# Induction as Deduction Modulo

Eric Deplagne, Claude Kirchner

# Induction as Deduction Modulo

Eric Deplagne Claude Kirchner

*LORIA & INRIA*
*615, rue du Jardin Botanique. B.P. 101*
*54602 Villers-lès-Nancy Cedex, France*
`First.Last@loria.fr`

**Abstract**

Inductive proofs can be built either explicitly by making use of an induction principle or implicitly by using the so-called induction by rewriting and inductionless induction methods. When mechanizing proof construction, explicit induction is used in proof assistants and implicit induction is used in rewrite based automated theorem provers. The two approaches are clearly complementary but up to now there was no framework able to encompass and to understand uniformly the two methods. In this paper, we propose such an approach based on the general notion of deduction modulo. We extend slightly the original version of the deduction modulo framework and we provide modularity properties for it. We show how this applies to a uniform understanding of the so called induction by rewriting method and how this relates directly to the general use of an induction principle.

*Key words:* Induction, rewriting, deduction modulo

## 1 Introduction

Proof by induction is a fundamental proof method in mathematics. Since the emergence of computer science, it has been studied and used as one of the fundamental concepts to build mathematical proofs in a mechanized way. In the rising era of verified software and systems it plays a fundamental role in frameworks allowing to search for formal proofs. Therefore proofs by induction have a critical role in proof assistants and automated theorem provers. Of course these two complementary approaches of proof building use induction in very different ways. In proof assistants like COQ, ELF, HOL, Isabelle, Larch, NQTHM, PVS, ... induction is used explicitly since the induction axiom is applied in an explicit way: the human user or a clever tactics should find the right induction hypothesis as well as the right induction variables and patterns

*Preprint submitted to Elsevier Science* *27 November 2004*

to conduct the induction steps. In automated theorem provers specific methods have been developed to automatically prove inductive properties. The most elaborated ones are based on term rewriting and saturation techniques. They are respectively called induction by rewriting and inductionless induction or proof by consistency. Systems that implement these ideas are Spike, RRL or INKA.

These last methods have been studied since the end of the seventies and have shown their strengths on many practical examples from simple algebraic specifications to more complicated ones like the Gilbreath card trick. But what was intriguing from the conceptual point of view was the relationship between explicit and implicit induction: implicit induction was shown to prove inductive theorems, but the relationship with the explicit use of the induction principle was open.

*Our contributions.* We provide a framework to understand *both* approaches in a *unified* way. One important consequence is that it allows us to combine in a well-understood way automated and assisted proof search methods. This reconciliation of the two approaches will allow automated theorem provers and proof assistants to collaborate in a safe way. It will also allow proof assistants to embark powerful proof search tactics corresponding to implicit induction techniques. This corresponds to the deduction versus computation scheme advocated in [DHK03] [1] under the name of deduction modulo: we want some computations to be made blindly i.e. without the user interaction and in this case this corresponds to implicit induction; but one also needs to explicitly control deduction, first because we know this is unavoidable but also because this may lead to more efficient proof search.

It is thus not surprising to have our framework based on deduction modulo. This presentation of first-order logic relies on the sequent calculus modulo a congruence defined on terms and propositions. But since we need to formalize the induction axiom which is in essence a second-order proposition, we need to use the first-order representation of higher-order logic designed in [DHK01]. In this formalism, switching from explicit induction to implicit one becomes clear and amounts to push into the congruence some of the inductive reasoning, then to apply standard automated reasoning methods to simplify the goal to be proved and possibly get a better representation of the congruence.

This paper relies on the notions and notations of deduction modulo [DHK03] as well as on the first-order presentation of higher-order logic called $HOL_{\lambda\sigma}$ presented in [DHK01]. We refer to these two papers for full definitions, details and motivations of the framework. In this context, our contributions are the following:

[1] Before publication, this work was available as [DHK98].

(1) At the conceptual level, we provide through deduction modulo a uniform framework to combine automated and assisted proof tools.
(2) This is used as a uniform framework that allows us to combine explicit and implicit induction mechanisms.
(3) We extend the formalism of deduction modulo introduced in [DHK03] and used in [DHK01] to *conditional* axioms and *conditional* rewrite rules. We also give crucial modularity properties of the deduction modulo paradigm.
(4) We show how rewrite based techniques based on orderings help to simplify the application of the induction hypothesis.
(5) Using this new framework, we uniformly review the induction by rewriting method and show how it directly relates to the induction principle, thus providing proof theoretic instead of model theoretic proofs of this rewrite based method.

Consequently, since the proof method is completely proof theoretic, to any rewrite based inductive proof we can canonically associate an explicit proof in the sequent calculus, thus providing a proof assistant with all necessary information to replay the proof as needed.

*Related works.* The use of rewriting techniques to perform inductive proofs consists of two quite different methods. The first is based on a consistency check and is also call inductionless induction after [Lan81]. Because it is mainly based on a saturation procedure which aims to detect inconsistency, it is quite different from the second method, often called term rewriting induction that uses mainly the fact that a terminating relation determines an induction principle that is implicitly applied when rewriting. This second method has been historically derived from the first one in [HK88] and fully developed under the name of "term rewriting induction" in [Red90] using the so called "cover sets" and in a series of papers initiated by [KR90a,KR90b] and using "test sets".

Inductionless induction can be explained in deduction modulo as well, but we focus in this paper on term rewriting induction. It should be noticed that our approach and results are quite different from [Red90] since in particular we explicitly relate the induction principle and its use in the presentation of first-order logic provided by the sequent calculus to the direct use of conditional rewriting. In particular this can be spotted in the semantic definition of what inductive consequences are in [Red90][def. 1, page 164], where we use a completely proof theoretic definition of such consequences.

*Roadmap.* Section 2 first recalls the main notations and concepts. It extends the deduction modulo paradigm to deal with congruences defined conditionally and also proves the modularity results.

Section 3 is fully devoted to recalling the first-order presentation of higher-order logic based on explicit substitutions.

Section 4 shows how the induction axiom, which is most naturally expressed in second-order logic, can be encoded in deduction modulo (i.e. at first-order) and how the induction hypothesis can be internalized in the congruence modulo which the sequent calculus works. It also provides useful sufficient conditions to avoid checking ordering conditions when applying such an induction hypothesis.

Section 5 shows how the framework can be applied to uniformly review the proof by rewriting method.

We finally conclude and provide tracks for further researches in section 6.

## 2 Deduction Modulo

We use a presentation of first-order logic, called *deduction modulo*, where terms as well as propositions can be identified modulo an equivalence relation [DHK03]. In deduction modulo, the notions of term and proposition are that of many sorted first-order logic. We consider theories formed with a set of axioms $\Gamma$ and an equivalence relation, denoted by $\sim$, defined on terms and propositions and which can typically be defined by *conditional* equations.

Of course, the equivalence has an impact on the formulation of the sequent calculus itself. The resulting calculus is called sequent calculus modulo and is presented in section 2.1. We use the notation $\Gamma \vdash \Delta$ to denote a derivable sequent in the standard sequent calculus [GLT89,Gal86], and the notation $\Gamma \vdash_\sim \Delta$ to denote a derivable sequent in the sequent calculus modulo. To take the equivalence into account, each rule of the sequent calculus is modified, for example:

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$$

becomes

$$\frac{\Gamma \vdash_\sim A, \Delta \quad \Gamma \vdash_\sim B, \Delta}{\Gamma \vdash_\sim D, \Delta} \text{ if } D \sim A \wedge B$$

to take into account the fact that $D$ does not necessarily have the syntactical form $A \wedge B$ but is only equivalent to it.

### *2.1 Sequent calculus modulo*

We consider a set of fixed arity function symbols $\mathcal{F}$, a set of fixed arity predicate symbols $\mathcal{P}$ and a set of variables $\mathcal{X}$ with no symbol in common. $\mathcal{T}(\mathcal{F}, \mathcal{X})$

denotes the set of terms built on $\mathcal{F}$ and $\mathcal{X}$. We call $\mathcal{AP}(\mathcal{P}, \mathcal{F}, \mathcal{X})$ the set of atomic propositions built on the set of predicates $\mathcal{P}$ and the set of terms $\mathcal{T}(\mathcal{F}, \mathcal{X})$. $\mathcal{P}rop(\mathcal{P}, \mathcal{F}, \mathcal{X})$ is the set of first-order propositions build on the atomic propositions in $\mathcal{AP}(\mathcal{P}, \mathcal{F}, \mathcal{X})$ with the usual connectors $\wedge, \vee, \Rightarrow, \neg, \bot$ and quantifiers $\forall$ and $\exists$.

We give in figure 1, the definition of the *sequent calculus modulo*. It extends the usual calculus [GLT89,Gal86] by working modulo the equivalence $\sim$ and the sequent calculus modulo given in [DHK03] by allowing the equivalence to be defined conditionally. In the rules, $\Gamma$ and $\Delta$ are finite multisets of propositions; $P$, $Q$ and $R$ denote propositions. When the equivalence $\sim$ is simply identity, this sequent calculus collapses to the usual one. In that case sequents are written as usual with the $\vdash$ symbol.

In order to evaluate the conditions, we may also have to take into account the *context* in which the equivalence is used. Therefore the equivalence takes three arguments: the two objects to be compared and a set of axioms $\Gamma$ called a local context. When we want to emphasize the presence of a context we add it to the notation, and denote the equivalence by $\sim^{\Gamma}$.

Proof checking decidability for the sequent calculus modulo reduces to the decidability of the equivalence, since we can check for each rule that the conditions of application are satisfied and provide the needed information in the quantifier rules.

When the equivalence is decidable, to check the equivalence of two terms or of two propositions is just computation. In this case deduction modulo allows to draw a clear separation between deduction (in general undecidable) and a decidable part of the reasoning process seen as computation.

When the equivalence is not decidable, the use of constraints allows us to proceed, using the solution of easy to solve constraints and preserving the others, in the same way as one deals with higher order unification constraints in constraint resolution [Hue72].

$$\frac{}{\Gamma, P \vdash_\sim Q}\text{axiom} \quad \text{if } P \sim^\Gamma Q \qquad\qquad \frac{\Gamma, P \vdash_\sim \Delta \quad \Gamma \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim \Delta}\text{cut} \quad \text{if } P \sim^\Gamma Q$$

$$\frac{\Gamma, Q, R \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}\text{contr-l} \quad \text{if } (\mathbf{A}) \qquad\qquad \frac{\Gamma \vdash_\sim Q, R, \Delta}{\Gamma \vdash_\sim P, \Delta}\text{contr-r} \quad \text{if } (\mathbf{A})$$

$$\frac{\Gamma \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}\text{weak-l} \qquad\qquad \frac{\Gamma \vdash_\sim \Delta}{\Gamma \vdash_\sim P, \Delta}\text{weak-r}$$

$$\frac{\Gamma, P, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\wedge\text{-l} \quad \text{if } (\mathbf{B}) \qquad\qquad \frac{\Gamma \vdash_\sim P, \Delta \quad \Gamma \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\wedge\text{-r} \quad \text{if } (\mathbf{B})$$

$$\frac{\Gamma, P \vdash_\sim \Delta \quad \Gamma, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\vee\text{-l} \quad \text{if } (\mathbf{C}) \qquad\qquad \frac{\Gamma \vdash_\sim P, Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\vee\text{-r} \quad \text{if } (\mathbf{C})$$

$$\frac{\Gamma \vdash_\sim P, \Delta \quad \Gamma, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\Rightarrow\text{-l} \quad \text{if } (\mathbf{D}) \qquad\qquad \frac{\Gamma, P \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\Rightarrow\text{-r} \quad \text{if } (\mathbf{D})$$

$$\frac{\Gamma \vdash_\sim P, \Delta}{\Gamma, R \vdash_\sim \Delta}\neg\text{-l} \quad \text{if } R \sim^\Gamma \neg P \qquad\qquad \frac{\Gamma, P \vdash_\sim \Delta}{\Gamma \vdash_\sim R, \Delta}\neg\text{-r} \quad \text{if } R \sim^\Gamma \neg P$$

$$\frac{}{\Gamma, P \vdash_\sim \Delta}\bot\text{-l} \quad \text{if } P \sim^\Gamma \bot$$

$$\frac{\Gamma, Q\{t/x\} \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}(Q, x, t)\ \forall\text{-l} \quad \text{if } (\mathbf{E}) \qquad\qquad \frac{\Gamma \vdash_\sim Q\{y/x\}, \Delta}{\Gamma \vdash_\sim P, \Delta}(Q, x, y)\ \forall\text{-r} \quad \text{if } (\mathbf{F})$$

$$\frac{\Gamma, Q\{y/x\} \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}(Q, x, y)\ \exists\text{-l} \quad \text{if } (\mathbf{G}) \qquad\qquad \frac{\Gamma \vdash_\sim Q\{t/x\}, \Delta}{\Gamma \vdash_\sim P, \Delta}(Q, x, t)\ \exists\text{-r} \quad \text{if } (\mathbf{H})$$

$$\mathbf{A} = P \sim^\Gamma Q \text{ and } P \sim^\Gamma R$$

$$\mathbf{B} = R \sim^\Gamma (P \wedge Q) \quad \mathbf{C} = R \sim^\Gamma (P \vee Q), \quad \mathbf{D} = R \sim^\Gamma (P \Rightarrow Q)$$

$$\mathbf{E} = P \sim^\Gamma \forall x\ Q, \quad \mathbf{F} = P \sim^\Gamma \forall x\ Q, y \text{ fresh variable}$$

$$\mathbf{G} = P \sim^\Gamma \exists x\ Q, y \text{ fresh variable}, \quad \mathbf{H} = P \sim^\Gamma \exists x\ Q$$

Fig. 1. The sequent calculus modulo

### *2.2 Relating deduction modulo and classical deduction: compatibility*

To understand the relationship between deduction modulo and standard deduction, we first introduce the fundamental notion of *compatibility.* A theory $\mathcal{T}$ and an equivalence $\sim$ are compatible when deduction modulo $\sim$ is equivalent to deduction with $\mathcal{T}$ as hypothesis. But this equivalence may depend on a context $\mathcal{U}$, therefore leading to following definition:

**Definition 2.1** A set of axioms $\mathcal{T}$ and an equivalence $\sim$ are *compatible up to* a set of axioms $\mathcal{U}$ when, for all $\Gamma, \Delta$:

$$\mathcal{T}, \mathcal{U}, \Gamma \vdash \Delta \text{ if and only if } \mathcal{U}, \Gamma \vdash_{\sim} \Delta.$$

Checking compatibility can be achieved by enforcing the following two conditions on the equivalence relation:

- First ensure that any deduction using the equivalence can be reproduced without it, namely:

$$\mathcal{T}, \mathcal{U}, \Gamma \vdash \Delta \text{ if and only if } \mathcal{T}, \mathcal{U}, \Gamma \vdash_{\sim} \Delta.$$

- Second, ensure that the theory $\mathcal{T}$ is not needed in the calculus modulo, namely:

$$\mathcal{T}, \mathcal{U}, \Gamma \vdash_{\sim} \Delta. \text{ if and only if } \mathcal{U}, \Gamma \vdash_{\sim} \Delta.$$

  Of course $\mathcal{T}$ will remain needed in practice if the equivalence is not decidable or not tractable.

This first condition is ensured when there exists a proof of the equivalence of two propositions only by logical means. This is formalized by the following property:

**Lemma 2.1** Let $\mathcal{T}$ and $\sim$ be a theory and an equivalence. The two following statements are equivalent:

(A) for all propositions $P$ and $Q$ and context $\Gamma$,

$$P \sim^{\Gamma} Q \text{ implies } \mathcal{T}, \mathcal{U}, \Gamma \vdash P \Leftrightarrow Q,$$

(B) $\mathcal{T}, \mathcal{U}, \Gamma \vdash \Delta$ if and only if $\mathcal{T}, \mathcal{U}, \Gamma \vdash_{\sim} \Delta$.

**Proof:** See appendix A □

**Remark 2.1** The proof from A to B extensively uses the monotonicity of the logic, embodied by the weakening and contraction rules.

The proof from B to A uses the reflexivity and symmetry properties of the equivalence, but transitivity is not used.

The second condition is that the axioms of $\mathcal{T}$ have to be tautologies in the calculus modulo. The following proposition asserts that this condition allows working with or without the axioms of $\mathcal{T}$ in the calculus modulo.

**Lemma 2.2** Let $\mathcal{T}$ and $\sim$ be a theory and an equivalence. The two following statements are equivalent:

(1) for every proposition $P$ in $\mathcal{T}$, we have $\mathcal{U} \vdash_{\sim} P$,
(2) $\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} \Delta$ if and only if $\mathcal{U},\Gamma \vdash_{\sim} \Delta$.

