymbol{\Delta}'(\Psi_1)\&\forall x(\boldsymbol{\Delta}'(\Phi_1))\&\Psi_1}(\Phi_1).$$

By induction, Ψ_1 and Ψ_2 are interchangeable; one easily derives that also $\boldsymbol{\Delta}(\Psi_1)$ and $\boldsymbol{\Delta}(\Psi_2)$ are interchangeable too; the analogous results for Φ_1 and Φ_2 also hold. By repeatedly applying the first part, one easily gets the required sequent

$$\Psi \vdash_{\iota} \iota x_{\Psi}(\Phi_1) = \iota x_{\boldsymbol{\Delta}'(\Psi_2)\&\forall x(\boldsymbol{\Delta}'(\Phi_2))\&\Psi_2}(\Phi_2).$$

This concludes the base case on the induction on the expansion ranks. Note that we were able to prove all parts of the theorem separately, which will not be possible any more in the induction step (for example, to prove the induction step for the first part, we need the fourth part).

For the induction step, we again perform induction on the nesting depths of the ι-terms, which we again prove by induction on $\operatorname{cpl} A(\alpha)$ and $\operatorname{cpl} t(\alpha)$ for the first part, $\operatorname{cpl} A(t_1)$ and $\operatorname{cpl} t(t_1)$ for the second part and $\operatorname{cpl}(\alpha)+1$ and $\operatorname{cpl}(t)+1$ for the other parts.

For the first part, the new cases are

- $t(\alpha) \equiv g(t_1(\alpha), t_2(\alpha), \ldots, t_n(\alpha))$ where g is a defined function symbol
  We can apply induction on $\widetilde{g^*}$ on the fourth part, since $\rho\big(\widetilde{g^*}\big) < \rho(t(\alpha))$.

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} g(t_1(\alpha), \ldots, t_n(\alpha)) = \left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots & t_n(\alpha) \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*} \quad \text{definition}$$

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\beta) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} g(t_1(\beta), \ldots, t_n(\beta)) = \left[\!\!\left[\begin{matrix} t_1(\beta) & \cdots & t_n(\beta) \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*} \quad \text{definition}$$

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} \left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots & t_n(\alpha) \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*} = \left[\!\!\left[\begin{matrix} t_1(\beta) & \cdots & t_n(\beta) \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*} \quad \text{induction}$$

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} g(t_1(\alpha), \ldots, t_n(\alpha)) = \left[\!\!\left[\begin{matrix} t_1(\beta) & \cdots & t_n(\beta) \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*} \quad \text{ET2}$$

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} \boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \,\&\, \boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\beta) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \quad \text{defCons}$$

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} \boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\beta) & \cdots & t_n(\beta) \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \quad \text{\&-elim}$$

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} g(t_1(\beta), \ldots, t_n(\beta)) = \left[\!\!\left[\begin{matrix} t_1(\beta) & \cdots & t_n(\beta) \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*} \quad \text{Cut}$$

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} \left[\!\!\left[\begin{matrix} t_1(\beta) & \cdots & t_n(\beta) \\ x_1 & & x_n \end{matrix}\right]\!\!\right] \widetilde{g^*} = g(t_1(\beta), \ldots, t_n(\beta)) \quad \text{ESy2}$$

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix} t_1(\alpha) & \cdots \\ x_1 & \end{matrix}\right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} g(t_1(\alpha), \ldots, t_n(\alpha)) = g(t_1(\beta), \ldots, t_n(\beta)) \quad \text{ET2}$$

Note that to apply the definition rule, we have to provide $UC(t_1(\alpha))$, $UC(t_2(\alpha))$, …, which are given, and $UC(t_1(\beta))$, $UC(t_2(\beta))$, which are easy to obtain using induction on the $t_i(\alpha)$ and the UC rule.

In the proof above, we supposed that in both applications of the definition rule, the result of renaming the bound variables in g^* was twice the same. Since the first application of the definition rule uses a renaming of bound variables of g^* in which no free variable of $t_i(\alpha)$ is bound and the second application uses a renaming of bound variables of g^* in which no free variable of $t_i(\beta)$ is bound, it might well be possible that

both formulae are different, so we will denote the latter as $\widetilde{\widetilde{g^*}}$. Finally, we consider an alphabetic variant of g^* in which no free variable of both $t_i(\alpha)$ and $t_i(\beta)$ is bound and denote it as $\widehat{g^*}$. It is easy to see that $\left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots & t_n(\alpha) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}$ and $\left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots & t_n(\alpha) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widehat{g^*}$ are alphabetic variants of each other, and so are $\left[\!\!\left[\begin{smallmatrix} t_1(\beta) & \dots & t_n(\beta) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{\widetilde{g^*}}$ and $\left[\!\!\left[\begin{smallmatrix} t_1(\beta) & \dots & t_n(\beta) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widehat{g^*}$. Hence we can start our proof as follows:

$$\mathbf{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots \\ x_1 & \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} g(t_1(\alpha), \dots, t_n(\alpha)) = \left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots & t_n(\alpha) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*} \quad \text{definition}$$

$$\mathbf{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1(\beta) & \dots \\ x_1 & \end{smallmatrix} \right]\!\!\right] \widetilde{\widetilde{g^*}}\right) \vdash_{\iota'} g(t_1(\beta), \dots, t_2(\beta)) = \left[\!\!\left[\begin{smallmatrix} t_1(\beta) & \dots & t_n(\beta) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{\widetilde{g^*}} \quad \text{definition}$$

$$\mathbf{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots \\ x_1 & \end{smallmatrix} \right]\!\!\right] \widehat{g^*}\right) \vdash_{\iota'} \left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots & t_n(\alpha) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widehat{g^*} = \left[\!\!\left[\begin{smallmatrix} t_1(\beta) & \dots & t_n(\beta) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widehat{g^*} \text{induction}$$

and then use induction on the third part to obtain

$$\mathbf{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots \\ x_1 & \end{smallmatrix} \right]\!\!\right] \widehat{g^*}\right) \vdash_{\iota'} g(t_1(\alpha), \dots, t_n(\alpha)) = \left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots & t_n(\alpha) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widehat{g^*}$$

and

$$\mathbf{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1(\beta) & \dots \\ x_1 & \end{smallmatrix} \right]\!\!\right] \widehat{g^*}\right) \vdash_{\iota'} g(t_1(\beta), \dots, t_n(\beta)) = \left[\!\!\left[\begin{smallmatrix} t_1(\beta) & \dots & t_n(\beta) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widehat{g^*}$$

We then continue the proof as before to end with

$$\mathbf{\Delta'}\left(\left[\!\!\left[\begin{smallmatrix} t_1(\alpha) & \dots & t_n(\alpha) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widehat{g^*}\right) \vdash_{\iota'} g(t_1(\alpha), \dots, t_n(\alpha)) = g(t_1(\beta), \dots, t_n(\beta))$$

which we can easily transform into the required sequent using induction on the third part.

- $A(\alpha) \equiv q(t_1(\alpha), t_2(\alpha), \dots, t_n(\alpha))$ where q is a defined predicate symbol Analogous.

The new cases for the second part are almost identical.

For the third part, the new cases involving defined symbols are easy:

- $t \equiv g(t_1, t_2, \dots, t_n)$ By structural induction, we have $\mathbf{\Delta'}(t_i) \vdash_{\iota'} t_i = \widetilde{t_i}$ for all $i = 1, \dots, n$. We can apply induction on the second part repeatedly (here again we need the induction on $\mathrm{cpl}(t) + 1$ and not $\mathrm{cpl}(t)$), yielding

$$
\begin{array}{rl}
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} g(t_1,t_2,t_3,\ldots,t_n) = g(\widetilde{t_1},t_2,t_3,\ldots,t_n) & \text{induction}\\
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} \boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots)) \mathbin{\&} \boldsymbol{\Delta}'\big(g(\widetilde{t_1},t_2,t_3,\ldots)\big) & \text{defCons}\\
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} \boldsymbol{\Delta}'\big(g(\widetilde{t_1},t_2,t_3,\ldots,t_n)\big) & \text{\&-elim}\\
\boldsymbol{\Delta}'\big(g(\widetilde{t_1},t_2,t_3,\ldots,t_n)\big) \vdash_{\iota'} g(\widetilde{t_1},t_2,t_3,\ldots,t_n) = g(\widetilde{t_1},\widetilde{t_2},t_3,\ldots,t_n) & \text{induction}\\
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} g(\widetilde{t_1},t_2,t_3,\ldots,t_n) = g(\widetilde{t_1},\widetilde{t_2},t_3,\ldots,t_n) & \text{Cut}\\
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} g(t_1,t_2,t_3,\ldots,t_n) = g(\widetilde{t_1},\widetilde{t_2},t_3,\ldots,t_n) & \text{ET2}\\
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} \boldsymbol{\Delta}'\big(g(\widetilde{t_1},t_2,t_3,\ldots)\big) \mathbin{\&} \boldsymbol{\Delta}'\big(g(\widetilde{t_1},\widetilde{t_2},t_3,\ldots)\big) & \text{defCons}\\
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} \boldsymbol{\Delta}'\big(g(\widetilde{t_1},\widetilde{t_2},t_3,\ldots)\big) & \text{\&-elim}\\
\boldsymbol{\Delta}'\big(g(\widetilde{t_1},\widetilde{t_2},t_3,\ldots,t_n)\big) \vdash_{\iota'} g(\widetilde{t_1},\widetilde{t_2},t_3,\ldots,t_n) = g(\widetilde{t_1},\widetilde{t_2},\widetilde{t_3},t_4,\ldots,t_n) & \text{induction}\\
\qquad\vdots & \\
\boldsymbol{\Delta}'\Big(g(\widetilde{t_1},\ldots,\widetilde{t_{n-1}},t_n)\Big) \vdash_{\iota'} g(\widetilde{t_1},\ldots,\widetilde{t_{n-1}},t_n) = g(\widetilde{t_1},\widetilde{t_2},\ldots,\widetilde{t_{n-1}},\widetilde{t_n}) & \text{induction}\\
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} g(\widetilde{t_1},\ldots,\widetilde{t_{n-1}},t_n) = g(\widetilde{t_1},\widetilde{t_2},\ldots,\widetilde{t_{n-1}},\widetilde{t_n}) & \text{Cut}\\
\boldsymbol{\Delta}'(g(t_1,t_2,t_3,\ldots,t_n)) \vdash_{\iota'} g(t_1,t_2,t_3,\ldots,t_n) = g(\widetilde{t_1},\widetilde{t_2},\ldots,\widetilde{t_{n-1}},\widetilde{t_n}) & \text{ET2}
\end{array}
$$

- $\alpha \equiv q(t_1,t_2,\ldots,t_n)$ Analogous.

For the fourth part, the new cases are

- $t \equiv g(\tau_1,\tau_2,\ldots,\tau_m)$ We have to derive

$$
\begin{aligned}
&\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix}\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right] & & \left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\\ y_1 & \cdots & y_m\end{matrix}\right]\!\!\right] \widetilde{g^*}\right)\\
&\qquad \vdash_{\iota'} g\left(\left[\!\!\left[\begin{matrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{matrix}\right]\!\!\right]\tau_1,\ldots,\left[\!\!\left[\begin{matrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{matrix}\right]\!\!\right]\tau_m\right)\\
&\qquad\qquad = g\left(\left[\!\!\left[\begin{matrix}s_1 & \cdots & s_n\\ x_1 & & x_n\end{matrix}\right]\!\!\right]\tau_1,\ldots,\left[\!\!\left[\begin{matrix}s_1 & \cdots & s_n\\ x_1 & & x_n\end{matrix}\right]\!\!\right]\tau_m\right)
\end{aligned}
$$

Induction on the fourth part on $\left[\!\!\left[\begin{smallmatrix}\tau_1 & \cdots & \tau_m\\ y_1 & & y_m\end{smallmatrix}\right]\!\!\right]\widetilde{g^*}$ yields (the required uniqueness conditions can be obtained from an application of the definition rule, as below)

$$
\begin{aligned}
&\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{matrix}\right]\!\!\right]\left[\!\!\left[\begin{matrix}\tau_1 & \cdots & \tau_m\\ y_1 & & y_m\end{matrix}\right]\!\!\right]\widetilde{g^*}\right) \vdash_{\iota'} \left[\!\!\left[\begin{matrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{matrix}\right]\!\!\right]\left[\!\!\left[\begin{matrix}\tau_1 & \cdots & \tau_m\\ y_1 & & y_m\end{matrix}\right]\!\!\right]\widetilde{g^*}\\
&\qquad\qquad = \left[\!\!\left[\begin{matrix}s_1 & \cdots & s_n\\ x_1 & & x_n\end{matrix}\right]\!\!\right]\left[\!\!\left[\begin{matrix}\tau_1 & \cdots & \tau_m\\ y_1 & & y_m\end{matrix}\right]\!\!\right]\widetilde{g^*}
\end{aligned}
$$

Applying induction on the fifth part, we see that this is interchangeable with

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \tau_1 \dots \right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} \left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] \tau_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix}\right]\!\!\right] \tau_m \right]\!\!\right] \widetilde{g^*} = \left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] \tau_1 \dots \right]\!\!\right] \widetilde{g^*}$$

and using induction on the first part, we actually can derive this sequent. The definition rule yields

$$\boldsymbol{\Delta}'\left(g\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \tau_1, \dots\right)\right) \vdash_{\iota'} g\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \tau_1, \dots\right) = \left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \tau_1 \dots \right]\!\!\right] \widetilde{g^*}$$

and an analogous sequent for the simultaneous substitution with the s_i. (Note that the antecedent is actually $\boldsymbol{\Delta}'\left(\left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] \tau_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix}\right]\!\!\right] \tau_m \right]\!\!\right] \widetilde{g^*}\right)$, given the definition of $\boldsymbol{\Delta}'$.) Using the ET2 and ERf2 rules, the required sequent is now easily obtained.