**Proof:**

- From 1 to 2, it is obvious.
- From 2 to 1, $\mathcal{U} \vdash_{\sim} P$ reduces to $\mathcal{T},\mathcal{U} \vdash_{\sim} P$ by 2, which is true by axiom since $P$ is in $\mathcal{T}$.

□

**Remark 2.2** The proof from 1 to 2 uses the weakening rule of the logic.

The proof from 2 to 1 uses the reflexivity property of the equivalence.

The following proposition sums up the two previous lemmas to asserts that the conjunction of the two conditions is equivalent to compatibility. It is the direct consequence of the two previous propositions.

**Proposition 2.1** A set of axioms $\mathcal{T}$ and an equivalence $\sim$ are *compatible up to* a set of axioms $\mathcal{U}$ iff both of the following conditions are met:

(A) for all propositions $P$ and $Q$ and context $\Gamma$,
$P \sim^{\Gamma} Q$ implies $\mathcal{T},\mathcal{U},\Gamma \vdash P \Leftrightarrow Q$,
(B) for every proposition $P$ in $\mathcal{T}$, we have $\mathcal{U} \vdash_{\sim} P$.

**Remark 2.3** In [DHK03], the conditions A and B above are given as the definition of compatibility.

Using compatibility, we can internalize propositions into the equivalence, and call this operation "push". We can also recover them at the level of logic, and call this operation "pop". We formalize this in the form of inference rules, where the theory $\mathcal{T}$ and the equivalence $\sim$ are compatible:

$$\frac{\Gamma \vdash_{\sim} \Delta}{\Gamma, \mathcal{T} \vdash \Delta}\ \text{push} \qquad\qquad \frac{\Gamma, \mathcal{T} \vdash \Delta}{\Gamma \vdash_{\sim} \Delta}\ \text{pop}$$

### *2.3 Modularity of compatibility*

Because induction hypotheses arise dynamically during the proof process, we need a modularity result to be able to add them to the equivalence.

**Definition 2.2** Let $\sim_1$ and $\sim_2$ be two equivalences, and $\Gamma$ be a given context.

The equivalence $(\sim_1 \cup \sim_2)^\Gamma$ is inductively defined by $a\,(\sim_1 \cup \sim_2)^\Gamma\, b$ if one of the following holds:

- $a \sim_1^\Gamma b$,
- $a \sim_2^\Gamma b$,
- there exists $c$ such that $a \sim_1^\Gamma c\,(\sim_1 \cup \sim_2)^\Gamma\, b$,
- there exists $c$ such that $a \sim_2^\Gamma c\,(\sim_1 \cup \sim_2)^\Gamma\, b$.

**Lemma 2.3** Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two sets of axioms, compatible respectively with the relations $\sim_1$ and $\sim_2$ up to the set of axioms $\mathcal{U}_1$ and $\mathcal{U}_2$. The relation $\sim_1 \cup \sim_2$ is compatible with $\mathcal{T}_1 \cup \mathcal{T}_2$ up to $\mathcal{U}_1 \cup \mathcal{U}_2$.

**Proof:** We shall prove that:

(A) for all propositions $P$ and $Q$ and contexts $\Gamma$,
$P\,(\sim_1 \cup \sim_2)^{\mathcal{U}_1,\mathcal{U}_2,\Gamma}\, Q$ implies $\mathcal{T}_1, \mathcal{T}_2, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash P \Leftrightarrow Q$.

We proceed by cases, according to definition 2.2.

- if $P \sim_1^{\mathcal{U}_1,\mathcal{U}_2,\Gamma} Q$, from the compatibility of $\mathcal{T}_1$ and $\sim_1$ up to $\mathcal{U}_1$ we get that for all $\Gamma'$, $P \sim_1^{\mathcal{U}_1,\Gamma'} Q$ implies $\mathcal{T}_1, \mathcal{U}_1, \Gamma' \vdash P \Leftrightarrow Q$.

  If $\Gamma' = \mathcal{U}_2, \Gamma$, we get that $P \sim_1^{\mathcal{U}_1,\mathcal{U}_2,\Gamma} Q$ implies $\mathcal{T}_1, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash P \Leftrightarrow Q$.

  Monotonicity of the logic allows us to add $\mathcal{T}_2$ and thus to conclude.
- if $P \sim_2^{\mathcal{U}_1,\mathcal{U}_2,\Gamma} Q$, from the compatibility of $\mathcal{T}_2$ and $\sim_2$ up to $\mathcal{U}_2$ we get that for all $\Gamma'$, $P \sim_2^{\mathcal{U}_2,\Gamma'} Q$ implies $\mathcal{T}_2, \mathcal{U}_2, \Gamma' \vdash P \Leftrightarrow Q$.

  If $\Gamma' = \mathcal{U}_1, \Gamma$, we get that $P \sim_2^{\mathcal{U}_1,\mathcal{U}_2,\Gamma} Q$ implies $\mathcal{T}_2, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash P \Leftrightarrow Q$.

  Monotonicity of the logic allows us to add $\mathcal{T}_1$ and thus to conclude.
- if $P \sim_1^{\mathcal{U}_1,\mathcal{U}_2,\Gamma} R\,(\sim_1 \cup \sim_2)^{\mathcal{U}_1,\mathcal{U}_2,\Gamma} Q$, we have $\mathcal{T}_1, \mathcal{T}_2, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash P \Leftrightarrow R$ and by induction hypothesis $\mathcal{T}_1, \mathcal{T}_2, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash R \Leftrightarrow Q$ from which we easily deduce $\mathcal{T}_1, \mathcal{T}_2, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash P \Leftrightarrow Q$.
- if $P \sim_2^{\mathcal{U}_1,\mathcal{U}_2,\Gamma} R\,(\sim_1 \cup \sim_2)^{\mathcal{U}_1,\mathcal{U}_2,\Gamma} Q$, we have $\mathcal{T}_1, \mathcal{T}_2, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash P \Leftrightarrow R$ and by induction hypothesis $\mathcal{T}_1, \mathcal{T}_2, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash R \Leftrightarrow Q$ from which we easily deduce $\mathcal{T}_1, \mathcal{T}_2, \mathcal{U}_1, \mathcal{U}_2, \Gamma \vdash P \Leftrightarrow Q$.

(B) For every proposition $P$ in $\mathcal{T}_1 \cup \mathcal{T}_2$ we have $\mathcal{U}_1, \mathcal{U}_2 \vdash_{\sim_1 \cup \sim_2} P$.

We have two cases:

- either $P$ is in $\mathcal{T}_1$ and we have $\mathcal{U}_1 \vdash_{\sim_1} P$ and thus $\mathcal{U}_1, \mathcal{U}_2 \vdash_{\sim_1 \cup \sim_2} P$,
- or $P$ is in $\mathcal{T}_2$ and we have $\mathcal{U}_2 \vdash_{\sim_2} P$ and thus $\mathcal{U}_1, \mathcal{U}_2 \vdash_{\sim_1 \cup \sim_2} P$.

□

From the previous lemma we immediately get modularity:

**Corollary 2.1** Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two sets of axioms, compatible respectively with the equivalences $\sim_1$ and $\sim_2$ up to the set of axioms $\mathcal{U}_1$ and $\mathcal{U}_2$. Let $\sim_{1 \cup 2}$

be an equivalence compatible with $\mathcal{T}_1 \cup \mathcal{T}_2$ up to $\mathcal{U}_1 \cup \mathcal{U}_2$.

for any $\Gamma$,

$$\begin{array}{c}
\mathcal{U}_1,\mathcal{U}_2,\Gamma \vdash_{\sim_{1\cup 2}} \Delta \\
\Updownarrow \\
\mathcal{T}_2,\mathcal{U}_1,\mathcal{U}_2,\Gamma \vdash_{\sim_1} \Delta \;\Leftrightarrow\; \mathcal{T}_1,\mathcal{T}_2,\mathcal{U}_1,\mathcal{U}_2,\Gamma \vdash \Delta \;\Leftrightarrow\; \mathcal{T}_1,\mathcal{U}_1,\mathcal{U}_2,\Gamma \vdash_{\sim_2} \Delta \\
\Updownarrow \\
\Gamma \vdash_{\sim_1 \cup \sim_2} \Delta
\end{array}$$

This property allows us to simply take $\sim_{1\cup 2}$ to be $\sim_1 \cup \sim_2$. It also allows us to internalize only $\mathcal{T}_1$ or $\mathcal{T}_2$ or to internalize dynamically one then the other.

This will be important for induction since induction hypotheses arise dynamically during the proof.

### *2.4 Conditional protected equality*

The interaction between the definition of equality via rewriting and the explicit use of orderings brings us to control their relationship. Let us first motivate why. Leibniz equality is the most commonly equality and it is defined as a congruence such that two objects are equivalent when no predicate allows to distinguish them.

Leibniz equality is too strong when one wants to explicitly use predicates like $>$ representing the noetherian ordering needed for induction. This is because in such a situation, if we assume as usual the equality to be defined on naturals by $x+0 \rightarrow x$ and $x+s(y) \rightarrow s(x+y)$, then $s(x)+0$ is not Leibniz-equal to $s(x)$. Indeed, the predicate $>$ defined as a RPO (recursive path ordering [Der87]) using the precedence $+ > s > 0$, allows us to distinguish between $s(x)$ and $s(x) + 0$ as we have $s(x) + 0 > s(s(x))$ and $s(s(x)) > s(x)$.

We therefore want to define an equality $\sim$ such that $x + 0 \sim x$, while having $>$ as an explicit predicate. This is achieved by forbiding the context propagation of $\sim$ equality below the $>$ predicate. In this case, we call $>$ a *protective* predicate.

### *2.4.1 Protective symbols*

For a given equivalence $\sim$, a symbol can be protective with respect to all, none or some of its argument. This is reflected in the following definition:

**Definition 2.3** Given an equivalence $\sim$, to each function symbol $f$ of arity $n$ in $\mathcal{F}$, we associate a subset of $\{1,\dots,n\}$ that we call the *protective arity* of $f$, denoted by $\mathit{prarity}(f,\sim)$.

To each predicate symbol $P$ of arity $n$ in $\mathcal{P}$, we associate a subset of $\{1,\dots,n\}$ that we call the *protective arity* of $P$, denoted by $\mathit{prarity}(P,\sim)$.

If $\mathit{prarity}(f,\sim)$ is non-empty, $f$ is said to be a *protective symbol* for $\sim$. We call *protective signature* a signature with given non-empty protective arities for its function and predicate symbols.

**Remark 2.4** The notion of protective symbols [Dep02,DKKN03] is very similar to the operational notion of frozen symbols introduced independently in the Maude system and in rewriting logic [BMM02,BM03].

**Notation 2.1** *To improve readability,*

- $x_1,\dots,x_n$ *will be denoted by* $\overline{x}$,
- *we write* $Q(\overline{t},v_i,\overline{s})$ *if* $v_i$ *is the* $i^{th}$ *argument of* $Q$.

From the notion of protective symbols, we can define the notion of protected congruence, where stability by context is only true at unprotected positions.

**Definition 2.4** Given a protective signature, an equivalence $\sim$ is extended to the *protected congruence* $\sim_p$ defined by:

- $x \sim y \Rightarrow x \sim_p y$
- $(x \sim_p y \wedge y \sim_p z) \Rightarrow x \sim_p z$
- for all $f$ of arity $n$ in $\mathcal{F}$, for all $i$ in $\{1,\dots,n\} \setminus \mathit{prarity}(f,\sim)$,

$$x_i \sim_p y_i \Rightarrow f(\overline{u},x_i,\overline{v}) \sim_p f(\overline{u},y_i,\overline{v})$$

- for all $P$ of arity $n$ in $\mathcal{P}$, for all $i$ in $\{1,\dots,n\} \setminus \mathit{prarity}(P,\sim)$,

$$x_i \sim_p y_i \Rightarrow P(\overline{u},x_i,\overline{v}) \Leftrightarrow P(\overline{u},y_i,\overline{v})$$

**Example 2.1** Let $f$ be a binary function symbol such that $\mathit{prarity}(f,\sim) = \{1\}$, and $a \sim b$. We have $f(x,a) \sim_p f(x,b)$, but not $f(a,y) \sim_p f(b,y)$.

**Example 2.2** Coming back to the motivating example of this section, the predicate $<$ is made protective with respect to its two arguments: $\mathit{prarity}(<,\sim) = \{1,2\}$. Doing so, we get that $x+0 \sim x$ and $s(x)+0 > s(s(x))$

do not imply $s(x) > s(s(x))$, thus not conflicting with the subset ordering $s(x) < s(s(x))$.

### *2.4.2 Protected equality*

We now define the axioms of protected equality for an equality $\asymp$.

**Definition 2.5** A binary predicate $\asymp$ in $\mathcal{P}$ is said to be a protected equality when it satisfies the theory:

$$Th_{\asymp} \triangleq \begin{cases} \forall x\, (x \asymp x) \\ \forall x\, \forall y\, (x \asymp y \Rightarrow y \asymp x) \\ \forall x\, \forall y\, \forall z\, ((x \asymp y \wedge y \asymp z) \Rightarrow x \asymp z) \\ \text{for all } f \text{ of arity } n \text{ in } \mathcal{F}, \text{ for all } i \text{ in } \{1, \ldots, n\} \setminus \mathit{prarity}(f, \asymp), \\ \forall \overline{u}, x_i, \overline{v}, y_i \\ \quad x_i \asymp y_i \Rightarrow f(\overline{u}, x_i, \overline{v}) \asymp f(\overline{u}, y_i, \overline{v}) \\ \text{for all } P \text{ of arity } n \text{ in } \mathcal{P}, \text{ for all } i \text{ in } \{1, \ldots, n\} \setminus \mathit{prarity}(P, \asymp), \\ \forall \overline{u}, x_i, \overline{v}, y_i \\ \quad x_i \asymp y_i \Rightarrow P(\overline{u}, x_i, \overline{v}) \Leftrightarrow P(\overline{u}, y_i, \overline{v}) \end{cases}$$

There can exist in a given context several equalities with different sets of protective symbol. These equalities will however have to be protective symbols with respect to each another, in order to avoid one equality to be spoiled by another one, as described by the following lemma:

**Lemma 2.4** Let $\asymp_1$ and $\asymp_2$ be two equality predicates such that $\mathit{prarity}(\asymp_1, \asymp_2) = \{1\}$ (i.e. $\asymp_1$ is not protecting its second argument from $\asymp_2$).

We have that for any $a$ and $b$, $a \asymp_2 b \Rightarrow a \asymp_1 b$.

**Proof:** Let $a$ and $b$ be arbitrary terms and assume that $a \asymp_2 b$. By reflexivity of $\asymp_1$, we have $a \asymp_1 a$. Using $\asymp_1$ as $P$ in the definition of $Th_{\asymp_2}$, we have $a \asymp_1 b$. □

**Remark 2.5** A direct consequence of the previous lemma is that in the classical setting – i.e. without protective symbols – there cannot be two different equality symbols.

We assume in this work that all the theories considered contain:

- an identity predicate, denoted by :=:, which has no protective symbols and satisfying the decomposition axioms:

$$\begin{cases} \forall \overline{u}, \overline{v} \\ \qquad f(\overline{u}) :=: f(\overline{v}) \Rightarrow \overline{u} :=: \overline{v} \\ \text{for all } f \text{ in } \mathcal{F} \end{cases}$$

- a equality predicate, denoted by ≈, which has protective symbols, including the previous :=: (i.e. $\mathit{prarity}(:=:, \approx) = \{1, 2\}$).

### *2.4.3 Protected conditional equational systems*

We assume the reader familiar with the basic notions of term rewriting as described for example in [BN98,DJ90,Ter02]. Indeed, we allow for a more expressive notion of rewriting called *conditional class rewriting* to define an equivalence on propositions and terms:

**Definition 2.6** A *conditional term rewrite rule* is a pair of terms $l, r$ together with a proposition $c$ called the condition. Such a rule is denoted by $l \rightarrow r$ $\mathsf{if}$ $c$. As usual, the variables of $r$ as well as the free variables of $c$ must occur in $l$. A *conditional term equational axiom* is a pair of terms $l, r$ together with a proposition $c$, it is denoted by $l \approx r$ $\mathsf{if}$ $c$. A *conditional proposition equational axiom* is a pair of atomic propositions $l, r$ together with a proposition $c$, it is denoted by $l \approx r$ $\mathsf{if}$ $c$. A *conditional proposition rewrite rule* is a 3-tuple of propositions $c, l, r$ where $l$ is atomic and $c, r$ are arbitrary, it is also denoted by $l \rightarrow r$ $\mathsf{if}$ $c$. As usual, the free variables of $c$ and $r$ must occur in $l$. In each case the condition can be omitted.