- $\alpha \equiv q(\tau_1, \tau_2, \dots, \tau_m)$ Analogous.

For the fifth part, the new cases are

- $t \equiv g(\tau_1, \tau_2, \dots, \tau_k)$ We have to show that $\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] g(\tau_1, \tau_2, \dots, \tau_k)$ and $\left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] s_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix}\right]\!\!\right] s_m \right]\!\!\right] g(\tau_1, \tau_2, \dots, \tau_k)$ are interchangeable.

  Induction on each of the τ_j yields the interchangeability of $\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \tau_j$ and $\left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] s_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix}\right]\!\!\right] s_m \right]\!\!\right] \tau_j$; hence we can apply induction on $\widetilde{g^*}$ on the fourth part to obtain the interchangeability of $\left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \tau_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \tau_k \right]\!\!\right] \widetilde{g^*}$ and $\left[\!\!\left[\left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] s_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix}\right]\!\!\right] s_m \right]\!\!\right] \tau_1 \dots \left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] s_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix}\right]\!\!\right] s_m \right]\!\!\right] \tau_k \right]\!\!\right] \widetilde{g^*}$.

  Using the definition axiom and the ET2 and ERf2 rules as above, it is easy to show that the former term is interchangeable with $g\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \tau_1, \dots, \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} s_1 & \dots & s_m \\ y_1 & & y_m \end{smallmatrix}\right]\!\!\right] \tau_k\right)$, i.e., the second term needed, and the latter term is interchangeable with $g\left(\left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] s_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix}\right]\!\!\right] s_m \right]\!\!\right] \tau_1, \dots, \left[\!\!\left[\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_1 & \end{smallmatrix}\right]\!\!\right] s_1 \dots \left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n \\ x_1 & & x_n \\ & y_m & \end{smallmatrix}\right]\!\!\right] s_m \right]\!\!\right] \tau_k\right)$, i.e., the first term needed.

- $\alpha \equiv q(\tau_1, \tau_2, \ldots, \tau_k)$ Analogous.

For the sixth part, the new cases are

- $t \equiv g(\tau_1, \tau_2, \ldots, \tau_k)$ Induction on $\left[\!\!\left[\begin{smallmatrix} \tau_1 & \cdots & \tau_k \\ y_1 & & y_k \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}$ yields

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \left[\!\!\left[\begin{smallmatrix} \tau_1 & \cdots & \tau_k \\ y_1 & & y_k \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix} \tau_1 & \cdots & \tau_k \\ y_1 & & y_k \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right).$$

Using the fifth part, we can rewrite the antecedent into the required $\boldsymbol{\Delta}'\Big(\Big[\!\!\Big[\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_1 \; \cdots \; \left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \tau_k \Big]\!\!\Big] \widetilde{g^*}\Big)$.

- $\alpha \equiv q(\tau_1, \tau_2, \ldots, \tau_k)$ Analogous.

For the seventh part, the new cases are

- $t \equiv g(\tau_1, \tau_2, \ldots, \tau_k)$ Apply induction on the seventh part on each of the τ_i, then apply the second part.

- $\alpha \equiv q(\tau_1, \tau_2, \ldots, \tau_k)$ Analogous.

□

Corollary 24, theorem 25 and corollary 26 can be transferred unmodified to the PITFOL' calculus.

Note that these theorems in combination with the definition rule yield that we may interchange $g(t_1, t_2, \ldots, t_n)$ and $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}$ where g is a defined function symbol, and likewise for defined predicate symbols.

Theorem 27 also holds in the PITFOL' calculus; we again have to prove somewhat more:

Theorem 71 *1. Given*

$$\begin{cases} \Sigma, \boldsymbol{\Delta}'(\alpha)\,;\alpha \vdash_{\iota'} \beta \\ \Sigma, \boldsymbol{\Delta}'(\alpha)\,;\beta \vdash_{\iota'} \alpha \end{cases}$$

then

$$\begin{cases} \Sigma, \boldsymbol{\Delta}'(A(\alpha))\,;A(\alpha) \vdash_{\iota'} A(\beta) \\ \Sigma, \boldsymbol{\Delta}'(A(\alpha))\,;A(\beta) \vdash_{\iota'} A(\alpha) \end{cases}$$

where $A(\alpha)$ is a formula possibly containing α and $A(\beta)$ is the same formula where a number of instances of α are replaced by β, and

$$\Sigma; \boldsymbol{\Delta}'(t(\alpha)) \vdash_{\iota'} t(\alpha) = t(\beta)$$

where $t(\alpha)$ is a term possibly containing α and $t(\beta)$ is the same term where a number of instances of α are replaced by β.

These results only hold if all of the following restrictions are met:

- *If α is replaced by β inside a quantifier $\forall x$, then x must not be a free variable of Σ.*
- *The uniqueness conditions for $A(\alpha)$, resp. $t(\alpha)$ must be derivable.*
- *When α is replaced by β inside a ι-term $\iota x_{\psi(\alpha)}(\varphi(\alpha))$, then the uniqueness conditions for both $\iota x_{\psi(\alpha)}(\varphi(\alpha))$ and $\iota x_{\psi(\beta)}(\varphi(\beta))$ must be derivable and x must not be a free variable of Σ.*

2. *Given*

$$\Sigma; \boldsymbol{\Delta}'(t_1) \vdash_{\iota'} t_1 = t_2$$

then analogously,

$$\begin{cases} \Sigma, \boldsymbol{\Delta}'(A(t_1)) \,; A(t_1) \vdash_{\iota'} A(t_2) \\ \Sigma, \boldsymbol{\Delta}'(A(t_1)) \,; A(t_2) \vdash_{\iota'} A(t_1) \\ \qquad\qquad \Sigma; \boldsymbol{\Delta}'(t(t_1)) \vdash_{\iota'} t(t_1) = t(t_2) \end{cases}$$

with analogous restrictions as in the first case.

3. *If*

$$\begin{cases} \Sigma; \boldsymbol{\Delta}'(t_1) \vdash_{\iota'} t_1 = s_1 \\ \Sigma; \boldsymbol{\Delta}'(t_2) \vdash_{\iota'} t_2 = s_2 \\ \qquad\qquad \dots \\ \Sigma; \boldsymbol{\Delta}'(t_2) \vdash_{\iota'} t_n = s_n \end{cases}$$

and the uniqueness conditions of $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \alpha$ and $\left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \alpha$, resp. $\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \tau$ and $\left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \tau$ are derivable, then for formulae α resp. terms τ, we have

$$\begin{cases} \Sigma, \boldsymbol{\Delta}'\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \alpha \right) ; \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \alpha \vdash_{\iota'} \left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \alpha \\ \Sigma, \boldsymbol{\Delta}'\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \alpha \right) ; \left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \alpha \vdash_{\iota'} \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \alpha \end{cases}$$

resp.

$$\Sigma; \boldsymbol{\Delta}'\left(\left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \tau \right) \vdash_{\iota'} \left[\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \tau = \left[\!\left[\begin{smallmatrix} s_1 & \cdots & s_n \\ x_1 & \cdots & x_n \end{smallmatrix} \right]\!\right] \tau$$

These results only hold if all of the following restrictions are met:

- *For each subformula of α of the form $\forall x(\beta)$, x must not be a free variable of Σ.*
- *For each subterm of α of the form $\iota x_{\psi}(\varphi)$, x must not be a free variable of Σ.*

and similar restrictions for terms τ.

Proof.

Analogous to theorem 70, by adding Σ in front of the context of all sequents involved. □

Corollary 28 transfers unmodified to the PITFOL' calculus, just as all following theorems and corollaries up to theorem 32.

In general, theorem 33 does not hold in the PITFOL' calculus. As a counterexample, define $g(x,y)$ as $\iota x_{y=\iota x_{\forall x(x=x)\vee x=x}(x=y)}(x=y)$ and take a term t for which x is not a free variable of t. Then we have

$$\begin{aligned}
\boldsymbol{\Delta}'([t/x]\,g(x,y)) &\equiv \boldsymbol{\Delta}'(g(t,y)) \\
&\equiv \boldsymbol{\Delta}'\Big(\left[\!\!\left[\begin{smallmatrix} t & y \\ x & y \end{smallmatrix}\right]\!\!\right] \iota x_{y=\iota x_{\forall x(x=x)\vee x=x}(x=y)}(x=y)\Big) \\
&\equiv \boldsymbol{\Delta}'\left(\begin{array}{c} \iota x_{\boldsymbol{\Delta}'(\left[\!\left[\begin{smallmatrix} t & y \\ x & y \end{smallmatrix}\right]\!\right](y=\iota x_{\forall x(x=x)\vee x=x}(x=y)))}(x=y) \\ \&\left[\!\!\left[\begin{smallmatrix} t & y \\ x & y \end{smallmatrix}\right]\!\!\right](y=\iota x_{\forall x(x=x)\vee x=x}(x=y)) \end{array}\right) \\
&\equiv (\neg(\forall x(x=x)) \Rightarrow (\boldsymbol{\Delta}'(t)\ \&\ \boldsymbol{\Delta}'(t)))\ \&\ \boldsymbol{\Delta}'(t)\ \&\ (\forall x(x=x)\vee t=t) \\
&\qquad \&\ y=\iota x_{(\neg(\forall x(x=x))\Rightarrow(\boldsymbol{\Delta}'(t)\&\boldsymbol{\Delta}'(t)))\&\boldsymbol{\Delta}'(t)\&(\forall x(x=x)\vee t=t)}(x=y) \\
&\equiv ((\neg\forall x(x=x)) \Rightarrow (\boldsymbol{\Delta}'(t)\ \&\ \boldsymbol{\Delta}'(t)))\ \&\ \boldsymbol{\Delta}'(t)\ \&\ (\forall x(x=x)\vee t=t)
\end{aligned}$$

and

$$\begin{aligned}
[t/x]\,\boldsymbol{\Delta}'(g(x,y)) &\equiv [t/x]\,\boldsymbol{\Delta}'\Big(\left[\!\!\left[\begin{smallmatrix} x & y \\ x & y \end{smallmatrix}\right]\!\!\right] \iota x_{y=\iota x_{\forall x(x=x)\vee x=x}(x=y)}(x=y)\Big) \\
&\equiv [t/x]\,\boldsymbol{\Delta}'\left(\begin{array}{c} \iota x_{\boldsymbol{\Delta}'(\left[\!\left[\begin{smallmatrix} x & y \\ x & y \end{smallmatrix}\right]\!\right](y=\iota x_{\forall x(x=x)\vee x=x}(x=y)))}(x=y) \\ \&\left[\!\!\left[\begin{smallmatrix} x & y \\ x & y \end{smallmatrix}\right]\!\!\right](y=\iota x_{\forall x(x=x)\vee x=x}(x=y)) \end{array}\right) \\
&\equiv [t/x]\,\boldsymbol{\Delta}'\Big(\iota x_{(\forall x(x=x)\vee x=x)\&y=\iota x_{\forall x(x=x)\vee x=x}(x=y)}(x=y)\Big) \\
&\equiv [t/x]\,\big((\forall x(x=x)\vee x=x)\ \&\ y=\iota x_{\forall x(x=x)\vee x=x}(x=y)\big) \\
&\equiv (\forall x(x=x)\vee t=t)\ \&\ y=\iota x_{\boldsymbol{\Delta}'(t)\&(\forall x(x=x)\vee t=t)}(x=y)
\end{aligned}$$

It is not difficult to see that $\boldsymbol{\Delta}'([t/x]g(x,y))$ is a validity, so if theorem 33 would hold, we could derive

$$\begin{array}{ll}
\vdash_{\iota'} (\forall x(x=x) \vee t=t) \,\&\, y = \iota x_{\boldsymbol{\Delta}'(t)\&(\forall x(x=x)\vee t=t)}(x=y) & \\
\vdash_{\iota'} y = \iota x_{\boldsymbol{\Delta}'(t)\&(\forall x(x=x)\vee t=t)}(x=y) & \&\text{-elim} \\
\vdash_{\iota'} \boldsymbol{\Delta}'(t) \,\&\, (\forall x(x=x) \vee t=t) & \text{defCons} \\
\vdash_{\iota'} \boldsymbol{\Delta}'(t) & \&\text{-elim}
\end{array}$$

for any term t. Taking $t \equiv \iota x_{\neg\forall x(x=x)}(x=y)$, this would yield a derivation of $\vdash_{\iota'} \neg\forall x(x=x)$, which is impossible by the equiconsistency result.

However, in the sequel we will prove a variant of theorem 33 (theorem 73). Since we used theorem 33 in the derivation of the PartCons3 rule, we cannot use this rule in the PITFOL' calculus; in the sequel, we will prove the slightly modified PartCons4 rule.

For property 34 until corollary 43, the proofs are unmodified in the PITFOL' calculus.

For theorem 44, extra cases are

- $\tau \equiv g(\tau_1, \tau_2, \ldots, \tau_n)$ We have to derive

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix} [t_1/x]\tau_1 & \ldots & [t_1/x]\tau_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} g([t_1/x]\tau_1, \ldots, [t_1/x]\tau_n) = g([t_2/x]\tau_1, \ldots, [t_2/x]\tau_n)$$

and an analogous sequent where t_1 and t_2 are interchanged. We will only derive the first sequent explicitly. Using the interchange theorem, it suffices to derive

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix} [t_1/x]\tau_1 & \ldots & [t_1/x]\tau_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} \left[\!\!\left[\begin{smallmatrix} [t_1/x]\tau_1 & \ldots & [t_1/x]\tau_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*} = \left[\!\!\left[\begin{smallmatrix} [t_2/x]\tau_1 & \ldots & [t_2/x]\tau_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}$$

which is readily obtained by theorem 70.4.

- $\tau \equiv q(\tau_1, \tau_2, \ldots, \tau_n)$ Analogous.

Lemma 45 and property 46 can also be transferred unmodified.

Since $\mathcal{R}$ and $\mathcal{D}$ are not defined for formulae containing defined symbols, we will skip the next few theorems.

Lemma 50 transfers unmodified to the PITFOL' calculus.

Next we consider theorem 51. The new cases are

- $\tau \equiv g(t_1, t_2, \ldots, t_n)$ We will only derive one of the two required sequents; the other is analogous. Induction yields $\boldsymbol{\Delta}'([t/x]t_i) \vdash_{\iota'} [t'/x]t_i = [t/x]t_i$ for $i = 1, 2, \ldots, n$. We can again use theorem 70.4 which yields

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix} [t/x]t_1 & \ldots & [t/x]t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}\right) \vdash_{\iota'} \left[\!\!\left[\begin{smallmatrix} [t/x]t_1 & \ldots & [t/x]t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*} = \left[\!\!\left[\begin{smallmatrix} [t'/x]t_1 & \ldots & [t'/x]t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{g^*}$$

and again invoke the interchange theorem to obtain the required sequent.

- $\tau \equiv q(t_1, t_2, \ldots, t_n)$ Analogous.

Theorem 52 also holds in the PITFOL' calculus:

Proof.

The proof of the first part also holds here.

For the second part, new cases arise when α or τ is a defined symbol. We will consider the case $\tau \equiv g(t_1, \ldots, t_n)$; the case $\alpha \equiv q(t_1, \ldots, t_n)$ is analogous. Induction and the ESy2 rule yield $\boldsymbol{\Delta}([t/x]t_i) \vdash_\iota [t/x]t_i = [\![t/x]\!]t_i$ for all $i = 1, \ldots, n$; hence we can apply theorem 70.4 to obtain

$$\boldsymbol{\Delta}'\Big(\Big[\!\Big[\begin{smallmatrix} [t/x]t_1 & \cdots & [t/x]t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big]\, \widehat{g^*}\Big) \vdash_{\iota'} \Big[\!\Big[\begin{smallmatrix} [t/x]t_1 & \cdots & [t/x]t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big]\, \widehat{g^*} = \Big[\!\Big[\begin{smallmatrix} [\![t/x]\!]t_1 & \cdots & [\![t/x]\!]t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big]\, \widehat{g^*}$$

where $\widehat{g^*}$ is an alphabetic variant of g^* in which the bound variables differ from the free variables of t and the t_i. Using the definition rule and theorem 70.3, we can easily show to be interchangeable with

$$\boldsymbol{\Delta}'(g([t/x]t_1, \ldots, [t/x]t_n)) \vdash_{\iota'} g([t/x]t_1, \ldots, [t/x]t_n) = g([\![t/x]\!]t_1, \ldots, [\![t/x]\!]t_n)$$

Applying ESy2 then finally yields the third required sequent, from which the first one readily follows.

To derive the second sequent, note that by induction, we have that $[t/x]t_i$ and $[\![t/x]\!]t_i$ are interchangeable under the condition $\boldsymbol{\Delta}(t)$ for all $i = 1, \ldots, n$; hence we can apply theorem 71.3 to get the sequent

$$\boldsymbol{\Delta}'(t)\,;\boldsymbol{\Delta}'\Big(\Big[\!\Big[\begin{smallmatrix} [\![t/x]\!]t_1 & \cdots & [\![t/x]\!]t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big]\, \widehat{g^*}\Big) \vdash_{\iota'} \Big[\!\Big[\begin{smallmatrix} [t/x]t_1 & \cdots \\ x_1 & \end{smallmatrix} \Big]\!\Big]\, \widehat{g^*} = \Big[\!\Big[\begin{smallmatrix} [\![t/x]\!]t_1 & \cdots \\ x_1 & \end{smallmatrix} \Big]\!\Big]\, \widehat{g^*},$$

from which we easily obtain the required sequent.

For the third part, again new cases arise when α or τ is a defined symbol; these are handled easily using theorem 70.4. □

Property 55 also holds in the PITFOL' calculus; this follows easily from theorem 70.4.

Lemma 56 also holds in the PITFOL' calculus.

Proof.

We again only explicitly mention the case $\tau \equiv g(\tau_1, \tau_2, \ldots, \tau_m)$. Induction yields for all i

$$\boldsymbol{\Delta}\Big(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big]\, \tau_i\Big) \vdash_\iota \begin{bmatrix} t'_1 & \cdots & t'_n \\ x_1 & & x_n \end{bmatrix} \tau_i = \Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big]\, \tau_i$$

hence we can apply theorem 70.4 to get

$$\boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{matrix}\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_1 & \cdots\\ y_1 & \end{matrix}\right]\!\!\right]\widehat{g^*}\right)\vdash_{\iota'}\left[\!\!\left[\begin{matrix}\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_1 & \cdots & \left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_m\\ y_1 & & y_m\end{matrix}\right]\!\!\right]\widehat{g^*}$$
$$=\left[\!\!\left[\begin{matrix}\left[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\right]\tau_1 & \left[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\right]\tau_m & \cdots\\ y_1 & y_m & \end{matrix}\right]\!\!\right]\widehat{g^*}$$

where $\widehat{g^*}$ is an alphabetic variant of g^* whose bound variables do not occur free in all $\left[\!\!\left[\begin{smallmatrix}t_1 & \cdots & t_n\\ x_1 & & x_n\end{smallmatrix}\right]\!\!\right]\tau_i$ and $\left[\begin{smallmatrix}t'_1 & \cdots & t'_n\\ x_1 & & x_n\end{smallmatrix}\right]\tau_i$.

Using the machinery developed above, this sequent is easily transformed into the required sequent. □

Corollary 57 also holds in the PITFOL' calculus and we already established the same about lemmas 58 up to 60 in the proof of theorem 70.

As already announced, we will prove a variant of theorem 33; first we prove the following lemma.

Lemma 72 *For each formula α and term t of the* PITFOL' *calculus of which the uniqueness conditions are derivable, $[\![t/x]\!]\alpha$ and $\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\alpha$ are interchangeable if at least one of both substitutions is defined (and then, so is the other one).*

Analogously for terms τ.

Proof.

We prove this by structural induction. The only case where the definition of both substitutions differs is the case $\tau \equiv \iota y_\psi(\varphi)$; the other cases are easy.

Using theorem 52.1 and the definition of simultaneous substitution, we have the following possibilities:

- $x \equiv y$ We have to show
$$\iota y_{\boldsymbol{\Delta}'([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) \rightleftharpoons \iota y_{\boldsymbol{\Delta}'(\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\psi)\&\forall y(\boldsymbol{\Delta}'(\varphi))\&\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\psi}(\varphi).$$
- $x \not\equiv y$ and x is not free in φ We have to show
$$\iota y_{\boldsymbol{\Delta}'([\![t/x]\!]\psi)\&[\![t/x]\!]\psi}(\varphi) \rightleftharpoons \iota y_{\boldsymbol{\Delta}'(\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\psi)\&\forall y(\boldsymbol{\Delta}'(\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\varphi))\&\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\psi}\left(\left[\!\!\left[\begin{matrix}t\\ x\end{matrix}\right]\!\!\right]\varphi\right).$$
Since x is not free in φ, the latter term is $\iota x_{\boldsymbol{\Delta}'(\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\psi)\&\forall y(\boldsymbol{\Delta}'(\varphi))\&\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\psi}(\varphi)$.
- $x \not\equiv y$ and x is free in φ We have to show
$$\iota y_{\boldsymbol{\Delta}'([\![t/x]\!]\psi)\&\forall y(\boldsymbol{\Delta}'([\![t/x]\!]\varphi))\&[\![t/x]\!]\psi}([\![t/x]\!]\varphi) \rightleftharpoons \iota y_{\boldsymbol{\Delta}'(\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\psi)\&\forall y(\boldsymbol{\Delta}'(\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\varphi))\&\left[\!\!\left[\begin{smallmatrix}t\\ x\end{smallmatrix}\right]\!\!\right]\psi}\left(\left[\!\!\left[\begin{matrix}t\\ x\end{matrix}\right]\!\!\right]\varphi\right).$$

Applying structural induction immediately yields the last case.
For the first two cases, one easily derives $\boldsymbol{\Delta}'\big(\llbracket {}^{t}_{x} \rrbracket\, \psi\big) \,\&\, \forall y(\boldsymbol{\Delta}'(\varphi)) \,\&\, \llbracket {}^{t}_{x} \rrbracket\, \psi \rightharpoonup \boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi) \,\&\, \llbracket t/x \rrbracket \psi$ so we only have left to derive $\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi) \,\&\, \llbracket t/x \rrbracket \psi \rightharpoonup \boldsymbol{\Delta}'\big(\llbracket {}^{t}_{x} \rrbracket\, \psi\big) \,\&\, \forall y(\boldsymbol{\Delta}'(\varphi)) \,\&\, \llbracket {}^{t}_{x} \rrbracket\, \psi$ which is easy except for $\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi) \,\&\, \llbracket t/x \rrbracket \psi \rightharpoonup \forall y(\boldsymbol{\Delta}'(\varphi))$:

$$\begin{array}{rl}
 & \psi \vdash_{\iota'} \exists! y(\varphi) \qquad \text{uniqueness condition} \\
 & \psi \vdash_{\iota'} \exists y(\varphi) \qquad \&\text{-elim} \\
 & \psi \vdash_{\iota'} \forall y(\boldsymbol{\Delta}'(\varphi)) \qquad \text{defCons} \\
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi)\,, \underbrace{\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \forall y(\boldsymbol{\Delta}'(\varphi)))}_{\equiv \boldsymbol{\Delta}'(\forall y(\boldsymbol{\Delta}'(\varphi)))}; \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \underbrace{\llbracket t/x \rrbracket \forall y(\boldsymbol{\Delta}'(\varphi))}_{\equiv \forall y(\boldsymbol{\Delta}'(\varphi))} \qquad \text{RefSubst} \\
 & \vdash_{\iota'} \boldsymbol{\Delta}'(\boldsymbol{\Delta}'(\forall y(\varphi))) \qquad \text{Ddef} \\
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi)\,; \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \forall y(\boldsymbol{\Delta}'(\varphi)) \qquad \text{CutCtxt} \\
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi) \,\&\, \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \forall y(\boldsymbol{\Delta}'(\varphi)) \qquad \text{FromCtxt}
\end{array}$$

□

Theorem 73 *For each formula α and term t of the* PITFOL' *calculus of which the uniqueness conditions are derivable, if the substitution $\llbracket t/x \rrbracket \alpha$ is defined, then also the substitution $\llbracket t/x \rrbracket \boldsymbol{\Delta}'(\alpha)$ is defined.*

If the uniqueness conditions for α and t are derivable, then $\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \alpha) \vdash_{\iota'} \llbracket t/x \rrbracket \boldsymbol{\Delta}'(\alpha)$.

The analogous theorem for terms τ of the PITFOL' *calculus also holds.*

Proof.

By induction on $\rho(\tau)$ and $\rho(\alpha)$, which we prove in turn by induction on the complexity of τ and α. Most cases are analogous to theorem 33.