Term rewrite rules like

$$x + 0 \rightarrow x \text{ and } l_1.x.l_2.y.l_3 \rightarrow l_1.y.l_2.x.l_3 \ \mathsf{if}\ y > x$$

are respectively part of the theory of groups and describing (the main part of) a sorting algorithm for lists. Examples of term equational axioms are

$$x + y \approx y + x \text{ and } x.(y.z) \approx (x.y).z$$

that are respectively part of the theory of Abelian groups and the theory of lists. An instance of a proposition rewrite rule is

$$x \in \mathcal{P}(y) \rightarrow \forall z\ (z \in x \Rightarrow z \in y)$$

that describes powerset in set theory. In this case, $x$ and $y$ are free in the rule, but $z$ is not. Finally, an instance of a proposition equational axiom is the commutativity of an equality symbol $\doteq$

$$(x \doteq y) \approx (y \doteq x)$$

**Definition 2.7** A *conditional class rewrite system* is a pair, denoted by $\mathcal{RE}$, consisting of:

- $\mathcal{R}$: a set of conditional rewrite rules on propositions or terms,
- $\mathcal{E}$: a set of conditional equational axioms on propositions or terms.

**Definition 2.8** To a conditional class rewrite system $\mathcal{RE}$, we associate the theory denoted by $T_{\mathcal{RE}}$ such that for each conditional rewrite rule $l \rightarrow r$ $\mathsf{if}$ $c$ or equational axiom $l \approx r$ $\mathsf{if}$ $c$, $T_{\mathcal{RE}}$ is the universal closure of the associated propositions, i.e.:

- $\forall \overline{x}(c \Rightarrow (l \Leftrightarrow r))$ when $l$ and $r$ are propositions.
- $\forall \overline{x}(c \Rightarrow (l \approx r))$ when $l$ and $r$ are terms.

**Definition 2.9** We define $\mathit{Prot}(t, \omega, \approx)$ to be true iff the position $\omega$ is in the protective scope of a protective symbol for $\approx$, i.e. a position prefix of $\omega$ is in a protective position of a protective symbol for $\approx$ in $t$.

**Definition 2.10** Given a conditional class rewrite system $\mathcal{RE}$, a set of propositions $\Gamma$, propositions $P, P', Q, Q'$ and $c$, a non-protected occurrence $\omega$ in $P$ (i.e. such that $\mathit{Prot}(P, \omega)$ is false), a substitution $\sigma$ such that $T_{\mathcal{RE}}, \Gamma \vdash \sigma(c)$, we define the following relations:

(1) $P \longleftrightarrow^{\Gamma}_{\mathcal{E}} P'$, if $P' = P[\sigma(r)]_\omega$, for some equational axiom $l \approx r$ $\mathsf{if}$ $c$ or $r \approx l$ $\mathsf{if}$ $c$ in $\mathcal{E}$ such that $\sigma(l) = P_{|\omega}$. We say that $P$ *$\mathcal{E}$-equates* $P'$ in context $\Gamma$.

(2) $P =^{\Gamma}_{\mathcal{E}} P'$ is the equivalence generated by $\mathcal{E}$ for a context $\Gamma$, i.e. the reflexive and transitive closure of $\longleftrightarrow^{\Gamma}_{\mathcal{E}}$.

(3) $P \longrightarrow^{\Gamma}_{\mathcal{R}} P'$, if $P' = P[\sigma(r)]_\omega$, for some rewrite rule $l \rightarrow r$ $\mathsf{if}$ $c$ in $\mathcal{R}$ such that $\sigma(l) = P_{|\omega}$. We say that $P$ *$\mathcal{R}$-rewrites* to $P'$ in context $\Gamma$.

(4) $Q \longrightarrow^{\Gamma}_{\mathcal{RE}} Q'$, if $Q =^{\Gamma}_{\mathcal{E}} P[\sigma(l)]_\omega$, $Q' =^{\Gamma}_{\mathcal{E}} P[\sigma(r)]_\omega$, for some rule $l \rightarrow r$ $\mathsf{if}$ $c$ in $\mathcal{R}$. We say that $Q$ *$\mathcal{RE}$-rewrites* to $Q'$ in context $\Gamma$

(5) $P \longrightarrow^{\Gamma}_{\mathcal{R},\mathcal{E}} P'$, if $P' = P[\sigma(r)]_\omega$, for some rule $l \rightarrow r$ $\mathsf{if}$ $c \in \mathcal{R}$ such that $\sigma(l) =^{\Gamma}_{\mathcal{E}} P_{|\omega}$. We say that $P$ *$\mathcal{R},\mathcal{E}$-rewrites* to $P'$ in context $\Gamma$. This is the classical equivalence adapted from Peterson and Stickel [PS81] where it was only defined on terms.

(6) $=^{\Gamma}_{\mathcal{RE}}$ is the equivalence generated by $\mathcal{R} \cup \mathcal{E}$ for a context $\Gamma$, thus the symmetric, reflexive and transitive closure of $\longleftrightarrow^{\Gamma}_{\mathcal{E}} \cup \longrightarrow^{\Gamma}_{\mathcal{R}}$.

When the context $\Gamma$ is either empty or clear from the context we omit it. The reflexive transitive closure of a relation $\longrightarrow$ is written as usual $\xrightarrow{*}$.

The equivalence $=^{\Gamma}_{\mathcal{RE}}$ is not decidable in general, in particular since one needs to decide of the validity of the condition to apply a conditional rule. Hovewer several restrictions of the general definition of conditional rewriting like non-equational un-conditional rewriting make it decidable. More interesting restrictions like decreasing rewrite systems are for example described in [DO90].

### *2.5 Deduction modulo using conditional equational systems*

Later we will use deduction modulo equivalences represented by protected conditional equational systems. We show now that for every conditional class rewrite system $\mathcal{RE}$, there exists a set of axioms $\mathcal{T}$ such that $\mathcal{T}$ and $\mathcal{RE}$ are compatible up to the theory of equality.

This result ensures that we indeed are in a convenient representation of logic and not in a more (or less) powerful setting.

**Lemma 2.5** For any conditional class rewrite system $\mathcal{RE}$, the theory $\mathcal{T}_{\mathcal{RE}}$ in definition 2.8 is compatible with $\mathcal{RE}$ up to $Th_{\approx}$.

**Proof:** The complete proof of this natural but important result is given in appendix B. We here give the sketch of the proof.

We shall prove that:

(A) for all propositions $P$ and $Q$ and all contexts $\Gamma$, we have $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$.

We proceed by induction on the length $n$ of the derivation $P \overset{n}{\longleftrightarrow}{}^{Th_{\approx},\Gamma}_{\mathcal{R}\cup\mathcal{E}} Q$.

(B) For every proposition $P$ in $\mathcal{T}_{\mathcal{RE}}$, we have $Th_{\approx} \vdash_{\sim} P$.

In each case, we apply the corresponding axiom or rule in $\mathcal{R}$ $\mathcal{E}$.

□

**Example 2.3** Let $\mathcal{RE} = \{x + 0 \rightarrow x, x + s(y) \rightarrow s(x + y)\}$ and the corresponding theory $\mathcal{T}_{\mathcal{RE}} = \{\forall x\,(x+0 \approx x), \forall x\,\forall y\,(x+s(y) \rightarrow s(x+y))\}$. $\mathcal{RE}$ and $\mathcal{T}_{\mathcal{RE}}$ are compatible up to $Th_{\approx}$.

**Remark 2.6** *Of course it is tempting to manage things to get class (conditional) rewrite systems $\mathcal{RE}$ that are confluent and Noetherian. This eases in particular equality decision modulo $\mathcal{RE}$. But we should notice that at this stage of the framework, there is a complete freedom and therefore no restriction about confluence or termination of $\mathcal{RE}$ is currently enforced.*

## 3 $HOL_{\lambda\sigma}$, a first-order expression of higher-order logic

Since our first goal is to formalize induction in the framework of deduction modulo, we have to express later in this paper the second-order axiom of induction. To do this in a first-order presentation of higher-order logic, we use the framework introduced in [DHK01] to which we refer the reader for all the details. We just recall now the basic notations and results that will be useful in the next sections.

We denote by HOL-$\lambda$ the usual presentation of higher-order logic [Chu40,And86]. Terms are those of a simply typed $\lambda$-calculus with two base types $\iota$ and $o$ and the following constants $\dot{\Rightarrow}$, $\dot{\wedge}$ and $\dot{\vee}$, of type $o \to o \to o$, $\dot{\neg}$ of type $o \to o$, $\dot{\bot}$ of type $o$, $\dot{\forall}_T$ and $\dot{\exists}_T$ of type $(T \to o) \to o$. Notice that we use a notation with a dot for the constants to distinguish them from the connectors and quantifiers of first-order logic. Propositions are terms of type $o$. We have no room here to recall the deduction rules of the logic, but they are made precise in e.g. [DHK01].

The first-order formulation of higher-order logic we are using is based on de Bruijn indices and explicit substitutions. This notation is also a first-order language with a binary function symbol $\alpha$, a unary function symbol $\lambda$ and individual symbols $1, 2, 3 \ldots$

Simple sorts are not sufficient anymore with de Bruijn indices. Indeed, we need to give a sort not only to terms like $(\lambda_A\ 1)$ (that gets the sort $A \to A$), but also to terms of the form 1. Thus, as detailed in [DHK00], we have to consider sorts of the form $\Gamma \vdash T$ where $T$ is a simple type and $\Gamma$ a context, i.e. a list of simple types: intuitively, $\Gamma$ gives the types of the free variables of a term which, in that context, has type $T$.

With de Bruijn indices conversion axioms use an external definition for substitution. Moreover this substitution is not well-defined on open terms of this first-order language. This is solved by considering an extension of this calculus: the *calculus of explicit substitutions* [ACCL91] also called $\lambda\sigma$-calculus. This calculus also introduces sorts of the form $\Gamma \vdash \Delta$ for substitutions that are lists of terms and symbols to build such substitutions $id$, $.$, $\uparrow$ and $\circ$. Then a new term constructor is introduced $\_[\_]$ that allows to apply an explicit substitution to a term. The rewrite rules describing the evaluation of the $\lambda\sigma$-calculus are given in Figure 2.

In this framework, $HOL_{\lambda\sigma}$ is the sequent calculus modulo defined in the following way.
(1) The syntax of the language consists of sorts of the form $\Gamma \vdash T$ and $\Gamma \vdash \Delta$ where $\Gamma$ and $\Delta$ are sequences of simple types and $T$ is a simple type. It contains

| | | |
|---|---|---|
| **Beta** | $(\lambda a)b$ | $\rightarrow a[b.id]$ |
| **Eta** | $\lambda(a\ \mathtt{1})$ | $\rightarrow b$ |
| | | $\mathsf{if}\ a =_\sigma b[\uparrow]$ |
| $\sigma$-reduction: | | |
| **App** | $(a\ b)[s]$ | $\rightarrow (a[s]\ b[s])$ |
| **VarCons** | $\mathtt{1}[a.s]$ | $\rightarrow a$ |
| **Id** | $a[id]$ | $\rightarrow a$ |
| **Abs** | $(\lambda a)[s]$ | $\rightarrow \lambda(a[\mathtt{1}.(s \circ \uparrow)])$ |
| **Clos** | $(a[s])[t]$ | $\rightarrow a[s \circ t]$ |
| | | |
| **IdL** | $id \circ s$ | $\rightarrow s$ |
| **ShiftCons** | $\uparrow \circ\ (a.s)$ | $\rightarrow s$ |
| **AssEnv** | $(s_1 \circ s_2) \circ s_3$ | $\rightarrow s_1 \circ (s_2 \circ s_3)$ |
| **MapEnv** | $(a.s) \circ t$ | $\rightarrow a[t].(s \circ t)$ |
| **IdR** | $s \circ id$ | $\rightarrow s$ |
| **VarShift** | $\mathtt{1}.\uparrow$ | $\rightarrow id$ |
| **Scons** | $\mathtt{1}[s].(\uparrow \circ\ s)$ | $\rightarrow s$ |

Fig. 2. The rewrite rules of $\lambda\sigma$-calculus

the function symbols given in Figure 3 and a single unary predicate symbol:

$$\varepsilon \text{ of rank } (\vdash o)$$

(2) The equivalence is defined by the class rewrite system denoted by $\lambda\sigma\mathcal{L}$ and consisting of the rewrite rules of $\lambda\sigma$-calculus together with the logical rules $\mathcal{L}$ given in Figure 4 and explained in detail in [DHK01]. As shown in [DHK01], the system $\lambda\sigma\mathcal{L}$ is weakly terminating and confluent on terms containing no substitution variables and the theory $HOL_{\lambda\sigma}$ is consistent and the cut rule is redundant in it.

As in [DHK00], we use a translation from $\lambda$-calculus to $\lambda\sigma$-calculus called *pre-cooking*. Bound variables are translated by the appropriate indices and free variables are translated variables of the first-order theory, relocated by an appropriate $[\uparrow^n]$ operator according to the context in which they occur. To each variable $x$ of type $T$, we associate the sort $\vdash T$ in $\lambda\sigma$-calculus. The pre-

$$\begin{array}{ll}
1^{\Gamma}_{A} & \text{of sort } \; A.\Gamma \vdash A \\
\alpha^{\Gamma}_{A\rightarrow B,A} & \text{of rank } (\Gamma \vdash A \rightarrow B, \Gamma \vdash A)\Gamma \vdash B \\
\lambda^{\Gamma}_{A,B} & \text{of rank } (A.\Gamma \vdash B)\Gamma \vdash A \rightarrow B \\
[\;]^{\Gamma,\Gamma'}_{A} & \text{of rank } (\Gamma' \vdash A, \Gamma \vdash \Gamma')\Gamma \vdash A \\
id^{\Gamma} & \text{of sort } \; \Gamma \vdash \Gamma \\
\uparrow^{\Gamma}_{A} & \text{of sort } \; A.\Gamma \vdash \Gamma \\
\cdot^{\Gamma,\Gamma'}_{A} & \text{of rank } (\Gamma \vdash A, \Gamma \vdash \Gamma')\Gamma \vdash A.\Gamma' \\
\circ^{\Gamma,\Gamma',\Gamma''} & \text{of rank } (\Gamma \vdash \Gamma'', \Gamma'' \vdash \Gamma')\Gamma \vdash \Gamma' \\
\dot{\Rightarrow} & \text{of sort } \; \vdash o \rightarrow o \rightarrow o \\
\dot{\wedge} & \text{of sort } \; \vdash o \rightarrow o \rightarrow o \\
\dot{\vee} & \text{of sort } \; \vdash o \rightarrow o \rightarrow o \\
\dot{\neg} & \text{of sort } \; \vdash o \rightarrow o \\
\dot{\bot} & \text{of sort } \; \vdash o \\
\dot{\forall}_{A} & \text{of sort } \; \vdash (A \rightarrow o) \rightarrow o \\
\dot{\exists}_{A} & \text{of sort } \; \vdash (A \rightarrow o) \rightarrow o
\end{array}$$

Fig. 3. $HOL_{\lambda\sigma}$ syntax

$$\begin{array}{rcl}
\varepsilon(\dot{\Rightarrow}\; x\; y) & \rightarrow & \varepsilon(x) \Rightarrow \varepsilon(y) \\
\varepsilon(\dot{\wedge}\; x\; y) & \rightarrow & \varepsilon(x) \wedge \varepsilon(y) \\
\varepsilon(\dot{\vee}\; x\; y) & \rightarrow & \varepsilon(x) \vee \varepsilon(y) \\
\varepsilon(\dot{\neg}\; x) & \rightarrow & \neg\varepsilon(x) \\
\varepsilon(\dot{\bot}) & \rightarrow & \bot \\
\varepsilon(\dot{\forall}_{T}\; x) & \rightarrow & \forall y\; \varepsilon(x\; y) \\
\varepsilon(\dot{\exists}_{T}\; x) & \rightarrow & \exists y\; \varepsilon(x\; y)
\end{array}$$

Fig. 4. The $\mathcal{L}$-rewrite rules

cooking of a $\lambda$-term $a$ is the $\lambda\sigma$-term defined by $a_F = F(a, [\;])$ where $F(a, l)$ is defined using the list of variables $l$ ([ ] being the empty list) by:

- $F((\lambda x.a), l) = \lambda(F(a, x.l))$,
- $F((a\; b), l) = F(a, l)F(b, l)$,

- $F(x,l) = 1[\uparrow^{k-1}]$, if $x$ is the $k$-th variable of $l$,
- $F(x,l) = x[\uparrow^{n}]$ where $n$ is the length of $l$ if $x$ is a variable not occurring in $l$ or a constant.

In this framework it can be proved that $HOL_{\lambda\sigma}$ is intentionally equivalent to the usual presentation of higher-order logic $HOL_{\lambda}$ [DHK01] and therefore from now on we use it as a first-order presentation of higher-order logic. The formulas in $HOL_{\lambda\sigma}$ can even be denoted by the corresponding higher-order formulas in spite of a quite different point of view, as we now see on the simple proposition:

$$\forall P\, \exists \tau\, \forall x\, (x \in \tau {\Rightarrow} P(x)) \tag{1}$$

In $HOL_{\lambda}$ it has the form:

$$\dot{\forall} \lambda P\, \dot{\exists} \lambda \tau\, \dot{\forall} \lambda x\, (x \in \tau \,\dot{\Rightarrow}\, P(x))$$

By applying pre-cooking we obtain:

$$\begin{aligned}&\varepsilon(\alpha(\dot{\forall}, \lambda\alpha(\dot{\exists}[\uparrow], \lambda\alpha(\dot{\forall}[\uparrow^2],\\ &\lambda\alpha(\alpha(\dot{\Rightarrow}[\uparrow^3], \alpha(\alpha(\in [\uparrow^3], \underline{1}), \underline{2})), \alpha(\underline{3}, \underline{1}))))))\end{aligned}$$

By $\lambda\sigma\mathcal{L}$-normalization, and by omitting the function symbol $\alpha$ wherever it does not enlight that the quantifications are first-order, we get:

$$\forall P\, \exists \tau\, \forall x\, (\varepsilon(x \in \tau){\Rightarrow}\varepsilon(\alpha(P, x))) \tag{2}$$

This form is quite close to the higher-order notation (1), at the main difference that it is now a first-order proposition. Indeed, if we omit to write the predicate symbol $\varepsilon$ and function symbol $\alpha$ we get the standard notation back. This amounts to consider (1) as a *notation* for the *first-order* proposition (2).

## 4 Proof by induction

We apply in this section the previous framework and results to show how Noetherian rewriting can be used, directly, to perform proof by induction. In the remainder of this paper and except if explicitly stated, we are only concerned with conditional equational goals.

Deduction modulo permits us to give a proof theoretic understanding of induction by rewriting. In the context of deduction modulo, the induction hypotheses arising from equational goals can be (dynamically) internalized into the relation. When doing this, the computational part of the deduction modulo appears to perform exactly induction by rewriting as done for instance by systems like Spike [BKR92] or RRL [KZ95].

We first recall the main notations and basic facts on induction that we use later. Then we show how this could be expressed in the first-order $HOL_{\lambda\sigma}$ sequent calculus.

### *4.1 Noetherian induction*

We first recall the standard notations and results about Noetherian induction [Wec92].

- First, subset inclusion is modeled by the proposition:

$$\tau_1 \subseteq \tau_2 \triangleq \forall x\,(x \in \tau_1 \Rightarrow x \in \tau_2).$$

- The property for an element $x$ to be *minimal* in a set $\tau$ w.r.t. a relation $R$ is defined by:

$$Minimal(R, \tau, x) \triangleq x \in \tau \wedge \neg\exists y\,(y \in \tau \wedge R(x, y)).$$

- A relation is *Noetherian* or *well-founded* if every non-empty set has a minimal element:

$$\mathit{Noeth}(R, \tau) \triangleq \forall \tau'((\tau' \subseteq \tau \wedge \exists x\,(x \in \tau')) \Rightarrow \exists y\, Minimal(R, \tau', y)).$$

- A proposition $P$ is *inductive* relative to the relation $R$ in the set $\tau$ when:

$$\mathit{Ind}(P, R, \tau) \triangleq \forall x\,((x \in \tau \wedge \forall y\,((y \in \tau \wedge R(x, y)) \Rightarrow P(y))) \Rightarrow P(x)).$$

- To denote a proposition true on a set $\tau$, we let:

$$\mathit{True}(P, \tau) \triangleq \forall x\,(x \in \tau \Rightarrow P(x)).$$

- The *Noetherian induction principle* can now be defined as:

$$\mathit{NoethInd}(P, R, \tau) \triangleq \mathit{Ind}(P, R, \tau) \Rightarrow \mathit{True}(P, \tau).$$

As shown for example in [Hue86,Wec92], noetherianity and the induction principle are equivalent:

**Lemma 4.1**

$$\forall\tau\,\forall P\,\exists\tau'\,(\forall x\,(x \in \tau' \Leftrightarrow (x \in \tau \wedge P(x))))$$
$$\vdash$$
$$\forall R\,\forall\tau\,(\mathit{Noeth}(R, \tau) \Leftrightarrow \forall P\,(\mathit{NoethInd}(P, R, \tau))).$$

**Proof:** This could be proved using Coq, in classical logic (Requires Classical).
□

This allows us to use the proposition $\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau)))$ to make available the induction principle while generating explicitly the noetherianity proof obligations.

### *4.2 Inductive consequence and inductive property*

A proposition $Q$ valid in all the Herbrand models of the theory $Th_u$ is called an *inductive property* of $Th_u$, which is denoted $Th_u \models_{Ind} Q$. When the theory $Th_u$ is a set of Horn clauses, there exists a unique smallest Herbrand model (with respect to inclusion) which could be used as a canonical representative. Even simpler is the case of equational axioms where the smallest Herbrand model is the initial algebra $\mathcal{T}(\mathcal{F})/Th_u$.

A proposition $Q$ is an *inductive consequence* of $Th_u$ when, assuming the noetherianity of a relation $R$, $Q$ could be derived from the induction principle and the user theory axioms, using the deduction rules of higher-order logic:

$$\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u \vdash_{HOL} Q.$$

We strongly insist on the fact that contrarily to some of the terminology used in the literature, an inductive *consequence* is a proposition that could be formaly derived in higher-order logic (e.g. using $\vdash_{HOL}$) with the induction principle and the user theory as context.

The inductive consequences of $Th_u$ are inductive properties of $Th_u$.

To prove a proposition $Q$ by induction amounts to:

(A) find a well-founded relation $R$ on a set $\tau$.
(B) such that

$$\begin{aligned}&\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))),\\ &\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, \mathit{NoethInd}(P,R,\tau), Th_u \vdash_{HOL} Q\end{aligned}$$

where $\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, \mathit{NoethInd}(P,R,\tau)$ denotes an instance of the induction principle using $R$.

Both steps are known to be non-trivial [Bun01]. In this paper we concentrate on the case where the well-founded relation $R$ contains a reduction ordering on terms [Der87,KK99]. This is still a very powerful situation that strictly contains structural induction, for instance standard Peano induction.

*From now on we work in $HOL_{\lambda\sigma}$ and therefore use the notation $\vdash^{\lambda\sigma}$ for deduction in $HOL_{\lambda\sigma}$*

### *4.3 Using the induction hypothesis*

We want to prove that a property $P$ is an inductive consequence of a user theory $Th_u$.

Using the induction principle gives rise to a new hypothesis called the induction hypothesis. We intend to give it the best form to be useful, ultimately internalizing it whenever possible.

**Notation 4.1** *We use capital $X$ for a free variable as freed by the rules $\forall$-r and $\exists$-l on quantifiers (see Figure 1).*

**Notation 4.2** *When the user theory is an equational theory, we also need to have $Th_{\approx}$ and $Th_{:=:}$ in the context. Therefore, we denote*

$$\mathcal{T}h \triangleq \forall R \forall \tau \, (Noeth(R,\tau) \Rightarrow \forall P \, (NoethInd(P,R,\tau))), Th_{\approx}, Th_{:=:}$$

#### *4.3.1 Single-goal case*

We want to prove that a property $\forall \overline{x} \, (\overline{x} \in \overline{\tau} \Rightarrow (C(\overline{x}) \Rightarrow Q(\overline{x})))$ is an inductive consequence of a user theory $Th_u$. To keep the notations simple, we deal with the case of two variables and one condition. Even if this is of no use in the first steps, we intend to focus on the case where $Q$ is an equality.

**Lemma 4.2** Given a user theory $Th_u$ and a Noetherian ordering $\prec$ on a set $\tau_x$, the equivalences described in Figure 5 hold.

**Proof:** By a simple application of the appropriate deduction rules of the sequent calculus modulo. This is detailed in appendix C. □

The main interest of this Lemma is to make clear the last statement (5) that will be used later, making use of the fact that $Q$ is assumed to be an equality. Indeed in that statement, the proposition

$$\forall \underline{x} \forall y \, ((\underline{x} \in \tau_x \wedge \underline{x} \prec X \wedge y \in \tau_y \wedge C(\underline{x}, y)) \Rightarrow Q(\underline{x}, y)) \qquad (6)$$

is what is called usualy the "induction hypothesis".

$$\begin{array}{ll} & \forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u \\ \vdash^{\lambda\sigma} & \forall x\, \forall y\, (x \in \tau_x \Rightarrow (y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))) \qquad (3) \\ \mathit{IFF} & \\ & \forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x \\ \vdash^{\lambda\sigma} & \forall y\, (y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \qquad (4) \\ \mathit{IFF} & \\ & \forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x, \\ & \forall \underline{x}\, \forall y\, ((\underline{x} \in \tau_x \wedge \underline{x} \prec X \wedge y \in \tau_y \wedge C(\underline{x},y)) \Rightarrow Q(\underline{x},y)) \\ \vdash^{\lambda\sigma} & \forall y\, (y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \qquad (5) \end{array}$$

Fig. 5. Towards using the induction hypothesis

### *4.3.2 Multiple-goal case*

When we have two properties on the same set (or a set and a subset of it), we are able to generate an induction hypothesis for each property. This possibility is very powerful, and is indeed used by Spike.

The hypothesis of the properties being on a set and a subset of it permits to obtain an equivalence between

$$\forall x\, (x \in \tau_1 \Rightarrow (C_1(x) \Rightarrow Q_1(x))) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))$$

and

$$\forall x\, (x \in \tau_1 \Rightarrow ((C_1(x) \Rightarrow Q_1(x)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))).$$

**Lemma 4.3** Given a user theory $Th_u$ and a Noetherian ordering $\prec$ on a set $\tau_1$, and given a subset $\tau_2$ of $\tau_1$, the relations described in Figure 6 hold.

**Proof:** Also by a simple application of the appropriate deduction rules of the sequent calculus modulo. This is detailed in appendix D. □

As in the single goal case, this result allows us to provide two induction hypothesis apppearing in statement (9). When $Q$ consists of an equality, the next section will show how this could be internalized as a conditional rewrite rule.

### *4.4 Internalization of induction hypotheses*

When the properties $Q$ to be proved are equational theorems of the form $t_1 \approx t_2$, the induction hypotheses shown in the previous section have exactly the form of canonical theories from Definition 2.8, and therefore we are able

$$
\begin{array}{ll}
 & \forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \\
 & \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1) \\
\vdash^{\lambda\sigma} & \forall x\, (x \in \tau_1 \Rightarrow (C_1(x) \Rightarrow Q_1(x))) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))) \quad (7) \\
IFF & \\
 & \forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_1, \\
 & \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1) \\
\vdash^{\lambda\sigma} & (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))) \quad (8) \\
IFF & \\
 & \forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_1, \\
 & \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), \\
 & \forall \underline{x}\, ((\underline{x} \in \tau_1 \wedge \underline{x} \prec X \wedge C_1(\underline{x})) \Rightarrow Q_1(\underline{x})), \\
 & \forall \underline{y}\, ((\underline{y} \in \tau_2 \wedge \underline{y} \prec X \wedge C_2(\underline{y})) \Rightarrow Q_2(\underline{y})) \\
\vdash^{\lambda\sigma} & (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))) \quad (9)
\end{array}
$$

Fig. 6. More induction hypotheses

to internalize them using deduction modulo. Lemma 4.2 and Lemma 4.3 give us two forms for the induction hypotheses:

- The first form is the most usual one, coming from the goal we apply induction on, as given by (6). We will note:

  $$\mathcal{RE}_{\mathit{Ind}(t_1 \approx t_2, C, x)}(X) =$$
  $$t_1(\underline{x}, y) \approx t_2(\underline{x}, y) \text{ if } \underline{x} \in \tau_x \wedge \underline{x} \prec X \wedge y \in \tau_y \wedge C(\underline{x}, y)$$

- The second form is the one arising by lemma 4.3 from the second goal. We will note:

  $$\mathcal{RE}_{\mathit{Ind}(t_3 \approx t_4, t_1 \approx t_2, C_2, z, x)}(X) =$$
  $$t_3(\underline{z}, w) \approx t_4(\underline{z}, w) \text{ if } \underline{z} \in \tau_z \wedge \underline{z} \prec X \wedge w \in \tau_w \wedge C_2(\underline{z}, w)$$

**Example 4.1** For instance, if we have a goal:

$$\forall x\, (x \in \mathit{Nat} \Rightarrow 0 + x \approx x) \wedge \forall y \forall z\, ((y \in \mathit{Nat} \wedge z \in \mathit{Nat}) \Rightarrow y + z \approx z + y)$$

by induction on $x$, we get three induction hypotheses:

$$\mathcal{RE}_{\mathit{Ind}(0+x \approx x, x)}(X) = 0 + \underline{x} \approx \underline{x} \text{ if } \underline{x} \in \mathit{Nat} \wedge \underline{x} \prec X$$
$$\mathcal{RE}_{\mathit{Ind}(y+z \approx z+y, 0+x \approx x, y, x)}(X) = \underline{y} + z \approx z + \underline{y} \text{ if } \underline{y} \in \mathit{Nat} \wedge z \in \mathit{Nat} \wedge \underline{y} \prec X$$
$$\mathcal{RE}_{\mathit{Ind}(y+z \approx z+y, 0+x \approx x, z, x)}(X) = y + \underline{z} \approx \underline{z} + y \text{ if } y \in \mathit{Nat} \wedge \underline{z} \in \mathit{Nat} \wedge \underline{z} \prec X$$

### *4.5 Using the membership hypothesis*

The next step consists in using the membership hypothesis $X \in \tau$. We shall make use of an induction scheme

$$\forall x \, (x \in \tau \Leftrightarrow \vee_i \exists \overline{x_i} \, (\overline{x_i} \in \overline{\tau_{x_i}} \wedge x :=: t_i(\overline{x_i})))$$

This induction scheme has to cover all the possible cases and to allow us to simplify the generated subgoals.

**Lemma 4.4** Applying an induction scheme to

$$\mathcal{T}h, Th_u, X \in \tau \vdash_{t_1(\underline{x}) \approx t_2(\underline{x}) \textsf{ if } \underline{x} \in \tau \wedge \underline{x} \prec X} Q(X)$$

we get subgoals of the form:

$$\mathcal{T}h, Th_u, X \in \tau, \overline{X_i} \in \overline{\tau_{x_i}}, \vdash_{t_1(\underline{x}) \approx t_2(\underline{x}) \textsf{ if } \underline{x} \in \tau \wedge \underline{x} \prec t_i(\overline{X_i})} Q(t_i(\overline{X_i}))$$

Using the fact that if the relation $=_{\mathcal{RE}_{Ind(Q)}(X)}^{\mathcal{T}h, Th_u, X \in \tau_x}$ can be oriented into a terminating term rewrite system, then only irreducible instances of $X$ are useful and should therefore be generated. This fact has been used to design various techniques based on:

- constructor discipline,
- test sets [BKR95],
- covering sets [ZKK88],
- tree automata [BJ99].

### *4.6 Simplifying the goal*

We can choose the best representation for the proposition $Q(X)$ modulo the conditional class rewrite system $\mathcal{RE}_{Ind(Q)}(X)$. A natural assumption will be to assume that the class rewrite system is confluent and Noetherian, in which case the canonical representative of the proposition $Q(X)$ by $\mathcal{RE}_{Ind(Q)}(X)$ will be its normal form. Indeed to use the equational form of the induction hypothesis $\mathcal{RE}_{Ind(Q)}(X)$ one needs to check the condition $\underline{x} \prec X$. Theorem 4.1 will show that, for the right choice of the induction ordering, as soon as the goal has been simplified, the condition is indeed true.

### *4.7 The induction ordering*

A key point in any proof by induction is to provide an appropriate Noetherian ordering $\prec$ to base the induction on. In our context, if the conditional class rewrite system $\mathcal{RE}_{Ind(Q)}(X)$ is Noetherian, it is natural to choose an ordering containing it to drive the induction. Other possibilities are up to the (semi-)automated system designer's choice, but we will stick to this, in particular because to the nice properties of the resulting framework.

Of course, the Noetherianity of $\prec$ is essential in the proofs. For example it allows us to close the following branch in the proof of Lemma 4.2:

$$\forall R \forall \tau\, (Noeth(R, \tau) \Rightarrow NoethInd(R, \tau)), Th_u \vdash^{\lambda\sigma} Noeth(\prec, \tau), P$$

We should also remark that $\prec$ (and its semantics) will usually not allow rewriting its arguments and therefore a protective symbol will appear in its definition.