- $\tau \equiv \iota y_{\psi}(\varphi)$
  - $x \equiv y$ or x is not a free variable of φ Using theorem 52.1, we have to prove $\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi) \,\&\, \llbracket t/x \rrbracket \psi \vdash_{\iota'} \llbracket t/x \rrbracket \psi$, which is easy:

$$\begin{array}{rl}
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi)\,; \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \llbracket t/x \rrbracket \psi \qquad \text{AssCtxt} \\
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi) \,\&\, \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \llbracket t/x \rrbracket \psi \qquad \text{fromCtxt}
\end{array}$$

  - $x \not\equiv y$ and x is a free variable of φ Again using theorem 52.1, we have to prove $\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi) \,\&\, \forall y(\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \varphi)) \,\&\, \llbracket t/x \rrbracket \psi \vdash_{\iota'} \llbracket t/x \rrbracket \psi$:

$$\begin{array}{rl}
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi)\,; \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \llbracket t/x \rrbracket \psi \qquad \text{AssCtxt} \\
 & \vdash_{\iota'} \boldsymbol{\Delta}'(\boldsymbol{\Delta}'(\forall y(\llbracket t/x \rrbracket \varphi))) \qquad \text{Ddef} \\
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi)\,; \forall y(\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \varphi)), \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \llbracket t/x \rrbracket \psi \qquad \text{Weak} \\
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi)\,; \forall y(\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \varphi)) \,\&\, \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \llbracket t/x \rrbracket \psi \qquad \text{AnU} \\
\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \psi) \,\&\, \forall y(\boldsymbol{\Delta}'(\llbracket t/x \rrbracket \varphi)) \,\&\, \llbracket t/x \rrbracket \psi & \vdash_{\iota'} \llbracket t/x \rrbracket \psi \qquad \text{fromCtxt}
\end{array}$$

- $\tau \equiv g(t_1, \ldots, t_n)$ where g is a defined function symbol We have to derive

$$\boldsymbol{\Delta}'\Big(\Big[\!\!\Big[\begin{matrix} [\![t/x]\!]t_1 & \ldots & [\![t/x]\!]t_n \\ x_1 & \ldots & x_n \end{matrix} \Big]\!\!\Big]\, \widetilde{g^*}\Big) \vdash_{\iota'} [\![t/x]\!]\, \boldsymbol{\Delta}'\Big(\Big[\!\!\Big[\begin{matrix} t_1 & \ldots & t_n \\ x_1 & \ldots & x_n \end{matrix} \Big]\!\!\Big]\, \widetilde{g^*}\Big)$$

Using theorem 55 and lemma 72, we have

$$\boldsymbol{\Delta}'\Big(\Big[\!\!\Big[\begin{matrix} [\![t/x]\!]t_1 & \ldots & [\![t/x]\!]t_n \\ x_1 & \ldots & x_n \end{matrix} \Big]\!\!\Big]\, \widetilde{g^*}\Big) \vdash_{\iota'} \boldsymbol{\Delta}'\Big(\Big[\!\!\Big[\begin{matrix} [\![{}^{t}_{x}]\!]\, t_1 & \ldots & [\![{}^{t}_{x}]\!]\, t_n \\ x_1 & \ldots & x_n \end{matrix} \Big]\!\!\Big]\, \widetilde{g^*}\Big)$$

Using theorem 70.5, we get

$$\boldsymbol{\Delta}'\Big(\Big[\!\!\Big[\begin{matrix} [\![t/x]\!]t_1 & \ldots & [\![t/x]\!]t_n \\ x_1 & \ldots & x_n \end{matrix} \Big]\!\!\Big]\, \widetilde{g^*}\Big) \vdash_{\iota'} \boldsymbol{\Delta}'\Big(\Big[\!\!\Big[\begin{matrix} t \\ x \end{matrix} \Big]\!\!\Big]\, \Big[\!\!\Big[\begin{matrix} t_1 & \ldots & t_n \\ x_1 & \ldots & x_n \end{matrix} \Big]\!\!\Big]\, \widetilde{g^*}\Big)$$

Again using lemma 72, we transform this into

$$\boldsymbol{\Delta}'\Big(\Big[\!\!\Big[\begin{matrix} [\![t/x]\!]t_1 & \ldots & [\![t/x]\!]t_n \\ x_1 & \ldots & x_n \end{matrix} \Big]\!\!\Big]\, \widetilde{g^*}\Big) \vdash_{\iota'} \boldsymbol{\Delta}'\Big([\![t/x]\!]\, \Big[\!\!\Big[\begin{matrix} t_1 & \ldots & t_n \\ x_1 & \ldots & x_n \end{matrix} \Big]\!\!\Big]\, \widetilde{g^*}\Big)$$

Using induction on the $\rho(\tau)$, we obtain the required sequent.

□

Since we used theorem 33 in the derivation of the PartCons3 rule, we cannot use this rule in the PITFOL' calculus. However, we are now able to derive

$$\begin{array}{rl}
 & \boxed{\text{PartCons4}} \\
\Sigma; \Gamma \vdash_{\iota'} [\![t/x]\!]\alpha & \text{prem} \\
\Sigma; \Gamma \vdash_{\iota'} \boldsymbol{\Delta}'([\![t/x]\!]\alpha) & \text{defCons} \\
\boldsymbol{\Delta}'([\![t/x]\!]\alpha) \vdash_{\iota'} [\![t/x]\!]\boldsymbol{\Delta}'(\alpha) & \text{Theorem 73} \\
\Sigma; \Gamma \vdash_{\iota'} [\![t/x]\!]\boldsymbol{\Delta}'(\alpha) & \text{Cut} \\
\Sigma; \Gamma \vdash_{\iota'} [\![t/x]\!](\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha) & \&\text{-intro} \\
\vdash_{\iota'} \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha)) & \text{Ddef} \\
\boldsymbol{\Delta}(\alpha) \vdash_{\iota} \boldsymbol{\Delta}(\alpha) & \text{ass} \\
\vdash_{\iota'} \boldsymbol{\Delta}(\alpha) \Rightarrow \boldsymbol{\Delta}(\alpha) & \text{DdRu2} \\
\vdash_{\iota'} \boldsymbol{\Delta}(\boldsymbol{\Delta}(\alpha) \,\&\, \alpha) & \&\text{-intro} \\
\vdash_{\iota'} \forall x(\boldsymbol{\Delta}'(\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha)) & \forall\text{-intro} \\
\forall x(\neg(\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha)) \vdash_{\iota'} \forall x(\neg(\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha)) & \text{ass} \\
\forall x(\neg(\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha)) \vdash_{\iota'} \neg(\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha) & \forall\text{-elim} \\
\boldsymbol{\Delta}'(\forall x(\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha))\,, & \\
\boldsymbol{\Delta}'([\![t/x]\!](\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha))\,; & \\
\forall x(\neg(\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha)) \vdash_{\iota'} [\![t/x]\!]\neg(\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha) & \text{RefSubst}
\end{array}$$

$$\begin{array}{rl}
\boldsymbol{\Delta}'([\![t/x]\!](\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha))\,;\forall x(\neg(\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha)) \vdash_{\iota'} [\![t/x]\!]\neg(\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha) & \text{CutCtxt}\\
\Sigma, \boldsymbol{\Delta}'([\![t/x]\!](\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha))\,; & \\
\quad \Gamma, \forall x(\neg(\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha)) \vdash_{\iota'} \neg\forall x(\neg(\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha)) & \text{contra}\\
\Sigma, \boldsymbol{\Delta}'([\![t/x]\!](\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha))\,;\Gamma \vdash_{\iota'} \neg\forall x(\neg(\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha)) & \text{SeDe}\\
\Sigma; \boldsymbol{\Delta}'([\![t/x]\!](\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha))\,,\Gamma \vdash_{\iota'} \neg\forall x(\neg(\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha)) & \text{FromCtxt2}\\
\Sigma;\Gamma \vdash_{\iota'} \boldsymbol{\Delta}'([\![t/x]\!](\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha)) & \text{defCons}\\
\Sigma;\Gamma \vdash_{\iota'} \neg\forall x(\neg(\boldsymbol{\Delta}'(\alpha)\ \&\ \alpha)) & \text{Cut}
\end{array}$$

5.6 Semantics

We extend the notion of **interpretation** from the PITFOL calculus to the PITFOL' calculus with

- If g is a defined function symbol, then $\mathcal{I}(g(t_1, t_2, \ldots, t_n)) := \mathcal{I}^{\mathcal{I}t_1\mathcal{I}t_2\ldots\mathcal{I}t_n}_{x_1\ x_2\ \ldots x_n}(g^*)$.
- If q is a defined predicate symbol, then $\mathcal{I}(q(t_1, t_2, \ldots, t_n)) := \mathcal{I}^{\mathcal{I}t_1\mathcal{I}t_2\ldots\mathcal{I}t_n}_{x_1\ x_2\ \ldots x_n}(q^*)$.

Note that it is possible that one or more of the t_i are undefined in $\mathcal{I}$, whereas in the definition of interpretation we required that a variable symbol always be interpreted as an element of the domain ω. Hence we define an **extended interpretation** which is identical to an "ordinary" interpretation, with the exception that a variable symbol may be given the status 'undefined' in an extended interpretation (in other words, $\mathcal{I}(x) = \perp$). We will call an interpretation in which all variables are interpreted as elements of the domain ω an **ordinary interpretation** to highlight the difference with extended interpretations. Given an (extended or ordinary) interpretation $\mathcal{I}$ and a variable symbol x, the notation $\mathcal{I}^{\perp}_{x}$ then indicates the extended interpretation which is identical to $\mathcal{I}$ except that $\mathcal{I}^{\perp}_{x}(x) = \perp$.

Remark that even in an extended interpretation $\mathcal{I}$, the formula $\forall x(\alpha)$ is valid if for each $a \in \omega$, α is valid in $\mathcal{I}^{a}_{x}$, so even in extended interpretations, we still do not consider the cases where $\mathcal{I}(x)$ is undefined to determine the interpretation of $\forall x(\alpha)$. The same holds for the interpretation of ι-terms. Also remark that the definitions of **consequence** and **sound sequent** are unchanged and only consider ordinary interpretations.

Theorem 74 *1. Given a formula α and terms t_1, t_2, ..., t_n of the* PITFOL' *calculus. If the uniqueness conditions of α, t_1, ... and t_n hold and the uniqueness conditions of all defining terms and formulae hold, then for any ordinary interpretation $\mathcal{I}$, $\mathcal{I}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix}\right]\!\!\right]\alpha\right) = \mathcal{I}^{\mathcal{I}t_1\ldots\mathcal{I}t_n}_{x_1\ \ldots x_n}(\alpha)$ if the simultaneous substitution is defined, and likewise for terms τ.*

2. *Given a formula β of the* PITFOL' *calculus for which the uniqueness conditions hold. If the uniqueness conditions of all defining terms and formulae hold, then for any ordinary interpretation $\mathcal{I}$, β is defined in $\mathcal{I}$ if and only if either $\boldsymbol{\Delta}'(\beta)$ is $\top$ or valid in $\mathcal{I}$ and β is undefined in $\mathcal{I}$ if and only if $\boldsymbol{\Delta}'(\beta) \not\equiv \top$ and $\boldsymbol{\Delta}'(\beta)$ is invalid in $\mathcal{I}$.*

   The analogous theorem for terms σ also holds.

Note that the second part implies that $\boldsymbol{\Delta}'(\beta)$ is either $\top$ or defined in $\mathcal{I}$.

Proof.

We prove this by simultaneous induction on $\rho\Big(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \alpha\Big)$ and $\rho(\beta)$, which we prove in turn by structural induction. The interesting cases are

- If $\tau \equiv x_i$ then we for the first part, have to show that $\mathcal{I}(t_i) = \mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}(x_i)$, which is trivial.
- If $\sigma \equiv x$ then we note that x is defined in any interpretation, and $\boldsymbol{\Delta}(x_i) \equiv \top$.
- If $\tau \equiv g(\tau_1, \tau_2, \ldots, \tau_m)$ where g is a defined function symbol, we have to show that

  $$\mathcal{I}\left(g\left(\Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \tau_1, \ldots, \Big[\!\Big[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \Big]\!\Big] \tau_m\right)\right) = \mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}(g(\tau_1, \ldots, \tau_m))$$

  i.e.,

  $$\mathcal{I}^{\mathcal{I}\left[\!\left[\begin{smallmatrix} t_1 \ldots t_n \\ x_1 \ldots x_n \end{smallmatrix} \right]\!\right]\tau_1 \ldots \mathcal{I}\left[\!\left[\begin{smallmatrix} t_1 \ldots t_n \\ x_1 \ldots x_n \end{smallmatrix} \right]\!\right]\tau_m}_{y_1 \qquad\quad y_m}(g^*) = \mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n \mathcal{I}\tau_1 \ldots \mathcal{I}\tau_m}_{x_1 \ \ldots x_n \ y_1 \ \ldots y_m}(g^*)$$

  Induction on each of the τ_i yields that the left hand side equals

  $$\mathcal{I}^{\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}(\tau_1) \ldots \mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}(\tau_m)}_{y_1 \qquad\quad y_m}(g^*)$$

  Since the only free variables of g^* are (a subset of) $\{y_1, y_2, \ldots, y_m\}$, it is now easy to see that this is equal to $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n \mathcal{I}\tau_1 \ldots \mathcal{I}\tau_m}_{x_1 \ \ldots x_n \ y_1 \ \ldots y_m}(g^*)$.
- If $\sigma \equiv g(\sigma_1, \sigma_2, \ldots, \sigma_m)$ where g is a defined function symbol, we first have to show that if σ is defined in $\mathcal{I}$, then $\boldsymbol{\Delta}'\Big(\Big[\!\Big[\begin{smallmatrix} \sigma_1 & \cdots & \sigma_m \\ y_1 & & y_m \end{smallmatrix} \Big]\!\Big] \widetilde{g^*}\Big)$ is either $\top$ or is valid in $\mathcal{I}$ and vice versa.

  By definition, σ is defined in $\mathcal{I}$ if and only if g^* is defined in $\mathcal{I}^{\mathcal{I}\sigma_1 \ldots \mathcal{I}\sigma_m}_{y_1 \ \ldots y_m}$; one sees easily that this in turn is equivalent with $\widetilde{g^*}$ being defined in $\mathcal{I}^{\mathcal{I}\sigma_1 \ldots \mathcal{I}\sigma_m}_{y_1 \ \ldots y_m}$.