#### *4.7.1 An ordering on equations*

In order to instantiate appropriately the Noetherian ordering $\prec$, we let it take into account more than simply the terms but also the structure of the goal to be proved. So the remainder of this work is based on the ordering used in Spike [Bou94] to compare equations. We give here two equivalent definitions for this ordering. Two more definitions and the proofs of equivalence of all the definitions are given in appendix E.

**Definition 4.1** Let $<$ be a noetherian ordering on terms.

$$\mathcal{C}(t \approx t') = \begin{cases} (\{t\}, \{t'\}) & \text{if } t' < t \\ (\{t'\}, \{t\}) & \text{if } t < t' \\ (\{t, t'\}, \{\}) & \text{otherwise} \end{cases}$$

and we define $>_e$ by

$$a \approx b >_e c \approx d \text{ if } \mathcal{C}(a \approx b) \gg_{lex} \mathcal{C}(c \approx d)$$

where $\gg_{lex}$ is the lexicographic extension of the set extension of $>$.

An alternative and equivalent definition is:

**Definition 4.2**

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow \{t_1, t_2\} \gg \{t_3, t_4\} \vee t_1 \# t_2 \wedge \bigvee \begin{cases} t_1 = t_3 \wedge t_1 > t_4 \\ t_2 = t_3 \wedge t_2 > t_4 \\ t_1 = t_4 \wedge t_1 > t_3 \\ t_2 = t_4 \wedge t_2 > t_3 \end{cases}$$

We state useful properties of the ordering $>_e$ that we will use throughout the proof. For any terms $t_1$, $t_2$ and $t$ and equation $E$:

**Lemma 4.5** $>_e$ respects the semantics of commutativity of $\approx$, therefore, for any terms $t_1$ and $t_2$ and any equation $E$:

$$t_1 \approx t_2 >_e E \Rightarrow t_2 \approx t_1 >_e E$$
$$E >_e t_1 \approx t_2 \Rightarrow E >_e t_2 \approx t_1$$

**Proof:** It is easy to check that $C(t_1 \approx t_2) = C(t_2 \approx t_1)$. □

**Lemma 4.6** Quasi-stability in equational context:

$$t_1 > t_2 \Rightarrow t_1 \approx t >_e t_2 \approx t \;\text{ if } t \neq t_1$$

**Proof:** By cases in the comparison between $t_1$ and $t$ on the one hand and of $t_2$ and $t$ on the other hand. □

**Lemma 4.7** The order of reduction of the two terms of an equation does not commute.

$$t_1 \approx t_2 >_e t_1' \approx t_2 >_e t_1' \approx t_2' \not\Rightarrow t_1 \approx t_2 >_e t_1 \approx t_2' >_e t_1' \approx t_2'$$

**Proof:** This is easily checked from the definition □

**Lemma 4.8** $>_e$ is stable by substitution if $>$ is.

**Proof:** Easy check that the complexity is preserved by substitution. □

### *4.7.2 Avoiding to check the conditions*

The next result finishes to explain the behavior of induction by rewriting in our proof theoretical setting. It asserts that when using a Noetherian ordering compatible with the rewrite relation, the conditions in the application of the equational axioms and rewrite rules are always satisfied.

**Lemma 4.9** Consider a goal $\alpha_1 \approx \alpha_2$, a conditional class rewrite system $\mathcal{R}$ $\mathcal{E}$ and an equation $t_1 \approx t_2$ or a rule $t_1 \rightarrow t_2$.

We have $\sigma(t_1) \approx \sigma(t_2) <_e \alpha_1 \approx \alpha_2$ when:

(A) $\alpha_1 \equiv \alpha_1[\sigma(t_1)]_\omega$ for a position $\omega$ in $\alpha_1$ such that $\omega \neq \varepsilon$.
(B) $\alpha_1 \approx \alpha_2 \rightarrow^+_{\mathcal{R}\cup\mathcal{E}} \alpha'_1 \approx \alpha_2$, with $\alpha_1 \approx \alpha_2 >_e \alpha'_1 \approx \alpha_2$, and
    (a) $\alpha'_1 \equiv \alpha'_1[\sigma(t_1)]_\omega$ at a position $\omega$ in $\alpha'_1$.
    (b) $\alpha_2 \equiv \alpha_2[\sigma(t_1)]_\omega$ for a position $\omega$ in $\alpha_2$ such that $\omega \neq \varepsilon$.
    (c) $\alpha_2 \equiv \sigma(t_1)$ if $\alpha_1 \not< \alpha_2$ or $\alpha_1 > \sigma(t_2)$
(C) $\alpha_1 \approx \alpha_2 \rightarrow^+_{\mathcal{R}\cup\mathcal{E}} \alpha'_1 \approx \alpha'_2$, with $\alpha_1 \approx \alpha_2 >_e \alpha'_1 \approx \alpha'_2$, and $\alpha'_1 \equiv \alpha'_1[\sigma(t_1)]_\omega$ for any position $\omega$ in $\alpha'_1$ (or symmetrically $\alpha'_2 \equiv \alpha'_2[\sigma(t_1)]_\omega$).

**Proof:** See appendix E. □

**Theorem 4.1** *Consider a goal $\alpha_1 \approx \alpha_2$, and an induction hypothesis of the form:*

$$t_1(\underline{x}) \approx t_2(\underline{x}) \text{ if } \underline{x} \in \tau \wedge t_1(\underline{x}) \approx t_2(\underline{x}) <_e t_1(X) \approx t_2(X).$$

*The condition $t_1(\underline{x}) \approx t_2(\underline{x}) <_e t_1(X) \approx t_2(X)$ is always satisfied if the hypothesis is used:*

*(A) on a strict subterm of $\alpha_1$*
*(B) after reducing the goal into $\alpha'_1 \approx \alpha_2$*
    *(a) on the term $\alpha'_1$*
    *(b) on a strict subterm of $\alpha_2$*
    *(c) on $\alpha_2$ at head position, if $\alpha_1 \not< \alpha_2$ or $\alpha_1 > \sigma(t_2)$*
*(C) after reducing the goal into $\alpha'_1 \approx \alpha'_2$*

**Proof:** The result follows from the previous lemma. □

**Conjecture 4.1** Lemma 4.9 is no longer true with an ordering based on a reduction ordering instead of a simplification ordering.

**Conjecture 4.2** Theorem 4.1 remains true with an ordering based on a reduction ordering instead of a simplification ordering.

## 5 Proving inductive properties by rewriting

The main ideas of induction by rewriting are:

(A) to base the induction on an ordering containing the simplification ordering generated by the Noetherian term rewrite system issued from the user theory $Th_u$,

(B) to use test sets as a way to expand the membership hypothesis $X \in \tau$.

These are the main techniques first described in [KR90b,KR95] and implemented with success in the Spike system [BKR92,BKR95].

We continue to assume for simplicity that the property to be proved is an equational theorem of the form $u \approx v$.

### *5.1 Simple examples in Peano arithmetics*

Let us begin to show how the general method can be applied to prove by induction that, in Peano arithmetics:

- 0 is a left and right neutral element for addition (simple),
- addition is commutative and then associative (more elaborated).

#### *5.1.1 Setting*

First, we set up the theory for Peano arithmetics. Let:

- A sort $Nat$ with $0 \in Nat$, $s(x) \in Nat$ and $x + y \in Nat$ for any $x \in Nat$ and $y \in Nat$. This is expressed by:

$$Th^{sort}_{Nat} = \begin{cases} \forall x\,(x \in Nat \Leftrightarrow \\ \quad (x{:=:}0 \vee \\ \quad \exists y (y \in Nat \wedge x{:=:}s(y)) \vee \\ \quad \exists y\, \exists z (y \in Nat \wedge z \in Nat \\ \quad\ \wedge\, x{:=:}y + z))) \end{cases}$$

**Remark 5.1** *Note that this could also be internalized, using our ability to rewrite propositions.*

- A definition for the symbol + expressed by:

$$Th^{def+}_{Nat} = \begin{cases} \forall x\,(x \in Nat {\Rightarrow} x + 0 \approx x) \\ \forall x \forall y\,(x \in Nat \wedge y \in Nat) {\Rightarrow} x + s(y) \approx s(x + y)) \end{cases}$$

Using lemma 2.5, we can build-in $Th^{def+}_{Nat}$ as:

$$\mathcal{RE}^{def+}_{Nat} = \begin{cases} x + 0 \approx x & \text{if } x \in Nat \\ x + s(y) \approx s(x + y) & \text{if } x \in Nat \wedge y \in Nat \end{cases}$$

$\mathcal{RE}^{def+}_{Nat}$ can be oriented into a confluent and terminating rewrite system:

$$\vec{\mathcal{RE}}^{def+}_{Nat} = \begin{cases} x + 0 \rightarrow x & \text{if } x \in Nat \\ x + s(y) \rightarrow s(x + y) & \text{if } x \in Nat \wedge y \in Nat \end{cases}$$

### *5.1.2 Let us prove that* 0 *is a left neutral element for addition*

We want to prove the proposition:

$$\forall x (x \in Nat \Rightarrow 0 + x \approx x)$$

We thus start from the following sequent:

$$\mathcal{T}h, Th^{sort}_{Nat} \vdash_{\mathcal{RE}^{def+}_{Nat}} \forall x (x \in Nat \Rightarrow 0 + x \approx x)$$

By internalizing the induction hypothesis as explained in Section 4.4, we get:

$$\mathcal{T}h, Th^{sort}_{Nat}, X \in Nat \vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(0+x \approx x, x)}(X)} 0 + X \approx X$$

The proof reduces to:

$$\mathcal{T}h, Th^{sort}_{Nat} \vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(0+x \approx x, x)}(0)} 0 + 0 \approx 0 \tag{10}$$

and

$$\mathcal{T}h, Th^{sort}_{Nat}, Y \in Nat \vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(0+x \approx x, x)}(s(Y))} 0 + s(Y) \approx s(Y) \tag{11}$$

(10) is trivial using $Th_{\approx}$.

(11) is also done using $Th_{\approx}$, with:

$$\begin{aligned} 0 + s(Y) \approx s(Y) &\longrightarrow s(0 + Y) \approx s(Y) \\ &\longrightarrow s(Y) \approx s(Y) \end{aligned}$$

Notice that in the use of the induction hypothesis, thanks to Theorem 4.1, the condition

$$0 + \underline{Y} \approx \underline{Y} < 0 + s(Y) \approx s(Y)$$

is verified since the goal has been reduced before.

### *5.1.3 Let us prove commutativity of the addition*

We want to prove the proposition

$$\forall x\,(x \in Nat \Rightarrow \forall y\,(y \in Nat \Rightarrow x + y \approx y + x))$$

Internalizing the induction hypothesis on $x$ we get:

$$\mathcal{T}h, Th^{sort}_{Nat} \vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(x+y\approx y+x,x)}(0)} \forall y\,(y \in Nat \Rightarrow 0 + y \approx y + 0) \tag{12}$$

and

$$\begin{aligned} &\mathcal{T}h, Th^{sort}_{Nat}, Z \in Nat \\ &\vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(x+y\approx y+x,x)}(s(Z))} \\ &\forall y\,(y \in Nat \Rightarrow s(Z) + y \approx y + s(Z)) \end{aligned} \tag{13}$$

(12) easily reduces to the example before.

For (13), an induction on $y$ gives us:

$$\begin{aligned} &\mathcal{T}h, Th^{sort}_{Nat}, Z \in Nat \\ &\vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(x+y\approx y+x,x)}(s(Z)) \cup \mathcal{RE}_{Ind(s(Z)+y\approx s(y+Z),y)}(0)} \\ &s(Z) + 0 \approx s(0 + Z) \end{aligned} \tag{14}$$

and

$$\begin{aligned} &\mathcal{T}h, Th^{sort}_{Nat}, Z \in Nat, V \in Nat \\ &\vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(x+y\approx y+x,x)}(s(Z)) \cup \mathcal{RE}_{Ind(s(Z)+y\approx s(y+Z),y)}(S(V))} \\ &s(Z) + s(V) \approx s(s(V) + Z) \end{aligned} \tag{15}$$

(14) is simple, using the result of the previous example.

(15) is reduced as follows:

$$\begin{aligned} & s(Z) + s(V) \approx s(s(V) + Z) \\ \longrightarrow\ & s(Z) + s(V) \approx s(Z + s(V)) \\ \longrightarrow\ & s(s(Z) + V) \approx s(s(Z + V)) \end{aligned}$$

then $s(s(Z) + V) \approx s(s(Z + V))$ is proved by an easy induction on $V$ and we are done.

The difficult step here is to apply the induction hypothesis. Denoting by $s\#t$ when $s$ is incomparable with $t$, and using a LPO ordering with precedence $0 < s < +$, we have:

$$\begin{aligned} s(Z) + s(V) &\;\#\; s(s(V) + Z) \\ s(Z) + s(V) &> s(Z + s(V)) \end{aligned}$$

which allows orienting the step, since

$$s(Z) + s(V) \approx s(s(V) + Z) >_e s(Z) + s(V) \approx s(Z + s(V)).$$

We also have:

$$\begin{aligned} s(Z) + s(V) &> Z + s(V) \\ s(V) + s(Z) &> s(V) + Z \end{aligned}$$

which allows us to check that, as stated by theorem 4.1, the induction condition

$$Z + s(V) \approx s(V) + Z <_e s(Z) + s(V) \approx s(V) + s(Z)$$

is ensured.

### *5.1.4 Let us prove associativity of the addition*

We now want to prove associativity of addition $+$.

It could be done by noetherian induction over one variable, but we will proceed as Spike does, and use an induction over two variables.

We want to prove the proposition:

$$\forall x\,(x \in Nat \Rightarrow \forall y\,(y \in Nat \Rightarrow \forall z\,(z \in Nat \Rightarrow x + (y + z) \approx (x + y) + z)))$$

This goal is oriented left to right by giving a lexicographic status right to left to the $+$ symbol.

An induction on $y$ and $z$ reduces the proof to:

$$\begin{aligned} &\mathcal{T}h, Th_{Nat}^{sort} \\ &\vdash_{\mathcal{RE}_{Nat}^{def+} \cup \mathcal{RE}_{Ind(x+(y+z)\approx(x+y)+z,[y,z])}([0,0])} \\ &\forall x\,(x \in Nat \Rightarrow x + (0 + 0) \approx (x + 0) + 0) \end{aligned} \tag{16}$$

and

$$\begin{aligned}&\mathcal{T}h, Th^{sort}_{Nat}, Z \in Nat\\ &\vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(x+(y+z)\approx(x+y)+z,[y,z])}([0,s(Z)])}\\ &\forall x\,(x \in Nat \Rightarrow x + (0 + s(Z)) \approx (x + 0) + s(Z))\end{aligned}\tag{17}$$

and

$$\begin{aligned}&\mathcal{T}h, Th^{sort}_{Nat}, W \in Nat\\ &\vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(x+(y+z)\approx(x+y)+z,[y,z])}([s(W),0])}\\ &\forall x\,(x \in Nat \Rightarrow x + (s(W) + 0) \approx (x + s(W)) + 0)\end{aligned}\tag{18}$$

and

$$\begin{aligned}&\mathcal{T}h, Th^{sort}_{Nat}, Z \in Nat, W \in Nat\\ &\vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}_{Ind(x+(y+z)\approx(x+y)+z,[y,z])}([s(W),s(Z)])}\\ &\forall x\,(x \in Nat \Rightarrow x + (s(W) + s(Z)) \approx (x + s(W)) + s(Z))\end{aligned}\tag{19}$$

(16) and (18) are trivially concluded, by simplification using the user theory.

(17) is reduced as follows:

$$\begin{aligned}& x + (0 + s(Z)) \approx (x + 0) + s(Z)\\ \longrightarrow\ & x + s(0 + Z) \approx (x + 0) + s(Z)\\ \longrightarrow\ & s(x + (0 + Z)) \approx (x + 0) + s(Z)\\ \longrightarrow\ & s((x + 0) + Z) \approx (x + 0) + s(Z)\\ \longrightarrow\ & s(x + Z) \approx (x + 0) + s(Z)\\ \longrightarrow\ & s(x + Z) \approx x + s(Z)\\ \longrightarrow\ & s(x + Z) \approx s(x + Z)\end{aligned}$$

(19) is reduced as follows:

$$\begin{aligned}& x + (s(W) + s(Z)) \approx (x + s(W)) + s(Z)\\ \longrightarrow\ & x + s(s(W) + Z) \approx (x + s(W)) + s(Z)\\ \longrightarrow\ & s(x + (s(W) + Z)) \approx (x + s(W)) + s(Z)\\ \longrightarrow\ & s((x + s(W)) + Z) \approx (x + s(W)) + s(Z)\\ \longrightarrow\ & s(s(x + W) + Z) \approx (x + s(W)) + s(Z)\\ \longrightarrow\ & s(s(x + W) + Z) \approx s((x + s(W)) + Z)\\ \longrightarrow\ & s(s(x + W) + Z) \approx s(s(x + W) + Z)\end{aligned}$$

### *5.1.5 An instance of crossed definitions: Even and Odd*

Although they cause no trouble in our framework or in induction by rewriting, crossed definitions are not syntactically possible in NQTHM and are handled by an ad hoc syntax in ACL2.

We expand the definitions of section 5.1.1 with:

$$Th^{sort}_{Bool} = \begin{cases} \forall x\,(x \in Bool \Leftrightarrow \\ \quad (x{:=:}true \vee x{:=:}false \vee \\ \qquad \exists y (y \in Nat \wedge x{:=:}Even(y)) \vee \exists y (y \in Nat \wedge x{:=:}Odd(y)))) \end{cases}$$

$$Th^{def(Even)}_{Bool} = \begin{cases} Even(0) \approx true \\ \forall x\,(x \in Nat \Rightarrow Even(s(x)) \approx Odd(x)) \end{cases}$$

$$Th^{def(Odd)}_{Bool} = \begin{cases} Odd(0) \approx false \\ \forall x\,(x \in Nat \Rightarrow Odd(s(x)) \approx Even(x)) \end{cases}$$

We want to prove the proposition:

$$\forall z\,(z \in Nat \Rightarrow Even(z+z) \approx true) \wedge \forall x\,\forall y\,((x \in Nat \wedge y \in Nat) \Rightarrow s(x)+y \approx s(x+y))$$

Internalizing the induction hypothesis on $z$ and using Lemma 4.3, reduces the proof to:

$$\begin{array}{l} \mathcal{T}h, Th^{sort}_{Nat}, Th^{sort}_{Bool} \\ \vdash \\ \quad \mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}^{def(Even)}_{Bool} \cup \mathcal{RE}^{def(Odd)}_{Bool} \cup \mathcal{RE}_{Ind(Even(z+z)\approx true, z)}(0) \cup \\ \quad \mathcal{RE}_{Ind(s(x)+y \approx s(x+y), Even(z+z)\approx true, x, z)}(0) \cup \\ \quad \mathcal{RE}_{Ind(s(x)+y \approx s(x+y), Even(z+z)\approx true, y, z)}(0) \\ Even(0+0) \approx true \wedge \forall x\,\forall y\,((x \in Nat \wedge y \in Nat) \Rightarrow s(x)+y \approx s(x+y)) \end{array}$$

and

$$\begin{array}{l}\mathcal{T}h, Th^{sort}_{Nat}, Th^{sort}_{Bool}, W \in Nat \\ \vdash \\ \quad \mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}^{def(Even)}_{Bool} \cup \mathcal{RE}^{def(Odd)}_{Bool} \cup \mathcal{RE}_{Ind(Even(z+z)\approx true,z)}(s(W)) \cup \\ \quad \mathcal{RE}_{Ind(s(x)+y\approx s(x+y),Even(z+z)\approx true,x,z)}(s(W)) \cup \\ \quad \mathcal{RE}_{Ind(s(x)+y\approx s(x+y),Even(z+z)\approx true,y,z)}(s(W)) \\ Even(s(W)+s(W)) \approx true \wedge \forall x \, \forall y \, ((x \in Nat \wedge y \in Nat) \Rightarrow s(x)+y \approx s(x+y))\end{array}$$

By sequent calculus we get:

$$\begin{array}{l}\mathcal{T}h, Th^{sort}_{Nat}, Th^{sort}_{Bool} \\ \vdash \\ \quad \mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}^{def(Even)}_{Bool} \cup \mathcal{RE}^{def(Odd)}_{Bool} \cup \mathcal{RE}_{Ind(Even(z+z)\approx true,z)}(0) \cup \\ \quad \mathcal{RE}_{Ind(s(x)+y\approx s(x+y),Even(z+z)\approx true,x,z)}(0) \cup \\ \quad \mathcal{RE}_{Ind(s(x)+y\approx s(x+y),Even(z+z)\approx true,y,z)}(0) \\ Even(0+0) \approx true\end{array} \tag{20}$$

and

$$\begin{array}{l}\mathcal{T}h, Th^{sort}_{Nat}, Th^{sort}_{Bool}, W \in Nat \\ \vdash \\ \quad \mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}^{def(Even)}_{Bool} \cup \mathcal{RE}^{def(Odd)}_{Bool} \cup \mathcal{RE}_{Ind(Even(z+z)\approx true,z)}(s(W)) \cup \\ \quad \mathcal{RE}_{Ind(s(x)+y\approx s(x+y),Even(z+z)\approx true,x,z)}(s(W)) \cup \\ \quad \mathcal{RE}_{Ind(s(x)+y\approx s(x+y),Even(z+z)\approx true,y,z)}(s(W)) \\ Even(s(W)+s(W)) \approx true\end{array} \tag{21}$$

and

$$\begin{array}{l}\mathcal{T}h, Th^{sort}_{Nat}, Th^{sort}_{Bool}, W \in Nat \\ \vdash_{\mathcal{RE}^{def+}_{Nat} \cup \mathcal{RE}^{def(Even)}_{Bool} \cup \mathcal{RE}^{def(Odd)}_{Bool}} \\ \forall x \, \forall y \, ((x \in Nat \wedge y \in Nat) \Rightarrow s(x)+y \approx s(x+y))\end{array} \tag{22}$$

(20) is trivial and (22) is proved by a simple induction.

(21) is reduced as follows:

$$\begin{aligned} & Even(s(W)+s(W)) \approx true \\ \longrightarrow\ & Even(s(s(W)+W)) \approx true \\ \longrightarrow\ & Odd(s(W)+W) \approx true \\ \longrightarrow\ & Odd(s(W+W)) \approx true \\ \longrightarrow\ & Even(W+W) \approx true \\ \longrightarrow\ & true \approx true \end{aligned}$$

The interesting step is here:

$$Odd(s(W)+W) \approx true \longrightarrow Odd(s(W+W)) \approx true$$

Which corresponds to the use of $\mathcal{RE}_{Ind(s(x)+y\approx s(x+y),Even(z+z)\approx true,y,z)}(s(W))$.

Its induction condition is trivially verified since $W < s(W)$.

### *5.2 The general case*

Generalizing the ideas illustrated in the previous example, we assume given:

- a set of sort definitions,
- an equational user theory $Th_u$,
- an equational property $Q$ of the form $t_1(x) \approx t_2(x)$ to be proved.

By application of Theorem 4.1, $Q$ is an inductive consequence of $Th_u$ if:

$$\mathcal{T}h, Th_u, X \in \tau \vdash_{\mathcal{RE}_{Ind(Q,x)}(X)} Q(X)$$

Assume that $\mathcal{RE}_{Th_u}$ is orientable in a terminating term rewrite system denoted $\vec{\mathcal{RE}}_{Th_u}$ and whose rewrite relation is contained in a simplification ordering $\prec$ total on ground terms.

Let us finally define the rewrite relation $\rightsquigarrow$ similarly (but a bit more general) as [KR95, page 144]:

(A) if $t_1 \succ t_2$ then

$u \rightsquigarrow v \Leftrightarrow$

$u(\xrightarrow{1}_{\vec{\mathcal{RE}}_{Th_u}} \cup \xrightarrow{1}{}^{\neq\Lambda}_{t_1\rightarrow t_2}) \circ (\xrightarrow{*}_{\vec{\mathcal{RE}}_{Th_u}\cup\{t_1\rightarrow t_2\}})v$

(B) in all the cases

$$
\begin{aligned}
&u \rightsquigarrow v \Leftrightarrow \\
&u(\xleftrightarrow{1}{}^{\neq\Lambda}_{t_1\approx t_2} \circ \longrightarrow_{\vec{\mathcal{RE}}_{Th_u}})\circ \\
&\qquad\qquad (\xleftrightarrow{*}_{t_1\approx t_2} \circ \longrightarrow_{\vec{\mathcal{RE}}_{Th_u}} \circ \xleftrightarrow{*}_{t_1\approx t_2})^* v
\end{aligned}
$$

**Proposition 5.1** Let $Q'$ such that

$$Q(X) \rightsquigarrow Q'(X)$$

$Q(X)$ is an inductive consequence of $Th_u$ if

$$\mathcal{T}h, Th_u, X \in \tau \vdash_{\mathcal{RE}_{Th_u} \cup \mathcal{RE}_{Ind(Q,x)}(X)} Q'(X)$$

**Proof:** This is a consequence of Theorem 4.1 together with the restrictions put on $Th_u$ and on the fact that $\vec{\mathcal{RE}}_{Th_u} \subseteq \succ$. □

This provides an alternative approach to the induction by rewriting introduced by E. Kounalis and M. Rusinowitch in [KR90b,KR95] (Proposition 5.1 is a reformulation of [KR95, Theorem 6.2, page 144] with the main advantage that it could be combined with explicit induction. It also shows the advantage introduced by deduction modulo to internalize normalization techniques in general deduction processes. Another interest of this approach is related to the skeptical interactions between provers. In this case, a prover should not only find a proof but also make it available to the prover it is cooperating with. In the approach described above, the automated proof by rewriting performed, because of compatibility, using $\mathcal{RE}_{Th_u} \cup \mathcal{RE}_{Ind(Q,x)}(X)$ can be faithfully reflected at the level of sequent calculus.

## 6 Conclusion

Until now the worlds of explicit and implicit inductions were understood separately. Thanks to the general notion of deduction modulo, we have shown that implicit induction can be seen as a way to internalize explicit induction using term rewriting techniques. This provides a uniform way to integrate explicit and implicit induction and thus to have a well understood framework to combine interactive and automatic inductive theorem provers.

In addition to provide a unified framework to understand and study the so called induction by rewriting technique, this framework allows us to go further since it gives the exact hypotheses needed to perform the inductive proofs. Moreover it allows us to chose the right subset of the theory under consideration that we want to internalize and on which standard automated deduction techniques apply. Consequently it provides a formal setting for the cooperation of proof assistants providing inductive capabilities like COQ and automated proof tools based on rewriting.

We did not formalize inductionless induction method (also called proof by consistency [Com01]) in our setting. But this could be done.

Since we have used the presentation in deduction modulo of $HOL_{\lambda\sigma}$ this allowed us to ripe benefits of the framework developed in [DHK98,DHK01]. But all the benefits have not yet been exploited, in particular, we did not use the possibility to rewrite not only terms but also propositions (except in $\lambda\sigma\mathcal{L}$). Moreover the use for induction of the proof search method developed in [DHK98] has also to be investigated.

Finally, the framework developed here has been designed for initial semantics, but it can be applied for other semantics of interest like final semantics [BBR98]. We therefore plan to apply the same techniques for combining assisted and automated reasoning in final models.

We are now designing a family of proof search method based on the approach presented here and in particular on narrowing. Indeed, the theorem 4.1 enlightens the importance of the "instantiate to reduce" paradigm which is just the narrowing concept.

*Acknowledgments*

This paper would not have existed without the many discussions and our many interactions with Gilles Dowek and Thérèse Hardin. We greatly appreciated our discussions with Michael Rusinowitch, and his always fruitful comments and hints about proof by induction and the Spike system. We also thank Hubert Comon for his comments on a preliminary version of this work.

## References

[ACCL91] Martin Abadi, Lucas Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. *Journal of Functional Programming*, 1(4):375–416, 1991.

[And86] P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth through Proof.* Academic Press inc., New York, 1986.

[BBR98] N. Berregeb, A. Bouhoula, and M. Rusinowitch. Observational proofs with critical contexts. In *Fundamental Approaches to Software*

*Engineering (ETAP 98)*, volume 1382 of *Lecture Notes in Computer Science*, pages 38–53. Springer-Verlag, 1998.

[BJ99] A. Bouhoula and F. Jacquemard. Automata based induction, 1999. Draft.

[BKR92] Adel Bouhoula, E. Kounalis, and M. Rusinowitch. Spike: An automatic theorem prover. In *Proceedings of the 1st International Conference on Logic Programming and Automated Reasoning, St. Petersburg (Russia)*, volume 624 of *Lecture Notes in Artificial Intelligence*, pages 460–462. Springer-Verlag, July 1992.

[BKR95] Adel Bouhoula, Emmanuel Kounalis, and Michaël Rusinowitch. Automated Mathematical Induction. *Journal of Logic and Computation*, 5(5):631–668, 1995.

[BM03] Roberto Bruni and José Meseguer. Generalized rewrite theories. In ?, editor, *Thirtieth International Colloquium on Automata, Languages and Programming*, volume ? of *Lecture Notes in Computer Science*, pages ?–? Springer-Verlag, 2003.

[BMM02] Roberto Bruni, José Meseguer, and Ugo Montanari. Tiling transactions in rewriting logic. In Fabio Gadducci and Ugo Montanari, editors, *Proceedings of the $4^{th}$ International Workshop on Rewriting Logic and Its Applications*, volume 71, pages 43–62, Pisa, Italy, September 2002. Elsevier Science. Preliminary version.

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and all That*. Cambridge University Press, 1998.

[Bou94] Adel Bouhoula. *Preuves Automatiques par Récurrence dans les Théories Conditionnelles*. Thèse de Doctorat d'Université, Université Henri Poincaré – Nancy 1, March 1994.

[Bun01] A. Bundy. The automation of proof by mathematical induction. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 13, pages 845–911. Elsevier Science, 2001.

[Chu40] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[Com01] H. Comon. Inductionless induction. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 14, pages 913–962. Elsevier Science, 2001.

[Dep02] Eric Deplagne. *Système de preuve modulo récurrence*. Thèse de doctorat, Université Nancy 1, November 2002.

[Der87] N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1 & 2):69–116, 1987. Special issue on Rewriting Techniques and Applications.

[DHK98] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. Rapport de Recherche 3400, Institut National de Recherche en Informatique et en Automatique, April 1998. `ftp://ftp.inria.fr/INRIA/publication/RR/RR-3400.ps.gz`.

[DHK00] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Higher-order unification via explicit substitutions. *Information and Computation*, 157(1/2):183–235, 2000.

[DHK01] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. *Mathematical Structures in Computer Science*, 11(1):21–45, 2001.

[DHK03] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31(1):33–72, Nov 2003.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite Systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 6, pages 244–320. Elsevier Science Publishers B. V. (North-Holland), 1990.

[DKKN03] Eric Deplagne, Claude Kirchner, Hélène Kirchner, and Quang-Huy Nguyen. Proof search and proof check for equational and inductive theorems. In *Proceedings 19th International Conference on Automated Deduction, Miami (Florida, USA)*, Lecture Notes in Computer Science. Springer-Verlag, 2003. Invited Talk.

[DO90] N. Dershowitz and M. Okada. A rationale for conditional equational programming. *Theoretical Computer Science*, 75:111–138, 1990.

[Gal86] J. H. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*, volume 5 of *Computer Science and Technology Series*. Harper & Row, New York, 1986.

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[HK88] D. Hofbauer and R. D. Kutsche. Proving inductive theorems based on term rewriting systems. In J. Grabowski, P. Lescanne, and W. Wechler, editors, *Proceedings 1st International Workshop on Algebraic and Logic Programming*, pages 180–190. Akademie Verlag, 1988.

[Hue72] G. Huet. *Constrained Resolution: A Complete Method for Type Theory*. PhD thesis, Case Western Reserve University, 1972.

[Hue86] Gérard Huet. Induction principles formalized in the calculus of constructions. In K. Fuchi and M. Nivat, editors, *Programming of Future Generation Computers: Proceedings of the first Franco-Japanese Symposium on Programming of Future Generation Computers*, pages 205–216, Tokyo, Japan, October 1986. Elsevier Science Publishers B. V. (North-Holland) 1988.

[KK99] Claude Kirchner and Hélène Kirchner. Rewriting, solving, proving. A preliminary version of a book available at `www.loria.fr/~ckirchne/rsp.ps.gz`, 1999.

[KR90a] E. Kounalis and M. Rusinowitch. Mechanizing inductive reasoning. *Bulletin of European Association for Theoretical Computer Science*, 41:216–226, June 1990.

[KR90b] E. Kounalis and M. Rusinowitch. Mechanizing inductive reasoning. In *Proceedings of the American Association for Artificial Intelligence Conference, Boston*, pages 240–245. AAAI Press and MIT Press, July 1990.

[KR95] E. Kounalis and M. Rusinowitch. Reasoning with conditional axioms. *Annals of Mathematics and Artificial Intelligence*, 15(2):125–149, 1995.

[KZ95] Deepak Kapur and Hantao Zhang. An overview of rewrite rule laboratory (rrl). *J. Computer and Mathematics with Applications*, 29(2):91–114, 1995.

[Lan81] D. S. Lankford. A simple explanation of inductionless induction. Technical report, Louisiana Tech, Ruston (Louisiana), 1981.