By corollary 64, $\rho\Big(\Big[\!\Big[{\sigma_1 \atop y_1} \cdots {\sigma_m \atop y_m} \Big]\!\Big] \widetilde{g^*}\Big) < \rho(g(\sigma_1, \ldots, \sigma_m))$, so we can apply induction on the first part and get that $\widetilde{g^*}$ is defined in $\mathcal{I}^{\mathcal{I}\sigma_1 \ldots \mathcal{I}\sigma_m}_{y_1 \ \ldots y_m}$ if and only if $\Big[\!\Big[{\sigma_1 \atop y_1} \cdots {\sigma_m \atop y_m} \Big]\!\Big] \widetilde{g^*}$ is defined in $\mathcal{I}$. Applying induction on the second part, this finally is equivalent with $\mathbf{\Delta}\Big(\Big[\!\Big[{\sigma_1 \atop y_1} \cdots {\sigma_m \atop y_m} \Big]\!\Big] \widetilde{g^*}\Big)$ being either $\top$ or valid in $\mathcal{I}$.

Next, we must prove σ is undefined in $\mathcal{I}$, whenever $\mathbf{\Delta'}\Big(\Big[\!\Big[{\sigma_1 \atop y_1} \cdots {\sigma_m \atop y_m} \Big]\!\Big] \widetilde{g^*}\Big)$ is not $\top$ and is invalid in $\mathcal{I}$. This is done analogously to the first equivalence.

- If $\tau \equiv \iota x_\psi(\varphi)$ then we must show that

$$\mathcal{I}(\iota x_\Psi(\Phi)) = \mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}(\iota x_\psi(\varphi))$$

when $x \notin \{x_1, \ldots, x_n\}$ (the other case is analogous) where

$$\Psi := \mathbf{\Delta'}\Big(\Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \psi\Big) \,\&\, \forall x \left(\mathbf{\Delta'}\Big(\Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \varphi\Big)\right) \,\&\, \Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \psi$$

and $\Phi := \Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \varphi$.

First, we show that when $\iota x_\Psi(\Phi)$ is undefined in $\mathcal{I}$, so is $\iota x_\psi(\varphi)$ in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$. The possible cases in which $\iota x_\Psi(\Phi)$ is undefined in $\mathcal{I}$ are:

  - Ψ is undefined in $\mathcal{I}$. Since by induction on the second part of the theorem, $\mathbf{\Delta'}\Big(\Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \psi\Big)$ and $\forall x \left(\mathbf{\Delta'}\Big(\Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \varphi\Big)\right)$ are always defined in $\mathcal{I}$, we conclude that $\Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \psi$ is undefined in $\mathcal{I}$. Hence, by structural induction, ψ is undefined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$, so $\iota x_\psi(\varphi)$ is undefined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$.
  - Ψ is invalid in $\mathcal{I}$. Then at least one of $\mathbf{\Delta'}\Big(\Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \psi\Big)$, $\forall x \left(\mathbf{\Delta'}\Big(\Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \varphi\Big)\right)$ and $\Big[\!\Big[{t_1 \atop x_1} \cdots {t_n \atop x_n} \Big]\!\Big] \psi$ is invalid in $\mathcal{I}$. Using structural induction on the first part of the theorem, we easily conclude that ψ is undefined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$, $\forall x \varphi$ is undefined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$, or ψ is invalid in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$. In the first and the last case, we see immediately that $\iota x_\psi(\varphi)$ is undefined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$. In the second case, there exists an $a \in \omega$ such that φ is undefined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n a}_{x_1 \ \ldots x_n \ x}$. But since the uniqueness condition of $\iota x_\psi(\varphi)$ holds, ψ cannot be defined or valid in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$.

– Ψ is valid in $\mathcal{I}$ and there is no a such that Φ is valid in $\mathcal{I}^a_x$ or there are multiple such a. Using structural induction, we get that the same holds about φ in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$ and hence again $\iota x_\psi(\varphi)$ is undefined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$.

Next, we show that when $\iota x_\Psi(\Phi)$ is defined in $\mathcal{I}$, so is $\iota x_\psi(\varphi)$ in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$ and their interpretations are the same.

When $\iota x_\Psi(\Phi)$ is defined in $\mathcal{I}$, Ψ is valid in $\mathcal{I}$ and there exists an unique a such that Φ is valid in $\mathcal{I}$. Hence $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \psi$ is valid in $\mathcal{I}$ and by induction, ψ is valid in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$. Also by induction, the unique a for which Φ is valid in $\mathcal{I}$ is the same a for which φ is valid in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$.

We now only have to show the other direction, i.e., when $\iota x_\psi(\varphi)$ is undefined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$, so is $\iota x_\Psi(\Phi)$ in $\mathcal{I}$, and when $\iota x_\psi(\varphi)$ is defined in $\mathcal{I}^{\mathcal{I}t_1 \ldots \mathcal{I}t_n}_{x_1 \ \ldots x_n}$, so is $\iota x_\Psi(\Phi)$ in $\mathcal{I}$ and their interpretations are the same. This is easy using contraposition.

- If $\sigma \equiv \iota x_\psi(\varphi)$ then first we must show that $\iota x_\psi(\varphi)$ is defined in $\mathcal{I}$ whenever ψ is valid in $\mathcal{I}$.
  By definition, $\iota x_\psi(\varphi)$ is defined in $\mathcal{I}$ whenever ψ is valid in $\mathcal{I}$ and there exists a unique $a \in \omega$ such that φ is valid in $\mathcal{I}^a_x$. Conversely, if ψ is valid in $\mathcal{I}$, the unique existence of such an a follows from the uniqueness condition of σ.

  Next, we have to prove that $\iota x_\psi(\varphi)$ is undefined in $\mathcal{I}$ whenever ψ is invalid in $\mathcal{I}$.
  By definition, $\iota x_\psi(\varphi)$ is undefined in $\mathcal{I}$ whenever ψ is invalid or undefined in $\mathcal{I}$ or there does not exist a unique $a \in \omega$ such that φ is valid in $\mathcal{I}^a_x$. Note that because the uniqueness condition of σ holds, if there does not exist a unique $a \in \omega$ such that φ is valid in $\mathcal{I}^a_x$, ψ cannot be valid in $\mathcal{I}$. Moreover, because of the uniqueness condition, ψ cannot be undefined in $\mathcal{I}$ either.

□

Lemma 75 *Given a formula α of the* PITFOL' *calculus and an ordinary interpretation $\mathcal{I}$. Suppose that the uniqueness conditions of α are valid in $\mathcal{I}$. Then*

$$\begin{aligned} \alpha \text{ is valid in } \mathcal{I} &\Leftrightarrow \mathcal{E}(\alpha) \text{ is valid in } \mathcal{I} \\ \alpha \text{ is invalid in } \mathcal{I} &\Leftrightarrow \mathcal{E}(\alpha) \text{ is invalid in } \mathcal{I} \\ \alpha \text{ is undefined in } \mathcal{I} &\Leftrightarrow \mathcal{E}(\alpha) \text{ is undefined in } \mathcal{I} \end{aligned}$$

Given a term t of the PITFOL' *calculus and an interpretation $\mathcal{I}$ for which all uniqueness conditions are valid in $\mathcal{I}$. Then $\mathcal{I}(t) = \mathcal{I}(\mathcal{E}(t))$.*

Note that $\mathcal{E}(\alpha)$ and $\mathcal{E}(t)$ are formulae without defined symbols, and the definition of interpretation in the PITFOL' calculus coincides with that of the PITFOL calculus. Hence, we have a connection between the interpretation of a formula or term in the PITFOL' calculus and the interpretation of its expansion in the PITFOL calculus.

Proof.

We prove this by induction on the complexity of α and t. The interesting cases are

- $\alpha = p(t_1, \ldots, t_n)$ If α is valid in $\mathcal{I}$, then all $\mathcal{I}(t_i)$ are defined and $\mathcal{I}(p)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))$ holds. By induction, $\mathcal{I}(p)(\mathcal{I}(\mathcal{E}(t_1)), \ldots, \mathcal{I}(\mathcal{E}(t_n)))$ also holds, i.e., $p(\mathcal{E}(t_1), \ldots, \mathcal{E}(t_n))$ is valid in $\mathcal{I}$.

  The other two equivalences are proved analogously.

- $\alpha = q(t_1, \ldots, t_n)$ where q is a defined predicate symbol We have to show that the interpretations of $q(t_1, \ldots, t_n)$ and $\left[\!\!\left[\begin{smallmatrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{q^*}$ are identical, which is easy using theorem 74.

□

Theorem 76 (Soundness of the pitfol' calculus) *If* $\Gamma \vdash'_\iota \alpha$ *then* $\Gamma \models'_\iota \alpha$.

Proof.

We know that we can translate a PITFOL' proof of $\Gamma \vdash'_\iota \alpha$ to a PITFOL proof of $\mathcal{E}(\Gamma) \vdash_\iota \mathcal{E}(\alpha)$. Because we showed the PITFOL calculus to be complete, we can conclude that $\mathcal{E}(\Gamma) \models_\iota \mathcal{E}(\alpha)$. Since expansions do not contain defined symbols, we also have $\mathcal{E}(\Gamma) \models_{\iota'} \mathcal{E}(\alpha)$. Applying lemma 75 completes the proof. □

Theorem 77 *Each* PITFOL' *formula* α *is interchangeable with its expansion* $\mathcal{E}(\alpha)$ *and likewise for terms* t.

Proof.

We prove this by induction on the expansion rank which we then prove by structural induction. Interesting cases are

- $\alpha \equiv q(t_1, \ldots, t_n)$ We have to show that α is interchangeable with $\left[\!\!\left[\begin{smallmatrix} \mathcal{E}(t_1) & \ldots & \mathcal{E}(t_n) \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \mathcal{E}\left(\widetilde{q^*}\right)$. Using theorem 67, the latter term is interchangeable with $\mathcal{E}\left(\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{q^*}\right)$. Using induction on the expansion rank, this is in turn interchangeable with $\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n \\ x_1 & & x_n \end{smallmatrix} \right]\!\!\right] \widetilde{q^*}$. Using the definition rule, we immediately get the required sequents.

- $t \equiv g(t_1, \ldots, t_n)$ Analogous.

□

Theorem 78 *If the* PITFOL *calculus is complete, then so is the* PITFOL' *calculus.*

Proof.

Suppose $\Gamma \models_{\iota'} \alpha$. Using lemma 75, we obtain that $\mathcal{E}(\Gamma) \models_{\iota'} \mathcal{E}(\alpha)$. Because this sequent does not contain defined symbols, this is equivalent with $\mathcal{E}(\Gamma) \models_{\iota} \mathcal{E}(\alpha)$. Applying the completeness of the PITFOL calculus yields that $\mathcal{E}(\Gamma) \vdash_{\iota} \mathcal{E}(\alpha)$. Because each PITFOL proof is also a PITFOL' proof, we have $\mathcal{E}(\Gamma) \vdash_{\iota'} \mathcal{E}(\alpha)$. Finally, using the previous theorem we get $\Gamma \vdash_{\iota'} \alpha$. □

5.7 Defined symbols with predicate arguments

We still are not able to define symbols such as $\Rightarrow$ or $\Leftrightarrow$ because they don't have terms but formulae as arguments. For this purpose, we enhance the definition of simultaneous substitution: as before, we allow a variable symbol to be substituted by a term, but now we also allow a predicate constant to be substituted by a formula. For example,

$$\left[\!\!\left[\begin{smallmatrix} x = y\, y = x \\ p_1 \qquad p_2 \end{smallmatrix} \right]\!\!\right] \neg(p_1 \,\&\, \neg p_2) \equiv \neg(x = y \,\&\, \neg y = x).$$

In general, we allow an arbitrary number of both kinds of substitutions to happen simultaneously, i.e., the general form of a simultaneous substitution is now $\left[\!\!\left[\begin{smallmatrix} t_1 & \ldots & t_n & \alpha_1 & \ldots & \alpha_m \\ x_1 & & x_n & p_1 & & p_m \end{smallmatrix} \right]\!\!\right] \alpha$.

Formally, we extend the definition of simultaneous substitution with the cases

- $\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n & \alpha_1 & \dots & \alpha_m \\ x_1 & & x_n & p_1 & & p_m \end{smallmatrix} \right]\!\!\right] p_i \equiv \alpha_i$
- $\left[\!\!\left[\begin{smallmatrix} t_1 & \dots & t_n & \alpha_1 & \dots & \alpha_m \\ x_1 & & x_n & p_1 & & p_m \end{smallmatrix} \right]\!\!\right] q \equiv q$ when q is a predicate symbol different from $p_1, \dots, p_m$.

We can use this extended notion of simultaneous substitution to introduce defined symbols in the same way as before. For instance, by defining $\text{implies}(\alpha_1, \alpha_2) = \neg(\alpha_1 \,\&\, \neg\alpha_2)$, the definition rule yields, given $UC(\alpha)$ and $UC(\beta)$, the sequent

$$\boldsymbol{\Delta}'(\alpha) \,\&\, \alpha \Rightarrow \boldsymbol{\Delta}(\beta) \vdash_{\iota'} \text{implies}(\alpha, \beta) = \neg(\alpha \,\&\, \neg\beta)$$

for any two formulae α and β.

Note that as before, when renaming the bound variable of q^* we need to avoid the free variables of $t_1, \dots, t_n$ and $\alpha_1, \dots, \alpha_m$. Also note that a defined symbol is allowed to have both terms and formulae as arguments. We illustrate both observations with the example definition

$$\text{ifthen}(f, a, b) = \iota x((f \Rightarrow x = a) \,\&\, ((\neg f) \Rightarrow x = b))$$

where f is a function symbol and a and b are variable symbols. Semantically, $\text{ifthen}(f, a, b)$ equals a when f is true and b otherwise. The definition rule then yields for example

$$\vdash_{\iota'} \text{ifthen}(x = y, 1, 0) = \iota z((x = y \Rightarrow z = 1) \,\&\, ((\neg(x = y)) \Rightarrow z = 0))$$

If we did not rename x to z, we would have obtained the term

$$\iota x((x = y \Rightarrow x = 1) \,\&\, ((\neg(x = y)) \Rightarrow x = 0))$$

of which we cannot prove the uniqueness condition (to see this, note that when $y = 0$, there is no x satisfying the definiens).

5.8 Defined quantifier symbols

The only abbreviations left in our system are the quantifiers $\exists$ and $\exists!$, so we will extend the definition mechanism to defined quantifier symbols.

5.8.1 Examples

Some quantifiers we would like to be able to define are

$$\exists x(p(x)) = \neg\forall x(\neg p(x))$$
$$\exists! x(p(x)) = \exists x(p(x)) \,\&\, \forall x\forall y((p(x) \,\&\, p(y)) \Rightarrow x = y)$$
$$\mu x(f(x)) = \iota y_{\exists y(\forall x(f(x)\geq y)\&\forall z(\forall x(f(x)>z)\Rightarrow y\geq z))} \begin{pmatrix} \forall x(f(x) \geq y) \\ \,\&\, \forall z(\forall x(f(x) \geq z) \Rightarrow y \geq z) \end{pmatrix}$$
$$\nu x(f(x), a, b) = \iota y_{\exists y(\ldots)} \begin{pmatrix} \forall x((a \leq x \,\&\, x \leq b) \Rightarrow f(x) \geq y) \\ \,\&\, \forall z(\forall x((a \leq x \,\&\, x \leq b) \Rightarrow f(x) \geq z) \Rightarrow y \geq z) \end{pmatrix}$$
$$\{x : p(x)\} = \iota y_{\exists y\forall x(x\in y\Leftrightarrow p(x))}(\forall x(x \in y \Leftrightarrow p(x)))$$
$$\{(x, y) : p(x, y)\} = \{z : \exists y\exists z(z = (x, y) \,\&\, p(x, y))\}$$

The third example, $\mu x(f(x))$, presumes a theory of real numbers and denotes the minimum value of its argument f; it is undefined if f does not have a minimum.

Likewise, $\nu x(f(x), a, b)$ denotes the minimum value of the function on the closed interval $[a, b]$ if f has a minimum value on that interval. Here we see that the first argument $f(x)$ is bound by the ν quantifier but the other two arguments are not bound.

The last examples illustrate set builder notation in a theory of sets. The last one illustrates a quantifier binding two variables x and y.

5.8.2 Extending simultaneous substitution

To be able to define quantifier symbols, we again extend the definition of simultaneous substitution: we allow a function or predicate symbols applied to variable symbols to be substituted by a term resp. a formula. So we will have

$$\left[\!\!\left[\begin{matrix} x = z \\ p(x) \end{matrix} \right]\!\!\right] \exists x(p(x)) \equiv \exists x(x = z).$$

Formally, we define

$$\left[\!\!\left[\begin{matrix} \tau_1 & & \tau_n & & \alpha_1 & & \alpha_m \\ f_1(y_1^1,\ldots,y_{N_1}^1) & \cdots & f_n(y_1^n,\ldots,y_{N_n}^n) & p_1(x_1^1,x_2^1,\ldots,x_{M_1}^1) & \cdots & p_m(x_1^m,\ldots,x_{M_m}^m) \end{matrix} \right]\!\!\right] p_i(y_1, y_2, \ldots, y_{M_i})$$
$$\equiv \begin{bmatrix} y_1 & \cdots & y_{M_i} \\ x_1^i & & x_{M_i}^i \end{bmatrix} \alpha_i$$

Note that to uniformise our treatment, we don't substitute variable symbols anymore; otherwise, the notation would have become

$$\left[\!\!\left[\begin{matrix} T_1 & & T_k & \tau_1 & & \tau_n & \alpha_1 & & \alpha_m \\ z_1 & \cdots & z_k & f_1(y_1^1,\ldots,y_{N_1}^1) & \cdots & f_n(y_1^n,\ldots,y_{N_n}^n) & p_1(x_1^1,x_2^1,\ldots,x_{M_1}^1) & \cdots & p_m(x_1^m,\ldots,x_{M_m}^m) \end{matrix} \right]\!\!\right] \ldots$$

We can use function constant symbols to fulfill the function variable symbols held before.

Note that the simultaneous substitution

$$\left[\!\!\left[\begin{smallmatrix} \tau_1 \\ f_1(y_1^1,\ldots,y_{N_1}^1) \end{smallmatrix} \cdots \begin{smallmatrix} \tau_n \\ f_n(y_1^n,\ldots,y_{N_n}^n)\, p_1(x_1^1,x_2^1,\ldots,x_{M_1}^1) \end{smallmatrix} \cdots \begin{smallmatrix} \alpha_m \\ p_m(x_1^m,\ldots,x_{M_m}^m) \end{smallmatrix} \right]\!\!\right] p_i(t_1, t_2, \ldots, t_{n_i})$$

is only defined when all t_i are variable symbols.

5.8.3 Extending the definition rule

We are now in a position to develop a variant of the definition rule for defined quantifiers. A definition of a defined quantifier is of the form $Qz_1 \ldots z_k(f_1(y_1^1, \ldots, y_{N_1}^1), \ldots, f_n(y_1^n, \ldots, y_{N_n}^n), p_1(x_1^1, x_2^1, \ldots, x_{M_1}^1), \ldots, p_m(x_1^m, \ldots, x_{M_m}^m)) = Q^*$. We call Q^* the **definition** of Q and $f_1(y_1^1, \ldots, y_{N_1}^1)$, $\ldots$, $f_n(y_1^n, \ldots, y_{N_n}^n)$, $p_1(x_1^1, x_2^1, \ldots, x_{M_1}^1)$, $\ldots$, $p_m(x_1^m, \ldots, x_{M_m}^m)$ the **argument symbols** of Q^*. We call the variable symbols $z_1, \ldots, z_k$ the **binding variable**s of the definition.

Note that to uniformise the treatment of argument symbols, the argument symbols cannot be free variables anymore. Instead, we can use a constant symbols, that is, function symbols without arguments (so $N_i = 0$). This is illustrated in the definition of $\nu x(f(x), a, b)$ where a and b are indeed constant symbols.

The definition rule for defined quantifiers is

$$\frac{\boxed{\text{definition}} \quad UC(t_1), UC(t_2), \ldots, UC(t_n), UC(\alpha_1), UC(\alpha_2), \ldots, UC(\alpha_m)}{\boldsymbol{\Delta}'(Qu_1 \ldots u_k(\ldots)) \vdash_{\iota'} Qu_1 \ldots u_k(t_1, \ldots, t_n, \alpha_1, \ldots, \alpha_m)} = \left[\!\!\left[\begin{smallmatrix} t_1 \\ \left[\begin{smallmatrix} u_1 \ldots u_k \\ z_1 \ldots z_k \end{smallmatrix}\right] f_1(y_1^1,\ldots,y_{N_1}^1) \end{smallmatrix} \cdots \begin{smallmatrix} \alpha_m \\ \left[\begin{smallmatrix} u_1 \ldots u_k \\ z_1 \ldots z_k \end{smallmatrix}\right] p_m(x_1^m,\ldots,x_{M_m}^m) \end{smallmatrix} \right]\!\!\right] \widetilde{Q^*}$$

where Q^* is a term containing only symbols defined before Q where we define

$$\boldsymbol{\Delta}'(Qu_1 \ldots u_k(t_1, \ldots, t_n, \alpha_1, \ldots, \alpha_m)) :\equiv \boldsymbol{\Delta}'\left(\left[\!\!\left[\begin{smallmatrix} t_1 \\ \left[\begin{smallmatrix} u_1 \ldots u_k \\ z_1 \ldots z_k \end{smallmatrix}\right] f_1(y_1^1,\ldots,y_{N_1}^1) \end{smallmatrix} \cdots \begin{smallmatrix} \alpha_m \\ \left[\begin{smallmatrix} u_1 \ldots u_k \\ z_1 \ldots z_k \end{smallmatrix}\right] p_m(x_1^m,\ldots,x_{M_m}^m) \end{smallmatrix} \right]\!\!\right] \widetilde{Q^*} \right)$$

or

$$\frac{\boxed{\text{definition}} \quad UC(t_1), UC(t_2), \ldots, UC(t_n), UC(\alpha_1), UC(\alpha_2), \ldots, UC(\alpha_m)}{\boldsymbol{\Delta}'(Qu_1 \ldots u_k(\ldots)) \vdash_{\iota'} Qu_1 \ldots u_k(t_1, \ldots, t_n, \alpha_1, \ldots, \alpha_m)} \Leftrightarrow \left[\!\!\left[\begin{smallmatrix} t_1 \\ \left[\begin{smallmatrix} u_1 \ldots u_k \\ z_1 \ldots z_k \end{smallmatrix}\right] f_1(y_1^1,\ldots,y_{N_1}^1) \end{smallmatrix} \cdots \begin{smallmatrix} \alpha_m \\ \left[\begin{smallmatrix} u_1 \ldots u_k \\ z_1 \ldots z_k \end{smallmatrix}\right] p_m(x_1^m,\ldots,x_{M_m}^m) \end{smallmatrix} \right]\!\!\right] \widetilde{Q^*}$$

where Q^* is a formula containing only symbols defined before Q and $\mathbf{\Delta}'(Qy(\tau_1, \ldots, \tau_n, \alpha_1, \ldots, \alpha_m))$ is defined as above.

We require that Q^* does not contain any free variables. We obtain $\widetilde{Q^*}$ by renaming the bound variables of Q^* as follows:

- A binding variable z_i of the definition must be renamed to u_i
- The other bound variables have to be renamed such that they are different from $u_1, \ldots, u_k$ and from $FV(t_1) \setminus \{y_1^1, \ldots, y_{N_1}^1\}, \ldots, FV(t_n) \setminus \{y_1^n, \ldots, y_{N_n}^n\}, FV(\alpha_1) \setminus \{x_1^1, \ldots, x_{M_1}^1\}, \ldots$ and $FV(\alpha_n) \setminus \{x_1^n, \ldots, x_{M_m}^m\}$.

The effect is that the variables $\left[\begin{smallmatrix} u_1 & \cdots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] y_1^i, \ldots, \left[\begin{smallmatrix} u_1 & \cdots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] y_{N_i}^i$ of τ_i are bound by the defined quantifier Q and likewise the variables $\left[\begin{smallmatrix} u_1 & \cdots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] x_1^i, \ldots, \left[\begin{smallmatrix} u_1 & \cdots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] x_{N_i}^i$ of α_i.
For example, in $\nu x(\alpha_1, \tau_1, \tau_2)$, the quantifier ν binds x in α_1 but binds no variables in τ_1 and τ_2.

Accordingly, we define substitution of defined binders as

$$[t/x]Qz_1 \ldots z_k(\tau_1, \ldots, \tau_n, \alpha_1, \ldots, \alpha_m) :\equiv Qz_1 \ldots z_k(\tau_1', \ldots, \tau_n', \alpha_1', \ldots, \alpha_m')$$

where $\tau_i' :\equiv [t/x]\tau_i$ when x is not free in $\left[\begin{smallmatrix} u_1 & \cdots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] f_i(y_1^i, \ldots, y_{N_i}^i)$ and $\tau_i' :\equiv \tau_i$ otherwise; likewise, $\alpha_i' :\equiv [t/x]\alpha_i$ when x is not free in $\left[\begin{smallmatrix} u_1 & \cdots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] p_i(x_1^i, \ldots, x_{M_i}^i)$ and $\alpha_i' :\equiv \alpha_i$otherwise.
If one of the necessary substitutions $[t/x]\tau_i$ or $[t/x]\alpha_i$ is not defined, then the substitution $[t/x]Qz_1 \ldots z_k(\tau_1, \ldots, \tau_n, \alpha_1, \ldots, \alpha_m)$ is also not defined.