[PS81] G. Peterson and M. E. Stickel. Complete sets of reductions for some equational theories. *Journal of the ACM*, 28:233–264, 1981.

[Red90] U. S. Reddy. Term rewriting induction. In M. E. Stickel, editor, *Proceedings 10th International Conference on Automated Deduction, Kaiserslautern (Germany)*, volume 449 of *Lecture Notes in Computer Science*, pages 162–177. Springer-Verlag, 1990.

[Ter02] Terese (M. Bezem, J. W. Klop and R. de Vrijer, eds). *Term Rewriting Systems*. Cambridge University Press, 2002.

[Wec92] Wolfgang Wechler. *Universal Algebra for Computer Scientists*, volume 25 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1992.

[ZKK88] H. Zhang, D. Kapur, and M. S. Krishnamoorthy. A mechanizable induction principle for equational specifications. In E. Lusk and R. Overbeek, editors, *Proceedings 9th International Conference on Automated Deduction, Argonne (Ill., USA)*, volume 310 of *Lecture Notes in Computer Science*, pages 162–181. Springer-Verlag, 1988.

# A Proof of lemma 2.1

We introduce two rules that can be deduced from sequent calculus and will simplify the proof :

- The first rule we introduce is the well-known *modus ponens*, we call it mp-r.

$$\dfrac{\Gamma \vdash A, \Delta \quad \Gamma \vdash A \Rightarrow B, \Delta}{\Gamma \vdash B, \Delta}\ \text{mp-r}$$

Suppose we have proofs $\pi$ and $\rho$ such that:

$$\dfrac{}{\Gamma \vdash A, \Delta}\ \pi$$

and

$$\dfrac{}{\Gamma \vdash A \Rightarrow B, \Delta}\ \rho$$

We build the proof:

$$\dfrac{\dfrac{}{\Gamma, A \Rightarrow B \vdash B, \Delta}\ \delta \quad \dfrac{\dfrac{}{\Gamma \vdash A \Rightarrow B, \Delta}\ \rho}{\Gamma \vdash A \Rightarrow B, B, \Delta}\ \text{weak-r}}{\Gamma \vdash B, \Delta}\ \text{cut}$$

Where $\delta$ is the proof:

$$\dfrac{\dfrac{}{\Gamma, B \vdash B, \Delta}\ \text{axiom} \quad \dfrac{\dfrac{}{\Gamma \vdash A, \Delta}\ \pi}{\Gamma \vdash A, B, \Delta}\ \text{weak-r}}{\Gamma, A \Rightarrow B \vdash B, \Delta}\ \Rightarrow\text{-l}$$

Hence we can use mp-r.

- The second rule we introduce is a left variant of *modus ponens*, we call it mp-l.

$$\dfrac{\Gamma, A \vdash \Delta \quad \Gamma \vdash B \Rightarrow A, \Delta}{\Gamma, B \vdash \Delta}\ \text{mp-l}$$

Suppose we have proofs $\pi$ and $\rho$ such that:

$$\dfrac{}{\Gamma, A \vdash \Delta}\ \pi$$

and

$$\dfrac{}{\Gamma \vdash B \Rightarrow A, \Delta}\ \rho$$

We build the proof:

$$\dfrac{\dfrac{}{\Gamma, B, B \Rightarrow A \vdash \Delta}\ \delta \quad \dfrac{\dfrac{}{\Gamma \vdash B \Rightarrow A, \Delta}\ \rho}{\Gamma, B \vdash B \Rightarrow A, \Delta}\ \text{weak-l}}{\Gamma, B \vdash \Delta}\ \text{cut}$$

Where $\delta$ is the proof:

$$\dfrac{\dfrac{\dfrac{}{\Gamma, A \vdash \Delta}\,\pi}{\Gamma, B, A \vdash \Delta}\,\text{weak-l} \qquad \dfrac{}{\Gamma, B \vdash B, \Delta}\,\text{axiom}}{\Gamma, B, B \Rightarrow A \vdash \Delta}\,\Rightarrow\text{-l}$$

Hence we can use mp-l.

**Lemma 2.1** Let $\mathcal{T}$ and $\sim$ be a theory and an equivalence. The two following statements are equivalent:

(A) for all propositions $P$ and $Q$ and context $\Gamma$,

$$P \sim^{\Gamma} Q \text{ implies } \mathcal{T},\mathcal{U},\Gamma \vdash P \Leftrightarrow Q,$$

(B) $\mathcal{T},\mathcal{U},\Gamma \vdash \Delta$ if and only if $\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} \Delta$.

**Proof:**

- From A to B, the proof is similar to the one presented in [DHK98].
  - · The "only if" part is an obvious induction on the structure of the derivation of $\mathcal{T},\mathcal{U},\Gamma \vdash \Delta$, expliciting the witnesses using first-order matching.
  - · For the "if" part, we first notice that using the contraction rule any proof of $\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} \Delta$ can be transformed into another where the propositions of $\mathcal{T}$ and $\mathcal{U}$ appear in the left part of every sequent.

    We then proceed by induction on the structure of the proof:

    axiom

    The proof has the form:

$$\dfrac{}{\mathcal{T},\mathcal{U},\Gamma, P \vdash_{\sim} Q}\,\text{axiom} \quad \text{where } P \sim^{\mathcal{T},\mathcal{U},\Gamma} Q.$$

We trivially have:

$$\dfrac{}{\mathcal{T},\mathcal{U},\Gamma, P \vdash P}\,\text{axiom}$$

We have $P \sim^{\mathcal{T},\mathcal{U},\Gamma} Q$, thus from A we get $\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow Q$. Using either mp-r or mp-l, we build a proof of $\mathcal{T},\mathcal{U},\Gamma, P \vdash Q$.

cut

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma, P \vdash_{\sim} \Delta}\,\pi \qquad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} Q, \Delta}\,\rho}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} \Delta}\,\text{cut} \quad \text{where } P \sim^{\mathcal{T},\mathcal{U},\Gamma} Q.$$

By induction hypothesis we have proofs $\pi'$ and $\rho'$ of $\mathcal{T},\mathcal{U},\Gamma \vdash P, \Delta$ and $\mathcal{T},\mathcal{U},\Gamma \vdash Q, \Delta$.

We have $P \sim^{\mathcal{T},\mathcal{U},\Gamma} Q$, thus from A we get $\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow Q$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow Q, \Delta$. We call $\delta$ a proof of this sequent.

We build the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,P\vdash\Delta}\,\pi' \quad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma\vdash P,\Delta}\,\gamma}{\mathcal{T},\mathcal{U},\Gamma\vdash\Delta}\,\text{cut}$$

where $\gamma$ is the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma\vdash P,\Delta}\,\rho' \quad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma\vdash Q\Rightarrow P,\Delta}\,\delta}{\mathcal{T},\mathcal{U},\Gamma\vdash P,\Delta}\,\text{mp-r}$$

We can also build the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,Q\vdash\Delta}\,\gamma' \quad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma\vdash Q,\Delta}\,\rho'}{\mathcal{T},\mathcal{U},\Gamma\vdash\Delta}\,\text{cut}$$

where $\gamma'$ is the proof

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,P\vdash\Delta}\,\pi' \quad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma\vdash Q\Rightarrow P,\Delta}\,\delta}{\mathcal{T},\mathcal{U},\Gamma,Q\vdash\Delta}\,\text{mp-l}$$

contr-l

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,Q,R\vdash_{\sim}\Delta}\,\pi}{\mathcal{T},\mathcal{U},\Gamma,P\vdash_{\sim}\Delta}\,\text{contr-l}\quad \text{where } P\sim^{\mathcal{T},\mathcal{U},\Gamma}Q \text{ and } P\sim^{\mathcal{T},\mathcal{U},\Gamma}R.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma,Q,R\vdash\Delta$. We have $P\sim^{\mathcal{T},\mathcal{U},\Gamma}Q$, thus from A we get $\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow Q$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow Q,\Delta$. We call $\delta$ a proof of this sequent.
We have $P\sim^{\mathcal{T},\mathcal{U},\Gamma}R$, thus from A we get $\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow R$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow R,\Delta$. We call $\varepsilon$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,Q,P\vdash\Delta}\,\gamma \quad \dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow Q,\Delta}\,\delta}{\mathcal{T},\mathcal{U},\Gamma,P\vdash P\Rightarrow Q,\Delta}\,\text{weak-l}}{\mathcal{T},\mathcal{U},\Gamma,P,P\vdash\Delta}\,\text{mp-l}}{\mathcal{T},\mathcal{U},\Gamma,P\vdash\Delta}\,\text{contr-l}$$

where $\gamma$ is the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,Q,R\vdash\Delta}\,\pi' \quad \dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,Q\vdash P\Rightarrow R,\Delta}\,\varepsilon}{\mathcal{T},\mathcal{U},\Gamma,Q\vdash P\Rightarrow R,\Delta}\,\text{weak-l}}{\mathcal{T},\mathcal{U},\Gamma,Q,P\vdash\Delta}\,\text{mp-l}$$

contr-r

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma\vdash_{\sim}Q,R,\Delta}\,\pi}{\mathcal{T},\mathcal{U},\Gamma\vdash_{\sim}P,\Delta}\,\text{contr-r}\quad \text{where } P\sim^{\mathcal{T},\mathcal{U},\Gamma}Q \text{ and } P\sim^{\mathcal{T},\mathcal{U},\Gamma}R.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma \vdash Q, R, \Delta$.
We have $P \sim^{\mathcal{T},\mathcal{U},\Gamma} Q$, thus from A we get $\mathcal{T},\mathcal{U},\Gamma \vdash Q \Rightarrow P$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash Q \Rightarrow P, \Delta$. We call $\delta$ a proof of this sequent.
We have $P \sim^{\mathcal{T},\mathcal{U},\Gamma} R$, thus from A we get $\mathcal{T},\mathcal{U},\Gamma \vdash R \Rightarrow P$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash R \Rightarrow P, \Delta$. We call $\varepsilon$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash Q, P, \Delta}\,\gamma \quad \dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash Q \Rightarrow P, \Delta}\,\delta}{\mathcal{T},\mathcal{U},\Gamma \vdash Q \Rightarrow P, P, \Delta}\,\text{weak-r}}{\mathcal{T},\mathcal{U},\Gamma \vdash P, P, \Delta}\,\text{mp-r}}{\mathcal{T},\mathcal{U},\Gamma \vdash P, \Delta}\,\text{contr-r}$$

where $\gamma$ is the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash Q, R, \Delta}\,\pi' \quad \dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash R \Rightarrow P, \Delta}\,\varepsilon}{\mathcal{T},\mathcal{U},\Gamma \vdash Q, R \Rightarrow P, \Delta}\,\text{weak-r}}{\mathcal{T},\mathcal{U},\Gamma \vdash Q, P, \Delta}\,\text{mp-r}$$

weak-l

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} \Delta}\,\pi}{\mathcal{T},\mathcal{U},\Gamma, P \vdash_{\sim} \Delta}\,\text{weak-l}$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma \vdash \Delta$.
We build the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash \Delta}\,\pi'}{\mathcal{T},\mathcal{U},\Gamma, P \vdash \Delta}\,\text{weak-l}$$

weak-r

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} \Delta}\,\pi}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} P, \Delta}\,\text{weak-r}$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma \vdash \Delta$.
We build the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash \Delta}\,\pi'}{\mathcal{T},\mathcal{U},\Gamma \vdash P, \Delta}\,\text{weak-r}$$

$\wedge$-l

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma, P, Q \vdash_{\sim} \Delta}\,\pi}{\mathcal{T},\mathcal{U},\Gamma, R \vdash_{\sim} \Delta}\,\wedge\text{-l} \quad \text{where } R \sim^{\mathcal{T},\mathcal{U},\Gamma} P \wedge Q.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma,P,Q\vdash\Delta$.
We have $R\sim^{\mathcal{T},\mathcal{U},\Gamma} P\wedge Q$, thus from A we get $\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow(P\wedge Q)$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow(P\wedge Q),\Delta$.
We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,P,Q\vdash\Delta}\;\pi'}{\mathcal{T},\mathcal{U},\Gamma,P\wedge Q\vdash\Delta}\;\wedge\text{-r}\qquad\overline{\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow(P\wedge Q),\Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma,R\vdash\Delta}\;\text{mp-l}$$

$\wedge$-r

The proof has the form:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma\vdash_{\sim}P,\Delta}\;\pi\qquad\overline{\mathcal{T},\mathcal{U},\Gamma\vdash_{\sim}Q,\Delta}\;\rho}{\mathcal{T},\mathcal{U},\Gamma\vdash_{\sim}R,\Delta}\;\wedge\text{-r}\quad\text{where } R\sim^{\mathcal{T},\mathcal{U},\Gamma}P\wedge Q.$$

By induction hypothesis we have proofs $\pi'$ and $\rho'$ of $\mathcal{T},\mathcal{U},\Gamma\vdash P,\Delta$ and $\mathcal{T},\mathcal{U},\Gamma\vdash Q,\Delta$.
We have $R\sim^{\mathcal{T},\mathcal{U},\Gamma} P\wedge Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma\vdash(P\wedge Q)\Rightarrow R$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma\vdash(P\wedge Q)\Rightarrow R,\Delta$.
We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma\vdash P\wedge Q,\Delta}\;\gamma\qquad\overline{\mathcal{T},\mathcal{U},\Gamma\vdash(P\wedge Q)\Rightarrow R,\Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma\vdash R,\Delta}\;\text{mp-r}$$

where $\gamma$ is the proof:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma\vdash P,\Delta}\;\pi'\qquad\overline{\mathcal{T},\mathcal{U},\Gamma\vdash Q,\Delta}\;\rho'}{\mathcal{T},\mathcal{U},\Gamma\vdash P\wedge Q,\Delta}\;\wedge\text{-r}$$

$\vee$-l

The proof has the form:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,P\vdash_{\sim}\Delta}\;\pi\qquad\overline{\mathcal{T},\mathcal{U},\Gamma,Q\vdash_{\sim}\Delta}\;\rho}{\mathcal{T},\mathcal{U},\Gamma,R\vdash_{\sim}\Delta}\;\vee\text{-l}\quad\text{where } R\sim^{\mathcal{T},\mathcal{U},\Gamma}P\vee Q.$$

By induction hypothesis we have proofs $\pi'$ and $\rho'$ of $\mathcal{T},\mathcal{U},\Gamma,P\vdash\Delta$ and $\mathcal{T},\mathcal{U},\Gamma,Q\vdash\Delta$.
We have $R\sim^{\mathcal{T},\mathcal{U},\Gamma} P\vee Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow(P\wedge Q)$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow(P\wedge Q),\Delta$.
We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,P\vee Q\vdash\Delta}\;\gamma\qquad\overline{\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow(P\vee Q),\Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma,R\vdash\Delta}\;\text{mp-l}$$

where $\gamma$ is the proof:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,P\vdash\Delta}\;\pi'\qquad\overline{\mathcal{T},\mathcal{U},\Gamma,Q\vdash\Delta}\;\rho'}{\mathcal{T},\mathcal{U},\Gamma,P\vee Q\vdash\Delta}\;\vee\text{-l}$$

$\vee$-r

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} P,Q,\Delta}\,\pi}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} R,\Delta}\,\vee\text{-r} \quad \text{where } R \sim^{\mathcal{T},\mathcal{U},\Gamma} P \vee Q.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma \vdash P,Q,\Delta$.
We have $R \sim^{\mathcal{T},\mathcal{U},\Gamma} P \vee Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma \vdash (P \wedge Q) \Rightarrow R$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash (P \wedge Q) \Rightarrow R,\Delta$.
We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash P,Q,\Delta}\,\pi'}{\mathcal{T},\mathcal{U},\Gamma \vdash P \vee Q,\Delta}\,\vee\text{-r} \qquad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash (P \vee Q) \Rightarrow R,\Delta}\,\delta}{\mathcal{T},\mathcal{U},\Gamma \vdash R,\Delta}\,\text{mp-r}$$

$\Rightarrow$-l

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} P,\Delta}\,\pi \qquad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma,Q \vdash_{\sim} \Delta}\,\rho}{\mathcal{T},\mathcal{U},\Gamma,R \vdash_{\sim} \Delta}\,\Rightarrow\text{-l} \quad \text{where } R \sim^{\mathcal{T},\mathcal{U},\Gamma} P \Rightarrow Q.$$

By induction hypothesis we have proofs $\pi'$ and $\rho'$ of $\mathcal{T},\mathcal{U},\Gamma \vdash P,\Delta$ and $\mathcal{T},\mathcal{U},\Gamma,Q \vdash \Delta$.
We have $R \sim^{\mathcal{T},\mathcal{U},\Gamma} P{\Rightarrow}Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma \vdash R \Rightarrow (P{\Rightarrow}Q)$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash R \Rightarrow (P{\Rightarrow}Q),\Delta$.
We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,P \Rightarrow Q \vdash \Delta}\,\gamma \qquad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash R \Rightarrow (P \Rightarrow Q),\Delta}\,\delta}{\mathcal{T},\mathcal{U},\Gamma,R \vdash \Delta}\,\text{mp-l}$$

where $\gamma$ is the proof:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma \vdash P,\Delta}\,\pi' \qquad \dfrac{}{\mathcal{T},\mathcal{U},\Gamma,Q \vdash \Delta}\,\rho'}{\mathcal{T},\mathcal{U},\Gamma,P \Rightarrow Q \vdash \Delta}\,\Rightarrow\text{-l}$$

$\Rightarrow$-r

The proof has the form:

$$\dfrac{\dfrac{}{\mathcal{T},\mathcal{U},\Gamma,P \vdash_{\sim} Q,\Delta}\,\pi}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} R,\Delta}\,\Rightarrow\text{-r} \quad \text{where } R \sim^{\mathcal{T},\mathcal{U},\Gamma} P \Rightarrow Q.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma,P \vdash Q,\Delta$.
We have $R \sim^{\mathcal{T},\mathcal{U},\Gamma} P{\Rightarrow}Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma \vdash (P{\Rightarrow}Q) \Rightarrow R$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash (P{\Rightarrow}Q) \Rightarrow R,\Delta$.
We call $\delta$ a proof of this sequent.