5.8.4 Example

We illustrate the working of the definition rule with an example introduced earlier. We have

$$\vdash_{\iota'} \exists!y(x = y) = \left[\!\!\left[\begin{smallmatrix} x = y \\ p(y) \end{smallmatrix}\right]\!\!\right] \exists y(p(y)) \,\&\, \forall y \forall z((p(y) \,\&\, p(z)) \Rightarrow y = z)$$

i.e.,

$$\vdash_{\iota'} \exists!y(x = y) = \exists y(x = y) \,\&\, \forall y \forall z((x = y \,\&\, x = z) \Rightarrow y = z)$$

Chapter 6

Conclusions

6.1 Summary

The PITFOL calculus encompasses the following 17 deduction rules:

$$\boxed{\text{ass}}\quad \frac{UC(\alpha) \qquad \Sigma;\vdash_\iota \boldsymbol{\Delta}(\alpha)}{\Sigma;\alpha\vdash_\iota \alpha}$$

When $\boldsymbol{\Delta}(\alpha)$ is $\top$, the context Σ must be empty.

$$\boxed{\text{\&-intro}}\quad \frac{\Sigma_1;\Gamma\vdash_\iota \alpha \qquad \Sigma_2;\Delta\vdash_\iota \beta}{\Sigma_1,\Sigma_2;\Gamma,\Delta\vdash_\iota \alpha\,\&\,\beta}$$

$$\boxed{\text{\&-elim}}\quad \frac{\Sigma;\Gamma\vdash_\iota \alpha\,\&\,\beta}{\Sigma;\Gamma\vdash_\iota \alpha}$$

$$\boxed{\text{\&-elim}}\quad \frac{\Sigma;\Gamma\vdash_\iota \alpha\,\&\,\beta}{\Sigma;\Gamma\vdash_\iota \beta}$$

$$\boxed{\text{rem}}\quad \frac{\Sigma_1;\Gamma,\alpha\vdash_\iota \beta \qquad \Sigma_2;\Delta,\neg\alpha\vdash_\iota \beta}{\Sigma_1,\Sigma_2;\Gamma,\Delta\vdash_\iota \beta}$$

$$\boxed{\text{contra}}\quad \frac{UC(\beta) \qquad \Sigma_1;\Gamma\vdash_\iota \alpha \qquad \Sigma_2;\Delta\vdash_\iota \neg\alpha}{\Sigma_1,\Sigma_2;\Gamma,\Delta\vdash_\iota \beta}$$

$$\boxed{\forall\text{-intro}}\quad \frac{\Sigma;\Gamma\vdash_\iota \alpha}{\Sigma;\Gamma\vdash_\iota \forall x(\alpha)}$$ provided x is not free in Γ and Σ

$$\boxed{\forall\text{-elim}}\quad \frac{\Sigma;\Gamma\vdash_\iota \forall x(\alpha)}{\Sigma;\Gamma\vdash_\iota \alpha}$$

$$\boxed{\text{subst}}\quad \frac{UC(t) \qquad \Sigma;\Gamma\vdash_\iota \alpha}{\boldsymbol{\Delta}(t)\,,[t/x]\Sigma;[t/x]\Gamma\vdash_\iota [t/x]\alpha}$$

$$\boxed{\text{eq}}\quad \frac{UC(t)}{\boldsymbol{\Delta}(t)\vdash_\iota t=t}$$

$$\boxed{\text{eqSubst}}\ \frac{UC(t) \quad \Sigma;\Gamma \vdash_\iota \alpha}{\Sigma, \boldsymbol{\Delta}(t)\,;\Gamma, x = t \vdash_\iota [t/x]\alpha} \qquad \boxed{\text{iota}}\ \frac{\psi \vdash_\iota \exists!x(\varphi)}{\psi \vdash_\iota [\iota x_\psi(\varphi)/x]\widetilde{\varphi}}$$

$$\boxed{\text{UC}}\ \frac{\Sigma;\Gamma \vdash_\iota \alpha}{\psi \vdash_\iota \exists!x(\varphi)}\ \text{where } \iota x_\psi(\varphi) \text{ is a } \iota\text{-term occurring in } \Sigma,\ \Gamma \text{ or } \alpha.$$

$$\boxed{\text{defAnt}}\ \frac{\Sigma;\Gamma,\alpha \vdash_\iota \beta}{\Sigma;\vdash_\iota \boldsymbol{\Delta}(\alpha)} \qquad \boxed{\text{defCons}}\ \frac{\Sigma;\Gamma \vdash_\iota \alpha}{\Sigma;\Gamma \vdash_\iota \boldsymbol{\Delta}(\alpha)} \qquad \boxed{\text{toCtxt}}\ \frac{\Sigma;\sigma\ \&\ \Gamma,\Delta \vdash_\iota \alpha}{\Sigma,\sigma;\Gamma,\Delta \vdash_\iota \alpha} \qquad \boxed{\text{fromCtxt}}\ \frac{\Sigma,\sigma;\Gamma \vdash_\iota \alpha}{\Sigma;\sigma\ \&\ \Gamma \vdash_\iota \alpha}$$

The PITFOL' calculus adds rules introducing defined symbols:

$$\boxed{\text{definition}}\ \frac{UC(t_1), UC(t_2), \ldots, UC(t_n), UC(\alpha_1), UC(\alpha_2), \ldots, UC(\alpha_m)}{\boldsymbol{\Delta}'(g(t_1,\ldots,\alpha_m)) \vdash_{\iota'} g(t_1,\ldots,t_n,\alpha_1,\ldots,\alpha_m) = \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n & \alpha_1 & \cdots & \alpha_m \\ x_1 & & x_n & p_1 & & p_m \end{matrix} \right]\!\!\right] \widetilde{g^*}}$$

$$\boxed{\text{definition}}\ \frac{UC(t_1), UC(t_2), \ldots, UC(t_n), UC(\alpha_1), UC(\alpha_2), \ldots, UC(\alpha_m)}{\boldsymbol{\Delta}'(q(t_1,\ldots,\alpha_m)) \vdash_{\iota'} q(t_1,\ldots,t_n,\alpha_1,\ldots,\alpha_m) \Leftrightarrow \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n & \alpha_1 & \cdots & \alpha_m \\ x_1 & & x_n & p_1 & & p_m \end{matrix} \right]\!\!\right] \widetilde{q^*}}$$

$$\boxed{\text{definition}}\ \frac{UC(t_1), UC(t_2), \ldots, UC(t_n), UC(\alpha_1), UC(\alpha_2), \ldots, UC(\alpha_m)}{\boldsymbol{\Delta}'(Qu_1\ldots u_k(\ldots)) \vdash_{\iota'} Qu_1\ldots u_k(t_1,\ldots,t_n,\alpha_1,\ldots,\alpha_m) = \left[\!\!\left[\begin{matrix} t_1 & & \alpha_m \\ \left[\begin{smallmatrix} u_1 & \ldots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] f_1(y_1^1,\ldots,y_{N_1}^1) & \cdots & \left[\begin{smallmatrix} u_1 & \ldots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] p_m(x_1^m,\ldots,x_{M_m}^m) \end{matrix} \right]\!\!\right] \widetilde{Q^*}}$$

$$\boxed{\text{definition}}\ \frac{UC(t_1), UC(t_2), \ldots, UC(t_n), UC(\alpha_1), UC(\alpha_2), \ldots, UC(\alpha_m)}{\boldsymbol{\Delta}'(Qu_1\ldots u_k(\ldots)) \vdash_{\iota'} Qu_1\ldots u_k(t_1,\ldots,t_n,\alpha_1,\ldots,\alpha_m) \Leftrightarrow \left[\!\!\left[\begin{matrix} t_1 & & \alpha_m \\ \left[\begin{smallmatrix} u_1 & \ldots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] f_1(y_1^1,\ldots,y_{N_1}^1) & \cdots & \left[\begin{smallmatrix} u_1 & \ldots & u_k \\ z_1 & & z_k \end{smallmatrix}\right] p_m(x_1^m,\ldots,x_{M_m}^m) \end{matrix} \right]\!\!\right] \widetilde{Q^*}}$$

6.2 Treatment of partially defined terms

In this section, we will summarise how the calculus handles partially defined terms. The main idea is that we can only talk about a potentially undefined term under conditions where one is sure that the undefinedness will not occur. For example, we can only mention $\frac{y}{x}$ under the condition $x \neq 0$. This already implies a semantics where the evaluation order of different parts of a formula is fixed, since we will want to ascertain that the condition $x \neq 0$ is evaluated before $\frac{y}{x}$. We will require that when we would encounter an undefined term when evaluating a formula following this fixed evaluation order, the whole formula becomes undefined. A consequence of this requirement is that the semantics is **monotone** with respect to undefinedness: when we replace an undefined subterm or subformula by a defined one, the resulting term or formula cannot become 'less defined': if it was defined, it cannot become undefined because of the replacement.

The calculus determines these conditions from the uniqueness conditions of partially defined ι-terms. In our example, $\frac{y}{x}$ is defined as $\iota z_{y\neq 0}(z \cdot y = x)$. The uniqueness condition of this ι-term is $y \neq 0 \vdash_\iota \exists! z(z \cdot y = x)$, which we expect to be able to derive in a theory of real numbers.

All rules of the calculus guarantee that the sequents they produce only contain ι-terms for which the uniqueness conditions are derivable. Hence, if we were to consider a ι-term whose uniqueness condition is not derivable, such as $\iota z(z \cdot y = x)$, we are not able to prove anything about it.

Another property of the calculus is that the only 'source of undefinedness' are the ι-terms. Since we know explicitly when these are or aren't defined (we just have to look at the domain formula of the ι-term), we can always mechanically compute a formula indicating when a given term or formula is (un)defined. We call this formula the definedness of the given term or formula and note the operation of calculating it as $\mathbf{\Delta}$.

Being able to mechanically compute the definedness of a formula is one of the reasons we note the domain formula of a ι-term explicitly. Moreover, ι-terms with the same definiens but different domain formulae can occur. For example, we can also reason about the ι-term $\iota z_{x>0 \& y>0}(z \cdot y = x)$, which represents division restricted to positive numbers. In general, one can always restrict the domain of a ι-term arbitrarily; the extreme is $\iota z_{\forall x(x\neq x)}(\varphi)$, which is always undefined (and whose uniqueness condition is always derivable for any formula φ).

Looking at the substitution rules, we see that they have to guarantee that the uniqueness conditions of $[t/x]\alpha$ must be derivable when those of α and t are derivable. This requirement has its implications on the definition of substitution: in some cases, an extra $\mathbf{\Delta}(t)$ appears in the domain formula of

the newly created ι-terms.

The substitution rules also must maintain the restriction that the newly created ι-terms only occur in situations where their domain formulae will be fulfilled; hence, an extra $\mathbf{\Delta}(t)$ appears also in the context of the newly derived sequent.

We also investigate the addition of partial function and predicate symbols to the calculus. We again have to put some restrictions on the partial functions and predicates we admit: they have to be defined by a formula or term already present in the calculus. This ensures that we again can mechanically compute the definedness of a formula and prevents one from introducing non-monotone functions or predicates.

Adding non-**strict** functions and predicates requires us to introduce a more refined variant of the substitution operation and the substitution deduction rules. The main reason is that the primitive rules of the system add $\mathbf{\Delta}(t)$ to the resulting term or formula 'too soon' in the evaluation order when substituting a term t for a variable (see the beginning of chapter 4 for an example).

6.3 Example

In [Suppes 1972], a book about Zermelo-Fraenkel set theory, set abstraction is defined by

$$\{x : \varphi(x)\} := y \Leftrightarrow [(\forall x(x \in y \Leftrightarrow \varphi(x))) \,\&\, y \text{ is a set}] \vee [y = \emptyset \& \neg\exists B \forall x(x \in B \Leftrightarrow \varphi(x))]$$

In other words, if the set $\{x : \varphi(x)\}$ is a set containing exactly those x satisfying $\varphi(x)$, except when there exists no such set, in which case the 'convenient value' $\emptyset$ is chosen. This leads to the strange situation that two theorems are proved in [Suppes 1972]:

$$\emptyset = \{x : x \neq x\} \qquad \text{and} \qquad \emptyset = \{x : x = x\}$$

The first one is what one would expect: the empty set contains all objects satisfying a contradictory condition, in other words, no elements at all. The second one is a 'pathological theorem' as we called it in the introduction: the universal set is too big to be a Zermelo-Fraenkel set, so the set abstraction construct evaluates to the 'convenient value' $\emptyset$.

Let us investigate how the PITFOL' calculus handles this situation. We define the set abstraction as

$$\{x : p(x)\} = \iota y_{\exists y \forall x(x \in y \Leftrightarrow p(x))}(\forall x(x \in y \Leftrightarrow p(x)))$$

The definition rule then yields

$$\exists y \forall x(x \in y \Leftrightarrow x \neq x) \vdash_{\iota'} \{x : x \neq x\} = \iota y_{\exists y \forall x(x \in y \Leftrightarrow x \neq x)}(\forall x(x \in y \Leftrightarrow x \neq x))$$

One easily shows that $x \in y \Leftrightarrow x \neq x$ and $x \notin y$ are interchangeable, so we can derive

$$\exists y \forall x(x \notin y) \vdash_{\iota'} \{x : x \neq x\} = \iota y_{\exists y \forall x(x \notin y)}(\forall x(x \notin y))$$

Since $\vdash_{\iota'} \exists y \forall x(x \notin y)$ is an axiom of Zermelo-Fraenkel set theory (expressing the existence of an empty set), we can easily derive

$$\vdash_{\iota'} \{x : x \neq x\} = \iota y(\forall x(x \notin y))$$

and after adding the definition $\emptyset = \iota y(\forall x(x \notin y))$ we indeed obtain

$$\vdash_{\iota'} \{x : x \neq x\} = \emptyset$$

On the other hand, another application of the definition rule yields

$$\exists y \forall x(x \in y \Leftrightarrow x = x) \vdash_{\iota'} \{x : x = x\} = \iota y_{\exists y \forall x(x \in y \Leftrightarrow x = x)}(\forall x(x \in y \Leftrightarrow x = x))$$

which we easily can transform into

$$\exists y \forall x(x \in y) \vdash_{\iota'} \{x : x = x\} = \iota y_{\exists y \forall x(x \in y)}(\forall x(x \in y))$$

However, $\vdash_{\iota'} \neg \exists y \forall x(x \in y)$ is a theorem of Zermelo-Fraenkel set theory, so the sequent obtained is 'vacuously valid', and the ι-term occurring in it is always undefined.

6.4 Implementing the calculus

As already stated in the introduction, a computerised implementation of the calculus is highly desirable, since writing and checking proofs by hand gets tedious quickly.

In this section, we will describe how one could go about implementing the PITFOL calculus.

6.4.1 LCF-style implementation

The kernel of such an implementation can be rather small: we only need the 17+4 primitive deduction rules stated at the beginning of this chapter.

Next to the classical ingredients such as a representation for terms, formulae, lists of formulae, sequents, ..., this also requires an implementation of $\mathbf{\Delta}$, substitution, simultaneous substitution, alphabetic variants of a formula, and the generation of the list of uniqueness conditions of the top-level ι-terms of a formula: all of these are not hard to implement (we have done this in an experimental implementation).

All other rules can then be implemented on top of this kernel: in the resulting system, sequents are only produced by the routines corresponding to the primitive deduction rules. In other words, a derived rule can use the primitive rules or other derived rules (which will in eventually result in a number of calls to the primitive rules), but not create sequents at random.

This is the so called LCF approach to theorem proving: supposing that the kernel is implemented correctly (which is doable, since it is as indicated a very small piece of software), if one would make a programming mistake in a derived rule, one can never derive invalid sequents. The kind of errors that can happen in the implementation of derived rules are typically attempts to apply a primitive rule when it is not applicable, in which case the error is detected by the kernel of the system.

One could even use the mechanisms of expansion and translation of proofs to transform all generated proofs back into proofs of the classical calculus and use a classical theorem checker to check all proofs generated by the system.

6.4.2 Implementation on top of another system

Since the PITFOL calculus is an extension of the classical first order logic, one can take an existing first order prover and extend its rules.

The advantage of this approach is that one makes use of a (hopefully) well-tested system with a large library of theorems and derived rules.

The drawback is that the existing theorems and derived rules will not make use of the facilities for partially defined terms provided by our calculus, so one will have to adapt these too.

6.5 Future Research

Firstly, the further elaboration of the existing experimental implementation of the PITFOL system would be desirable: the best way to verify that a system is "reasonably faithful to mathematical practice" is to see whether mathematicians want to actually use it!

A computerised implementation calls for a proof procedure that is able to handle trivial proofs automatically. The adaptation of **tableau method**s

to the PITFOL calculus seems a promising area of research.

We expect that we can exploit the tableau method to prove that one needs the Definition rule for each combination of arguments, i.e., that just adding

$$\boldsymbol{\Delta}'(g^*) \vdash_{\iota'} g(x_1, \ldots, x_n) = \widetilde{g^*}$$

and then using a substitution rule to try and get

$$\boldsymbol{\Delta}'(g(t_1, \ldots, t_n)) \vdash_{\iota'} g(t_1, \ldots, t_n) = \left[\!\!\left[\begin{matrix} t_1 & \cdots & t_n \\ x_1 & & x_n \end{matrix} \right]\!\!\right] \widetilde{g^*}$$

is in general not possible.

It would also be interesting to investigate the use of ε-terms in addition to or in replacement of ι-terms, since one can consider the ε operator as a stronger version of the ι operator. One would expect that instead of the uniqueness conditions, an ε-term $\varepsilon x_\psi(\varphi)$ would only need an existence condition $\psi \vdash \exists x(\varphi)$, that the rule

$$\boxed{\text{epsilon}} \quad \frac{\psi \vdash_\iota \exists x(\varphi)}{\psi \vdash_\iota [\varepsilon x_\psi(\varphi)/x]\widetilde{\varphi}}$$

is added, ... but we expect that the general framework of the calculus will not have to be changed radically to accommodate these changes.

Summary

In *chapter 1*, we introduce the concept of formalisation of mathematics, and indicate the need for a treatment of partially defined functions which is faithful to mathematical practice.

In *chapter 2*, we describe our starting point: the well known first order predicate calculus with identity. This calculus has no support for partially defined functions.

In *chapter 3*, we add support for partially defined functions.

Section 3.1 introduces the way we will let undefinedness enter the calculus: we extend Hilbert and Bernays' method of adding iota terms to the calculus to adding partially defined iota terms.

In *section 3.2*, syntactical issues are handled: we introduce the metalogical definedness operator $\mathbf{\Delta}$ and define the way partially defined iota terms behave under substitutions.

In *section 3.3*, we discuss how undefined values are treated semantically in the calculus: the basic idea is that when one respects a left-to-right evaluation order, one will never encounter an undefined value when evaluating a valid sequent.

In *sections 3.4*, and *3.5*, we introduce deduction rules for our calculus. These are an extension of the rules of the calculus of chapter 2.

In *section 3.6*, we show that our calculus is consistent, provided the original calculus was consistent. This means that we prove that using the deduction rules, one cannot derive a contradiction (i.e., a proof of a theorem and its negation). We do this by showing that each proof in our calculus can be translated into a similar proof in the original calculus; if one would obtain a proof of contradiction in our calculus, by translating it, one would also have a proof of a contradiction in the original calculus.

In *section 3.7*, we show that our calculus is sound with respect to the semantics given. This means that using the deduction rules, one cannot derive a false theorem.

In *section 3.8*, we introduce a number of derived rules, which make the calculus easier to use for the sequel. Each application of a derived rule can be considered as an abbreviation of an application of a number of earlier defined rules.

In *section 3.9*, we show that our calculus is complete, provided the original one was complete: if a theorem is true, one must be able to produce a derivation of it using the deduction rules. We prove this by transforming the problem into a problem of the original calculus and exploiting its assumed completeness.

In *chapter 4*, we introduce a more sophisticated variant of substitution, which will be useful in the following section. Our original substitution is syntactically simpler; the variant is semantically subtler.

Chapter 5 discusses the addition of defined symbols to the calculus, which makes it possible to add definitions without affecting the consistency, soundness and completeness of the calculus.

In *chapter 6*, we give a concise overview of the PITFOL calculus and its treatment of partially defined terms, indicate how one could implement the calculus and suggest topics for future research.

Samenvatting

Je n'ai fait celle-ci plus longue
que parce que je n'ai pas eu le loisir
de la faire plus courte.
—Blaise Pascal, "Lettres provinciales, Lettre xvi"

In *hoofdstuk 1* stellen we het concept formele wiskunde voor, en wijzen we op de nood aan een ondersteuning voor partieel gedefinieerde functies die getrouw is aan de gangbare wiskundige praktijk.

In *hoofdstuk 2* beschrijven we ons vertrekpunt: de bekende eerste orde predicaatrekening met gelijkheid. Deze calculus bevat geen ondersteuning voor partieel gedefinieerde functies.

In *hoofdstuk 3* voegen we aan deze calculus ondersteuning toe voor partieel gedefinieerde functies.

Sectie 3.1 brengt de wijze aan waarop we ongedefinieerdheid toevoegen aan de calculus: we breiden Hilbert en Bernays' methode voor het toevoegen van iota-termen aan de calculus uit tot het toevoegen van partieel gedefinieerde iota-termen.

In *sectie 3.2* behandelen we syntactische onderwerpen: we definiëren de metalogische gedefineerdheids-operator $\mathbf{\Delta}$ en leggen vast hoe partieel gedefinieerde iota-termen zich gedragen onder substituties.

In *sectie 3.3* tonen we hoe ongedefinieerde waarden semantisch verwerkt worden in de calculus: het basisidee is dat er nooit een ongedefinieerde waarde zal optreden wanneer een geldige sequent 'van links naar rechts' geëvalueerd wordt.

In *secties 3.4* en *3.5* stellen we de afleidingsregels van onze calculus voor. Ze zijn een uitbreiding van de regels van de calculus uit hoofdstuk 2.

In *sectie 3.6* tonen we aan dat onze calculus consistent is, vooropgesteld dat de originele calculus dat ook was. Dit betekent dat we bewijzen dat door gebruik te maken van de afleidingsregels nooit een contradictie (d.i. een bewijs van een stelling en haar ontkenning) bekomen kan worden. We

bewijzen dit door aan te tonen dat elk bewijs in onze calculus vertaald kan worden naar een gelijkaardig bewijs in de oorspronkelijke calculus; als we dus een bewijs van een contradictie zouden kunnen bekomen in onze calculus, dan zouden we door dat bewijs te vertalen ook een bewijs van een contradictie vinden in de originele calculus.

In *sectie 3.7* tonen we aan dat onze calculus betrouwbaar is ten opzichte van de eerder gegeven semantiek. Dit betekent dat het onmogelijk is een onware stelling af te leiden met behulp van de afleidingsregels.

In *sectie 3.8* voeren we een aantal afgeleide regels in, die het gebruik van de calculus makkelijker zullen maken in de volgende secties. Elke toepassing van een afgeleide regel kan opgevat worden als een afkorting van een aantal eerder gedefinieerde regels.

In *sectie 3.9* tonen we aan dat onze calculus volledig is, vooropgesteld dat de originele calculus dat ook was: als een stelling waar is, moet ze ook afleidbaar zijn met behulp van de afleidingsregels. We bewijzen dit door het probleem om te zetten in een probleem van de originele calculus en de volledigheid van de orignele calculus uit te buiten.

In *hoofdstuk 4* ontwikkelen we een meer gesofistikeerde variant van de substitutie, die van pas zal komen in de volgende sectie. Onze oorspronkelijke substitutie is syntactisch eenvoudiger; de variant is semantisch subtieler.

Hoofdstuk 5 behandelt het toevoegen van gedefinieerde symbolen aan de calculus, waardoor definities kunnen toegevoegd worden zonder de consistentie, betrouwbaarheid en volledigheid van de calculus in het gedrang te brengen.

In *hoofdstuk 6* geven we een beknopt overzicht van de PITFOL calculus en de manier waarop partieel gedefinieerde termen behandeld worden, geven aan hoe de calculus geïmplementeerd kan worden en stellen onderwerpen voor toekomstig onderzoek voor.

Bibliography

[Farmer 1990] W.M. Farmer, A Partial Functions Version of Church's Simple Theory of Types, The Journal of Symbolic Logic, Vol. 55, No. 3 (Sep., 1990), 1269–1291.

[Fateman 1994] R.J. Fateman, On the Design and Construction of Algebraic Maniplulation Systems, http://www.cs.berkeley.edu/~fateman/papers/asmerev94.ps

[Hazewinkel 2003] M. Hazewinkel, Mathematical Knowledge Management is needed, Keynote speech at the November 2003 MKM meeting in Edinburg), http://www.macs.hw.ac.uk/~fairouz/ mkm-symposium03/abstracts/michiel-hazewinkel.pdf

[Harrison 1996] J. Harrison, Pure Mathematics in a Mechanized Logic, http://www.cl.cam.ac.uk/~jrh13/papers/fais.html, Proceedings of the Finnish Artificial Intelligence Society (FAIS) symposium Logic, Mathematics and the Computer: History, Foundations and Applications, University of Helsinki, Metsätalo, 3–4 June 1996, pp. 153-169.

[Harrison 1996b] J. Harrison, Formalized Mathematics, Technical Report 36, Turku Centre for Computer Science (TUCS), http://www.cl.cam.ac.uk/~jrh13/papers/form-math3.html.

[Hermes 1973] H. Hermes, Introduction to Mathematical Logic, Springer, 1973.

[Hilbert & Bernays 1968] D. Hilbert, P. Bernays, Grundlagen der Mathematik I, 2nd ed., Springer, 1968.

[Hoogewijs 1977] A. Hoogewijs, A calculus of partially defined predicates, Mathematical Scripts, Seminarie voor Algebra, Ghent University, 1977.

[Impens] C. Impens, Een hardnekkig misverstand, http://cage.ugent.be/~ci/impens.pdf

[Jones 1995] R.B. Jones, A Critique of the QED Manifesto, http://www.rbjones.com/rbjpub/logic/qedres08.htm

[Jones 1996] R.B. Jones, There Ain't No Such Thing As A Free Lunch (with partial functions). In William Farmer, Manfred Kerber, and Michael Kohlhase, editors, Proceedings of the CADE-13 Workshop on the Mechanization Of Partial Functions, pages 53–64, 1996.

[McCarthy 1967] J. McCarthy, A basis for a mathematical theory of computation, In P. Braffort and D. Hirschberg, editors, Computer Programming and Formal Systems, North-Holland Publishing Company, 1967.

[QED 1994] The QED Manifesto, "Automated Deduction—CADE 12", Springer (1994), Lecture Notes in Artificial Intelligence, Vol. 814, pp. 238–251.

[Suppes 1972] P. Suppes, Axiomatic Set Theory, Dover, 1972.

[Vernaeve & Hoogewijs 2006] G. Vernaeve, A. Hoogewijs, A formal treatment of partial iota terms, Bulletin of symbolic logic Vol 12, no 2, 2006, 349

[Vernaeve & Hoogewijs 2007a] Extending a first order predicate calculus with partially defined iota terms. Bull. Belg. Math. Soc. Simon Stevin, Volume 13, Number 5 (2007), 917–930.

[Vernaeve & Hoogewijs 2007b] A tale of two substitutions, Bulletin of symbolic logic Vol 13, no 2, 2007, 291

[Wiedijk & Zwanenburg 2003] First Order Logic with Domain Conditions. In: David Basin & Burkhart Wolff (eds.), Theorem Proving in Higher Order Logics, Proceedings of TPHOLs 2003, Springer LNCS 2758, 221–237, 2003, http://www.cs.ru.nl/~freek/pubs/partial.pdf.

Contents

Index

Symbols

A

B

C

D

E

F

H

I

L

M

N

O

P

R

S

T