We build the proof:

$$\dfrac{\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,P\vdash Q,\Delta}\;\pi'}{\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow Q,\Delta}\Rightarrow\text{-r}\qquad \overline{\mathcal{T},\mathcal{U},\Gamma\vdash(P\Rightarrow Q)\Rightarrow R,\Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma\vdash R,\Delta}\text{mp-r}$$

$\neg$-l

The proof has the form:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma\vdash_{\sim} P,\Delta}\;\pi}{\mathcal{T},\mathcal{U},\Gamma,R\vdash_{\sim}\Delta}\neg\text{-l}\quad \text{where } R\sim^{\mathcal{T},\mathcal{U},\Gamma}\neg P.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma\vdash P,\Delta$.
We have $R\sim^{\mathcal{T},\mathcal{U},\Gamma}\neg P$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow\neg P$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow\neg P,\Delta$. We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma\vdash P,\Delta}\;\pi'}{\mathcal{T},\mathcal{U},\Gamma,\neg P\vdash\Delta}\neg\text{-r}\qquad \overline{\mathcal{T},\mathcal{U},\Gamma\vdash R\Rightarrow\neg P,\Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma,R\vdash\Delta}\text{mp-l}$$

$\neg$-r

The proof has the form:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,P\vdash_{\sim}\Delta}\;\pi}{\mathcal{T},\mathcal{U},\Gamma\vdash_{\sim} R,\Delta}\neg\text{-r}\quad \text{where } R\sim^{\mathcal{T},\mathcal{U},\Gamma}\neg P.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma,P\vdash\Delta$.
We have $R\sim^{\mathcal{T},\mathcal{U},\Gamma}\neg P$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma\vdash\neg P\Rightarrow R$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma\vdash\neg P\Rightarrow R,\Delta$. We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,P\vdash\Delta}\;\pi'}{\mathcal{T},\mathcal{U},\Gamma\vdash\neg P,\Delta}\neg\text{-r}\qquad \overline{\mathcal{T},\mathcal{U},\Gamma\vdash\neg P\Rightarrow R,\Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma\vdash R,\Delta}\text{mp-r}$$

$\bot$-l

The proof has the form:

$$\overline{\mathcal{T},\mathcal{U},\Gamma,P\vdash_{\sim}\Delta}\;\bot\text{-l}\quad \text{where } P\sim^{\mathcal{T},\mathcal{U},\Gamma}\bot.$$

We have $P\sim^{\mathcal{T},\mathcal{U},\Gamma}\bot$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow\bot$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow\bot,\Delta$. We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,\bot\vdash\Delta}\;\bot\text{-l}\qquad \overline{\mathcal{T},\mathcal{U},\Gamma\vdash P\Rightarrow\bot,\Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma,P\vdash\Delta}\text{mp-l}$$

$\forall$-l

The proof has the form:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,Q\{t/x\} \vdash_{\sim} \Delta}\;\pi}{\mathcal{T},\mathcal{U},\Gamma,P \vdash_{\sim} \Delta}\;\forall\text{-l} \quad \text{where } P \sim^{\mathcal{T},\mathcal{U},\Gamma} \forall x\, Q.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma,Q\{t/x\} \vdash \Delta$. We have $P \sim^{\mathcal{T},\mathcal{U},\Gamma} \forall x\, Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow \forall x\, Q$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow \forall x\, Q, \Delta$. We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,Q\{t/x\} \vdash \Delta}\;\pi'}{\mathcal{T},\mathcal{U},\Gamma,\forall x\, Q \vdash \Delta}\;\forall\text{-l} \quad \overline{\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow \forall x\, Q, \Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma,P \vdash \Delta}\;\text{mp-l}$$

$\forall$-r

The proof has the form:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} Q\{y/x\}, \Delta}\;\pi}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} P, \Delta}\;\forall\text{-r} \quad \text{where } P \sim^{\mathcal{T},\mathcal{U},\Gamma} \forall x\, Q.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma \vdash Q\{y/x\}, \Delta$. We have $P \sim^{\mathcal{T},\mathcal{U},\Gamma} \forall x\, Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma \vdash (\forall x\, Q) \Rightarrow P$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash (\forall x\, Q) \Rightarrow P, \Delta$. We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma \vdash Q\{y/x\}, \Delta}\;\pi'}{\mathcal{T},\mathcal{U},\Gamma \vdash \forall x\, Q, \Delta}\;\forall\text{-r} \quad \overline{\mathcal{T},\mathcal{U},\Gamma \vdash (\forall x\, Q) \Rightarrow P, \Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma \vdash P, \Delta}\;\text{mp-r}$$

$\exists$-l

The proof has the form:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,Q\{y/x\} \vdash_{\sim} \Delta}\;\pi}{\mathcal{T},\mathcal{U},\Gamma,P \vdash_{\sim} \Delta}\;\exists\text{-l} \quad \text{where } P \sim^{\mathcal{T},\mathcal{U},\Gamma} \exists x\, Q.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma,Q\{y/x\} \vdash \Delta$. We have $P \sim^{\mathcal{T},\mathcal{U},\Gamma} \exists x\, Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow \exists x\, Q$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow \exists x\, Q, \Delta$. We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma,Q\{y/x\} \vdash \Delta}\;\pi'}{\mathcal{T},\mathcal{U},\Gamma,\exists x\, Q \vdash \Delta}\;\exists\text{-l} \quad \overline{\mathcal{T},\mathcal{U},\Gamma \vdash P \Rightarrow \exists x\, Q, \Delta}\;\delta}{\mathcal{T},\mathcal{U},\Gamma,P \vdash \Delta}\;\text{mp-l}$$

$\exists$-r

The proof has the form:

$$\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} Q\{t/x\},\Delta}\ \pi}{\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} P,\Delta}\ \exists\text{-r} \quad \text{where } P \sim^{\mathcal{T},\mathcal{U},\Gamma} \exists x\, Q.$$

By induction hypothesis we have a proof $\pi'$ of $\mathcal{T},\mathcal{U},\Gamma \vdash Q\{t/x\},\Delta$. We have $P \sim^{\mathcal{T},\mathcal{U},\Gamma} \exists x\, Q$, thus from A we get $\mathcal{T},\mathcal{T},\mathcal{U},\Gamma \vdash (\exists x\, Q) \Rightarrow P$, from which we easily prove $\mathcal{T},\mathcal{U},\Gamma \vdash (\exists x\, Q) \Rightarrow P,\Delta$. We call $\delta$ a proof of this sequent.
We build the proof:

$$\dfrac{\dfrac{\overline{\mathcal{T},\mathcal{U},\Gamma \vdash Q\{t/x\},\Delta}\ \pi'}{\mathcal{T},\mathcal{U},\Gamma \vdash \exists x\, Q,\Delta}\ \exists\text{-r} \qquad \overline{\mathcal{T},\mathcal{U},\Gamma \vdash (\exists x\, Q) \Rightarrow P,\Delta}\ \delta}{\mathcal{T},\mathcal{U},\Gamma \vdash P,\Delta}\ \text{mp-r}$$

- From B to A, $\mathcal{T},\mathcal{U},\Gamma \vdash P \Leftrightarrow Q$ reduces to $\mathcal{T},\mathcal{U},\Gamma \vdash_{\sim} P \Leftrightarrow Q$ by B. Then by sequent calculus modulo we easily get $\mathcal{T},\mathcal{U},\Gamma,P \vdash_{\sim} Q$ and $\mathcal{T},\mathcal{U},\Gamma,Q \vdash_{\sim} P$ which are both true by axiom since $P \sim^{\Gamma} Q$.

□

## B Proof of lemma 2.5

**Lemma 2.5** For any conditional class rewrite system $\mathcal{RE}$, the theory $\mathcal{T}_{\mathcal{RE}}$ in definition 2.8 is compatible with $\mathcal{RE}$ up to $Th_{\approx}$.

**Proof:** We shall prove that:

(A) for all propositions $P$ and $Q$ and all contexts $\Gamma$, we have $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$.

We proceed by induction on the length $n$ of the derivation $P \overset{n}{\longleftrightarrow}{}^{Th_{\approx},\Gamma}_{\mathcal{R}\cup\mathcal{E}} Q$.

- If $n = 0$ then $P = Q$ and the property is trivially satisfied,
- else $P \longleftrightarrow^{Th_{\approx},\Gamma}_{\mathcal{R}\cup\mathcal{E}} P' \overset{n-1}{\longleftrightarrow}{}^{Th_{\approx},\Gamma}_{\mathcal{R}\cup\mathcal{E}} Q$.

  There are six possibilities, according to the use of an axiom, a rule in direct way or a rule in reverse way, and according to the rule or axiom dealing with terms or propositions.

  - If the rule or axiom deals with terms, we have an axiom $\forall \overline{x}\, (c(\overline{x}) \Rightarrow (l(\overline{x}) \approx r(\overline{x})))$ in $\mathcal{T}_{\mathcal{RE}}$. We thus instantiate this axiom according to $\sigma$ and use it with the axioms of equality.

    For each case, we detail how to build a proof in sequent calculus.

    If we apply an axiom $f \approx g$ if $c$ or $g \approx f$ if $c$, we have

$$P = P[\sigma(f)]_{\omega} \approx_{\mathcal{E}}^{Th_{\approx},\Gamma} P[\sigma(g)]_{\omega} = P'$$

for some occurrence $\omega$ in $P$ such that $Prot(P,\omega,\approx)$ is false and some substitution $\sigma$ such that $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c)$.

By induction hypothesis we get $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$.
Let us now derive a proof of $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$:

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by contraction and instantiation]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(c(\overline{x}))) \Rightarrow (\sigma(f(\overline{x})) \approx \sigma(g(\overline{x}))) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by implication left and weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x}))$ and $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(f(\overline{x})) \approx \sigma(g(\overline{x})) \vdash P \Leftrightarrow Q$

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x}))$ is true by hypothesis.
$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(f(\overline{x})) \approx \sigma(g(\overline{x})) \vdash P \Leftrightarrow Q$ is proved as follows:

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(f(\overline{x})) \approx \sigma(g(\overline{x})) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [using the equality axioms]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(f(\overline{x})) \approx \sigma(g(\overline{x})) \vdash P' \Leftrightarrow Q$

$\Leftarrow$ [by weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$

which is exactly the induction hypothesis.
If we apply a rule $l \rightarrow r$ `if` $c$ in direct way, we have

$$P = P[\sigma(l)]_{\omega} \rightarrow_{\mathcal{R}}^{Th_{\approx}, \Gamma} P[\sigma(r)]_{\omega} = P'$$

for some occurrence $\omega$ in $P$ such that $Prot(P, \omega, \approx)$ is false and some substitution $\sigma$ such that $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c)$
By induction hypothesis we get $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$.
Let us now derive a proof of $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$:

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by contraction and instantiation]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(c(\overline{x}))) \Rightarrow (\sigma(l(\overline{x})) \approx \sigma(r(\overline{x}))) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by implication left and weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x})))$ and $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \approx \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x}))$ is true by hypothesis.
$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \approx \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$ is proved as follows:

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \approx \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [using the equality axioms]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \approx \sigma(r(\overline{x})) \vdash P' \Leftrightarrow Q$

$\Leftarrow$ [by weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$

which is exactly the induction hypothesis.
If we apply a rule $l \rightarrow r$ if $c$ in reverse way, we have

$$P = P[\sigma(r)]_\omega \leftarrow_{\mathcal{R}}^{Th_\approx,\Gamma} P[\sigma(l)]_\omega = P'$$

for some occurrence $\omega$ in $P'$ such that $Prot(P, \omega, \approx)$ is false and some substitution $\sigma$ such that $\sigma(l) = P'_{|\omega}$ and $\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash \sigma(c)$
By induction hypothesis we get $\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash P' \Leftrightarrow Q$.
Let us now derive a proof of $\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash P \Leftrightarrow Q$:

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by contraction and instantiation]

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma, \sigma(c(\overline{x}))) \Rightarrow (\sigma(l(\overline{x})) \approx \sigma(r(\overline{x}))) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by implication left and weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash \sigma(c(\overline{x})))$ and $\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma, \sigma(l(\overline{x})) \approx \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash \sigma(c(\overline{x}))$ is true by hypothesis.
$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma, \sigma(l(\overline{x})) \approx \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$ is proved as follows:

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma, \sigma(l(\overline{x})) \approx \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [using the equality axioms]

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma, \sigma(l(\overline{x})) \approx \sigma(r(\overline{x})) \vdash P' \Leftrightarrow Q$

$\Leftarrow$ [by weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash P' \Leftrightarrow Q$

which is exactly the induction hypothesis.

- If the rule or axiom deals with propositions, we have an axiom $\forall \overline{x}\, (c(\overline{x}) \Rightarrow (l(\overline{x}) \Leftrightarrow r(\overline{x})))$ in $\mathcal{T}_{\mathcal{RE}}$. We thus instantiate this axiom according to $\sigma$ and use the obtained equivalence.

For each case, we detail how to build a proof in sequent calculus.
If we apply an axiom $f \approx g$ if $c$ or $g \approx f$ if $c$, we have

$$P = P[\sigma(f)]_\omega \approx_{\mathcal{E}}^{Th_\approx,\Gamma} P[\sigma(g)]_\omega = P'$$

for some occurrence $\omega$ in $P$ such that $Prot(P, \omega, \approx)$ is false and some substitution $\sigma$ such that $\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash \sigma(c)$.
By induction hypothesis we get $\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash P' \Leftrightarrow Q$.
Let us now derive a proof of $\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash P \Leftrightarrow Q$:

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by contraction and instantiation]

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma, \sigma(c(\overline{x}))) \Rightarrow (\sigma(f(\overline{x})) \Leftrightarrow \sigma(g(\overline{x}))) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by implication left and weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma \vdash \sigma(c(\overline{x})))$ and $\mathcal{T}_{\mathcal{RE}}, Th_\approx, \Gamma, \sigma(f(\overline{x})) \Leftrightarrow \sigma(g(\overline{x})) \vdash P \Leftrightarrow Q$

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x}))$ is true by hypothesis.
$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(f(\overline{x})) \Leftrightarrow \sigma(g(\overline{x})) \vdash P \Leftrightarrow Q$ is proved as follows:

$$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(f(\overline{x})) \Leftrightarrow \sigma(g(\overline{x})) \vdash P \Leftrightarrow Q$$

$$\Leftarrow \text{[using the equivalence]}$$

$$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(f(\overline{x})) \Leftrightarrow \sigma(g(\overline{x})) \vdash P' \Leftrightarrow Q$$

$$\Leftarrow \text{ [by weakening]}$$

$$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$$

which is exactly the induction hypothesis.
If we apply a rule $l \rightarrow r$ $\mathsf{if}$ $c$ in reverse way, we have

$$P = P[\sigma(r)]_{\omega} \xleftarrow[\mathcal{R}]{Th_{\approx},\Gamma} P[\sigma(l)]_{\omega} = P'$$

for some occurrence $\omega$ in $P'$ such that $Prot(P, \omega, \approx)$ is false and some substitution $\sigma$ such that $\sigma(l) = P'_{|\omega}$ and $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c)$
By induction hypothesis we get $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$.
Let us now derive a proof of $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$:

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by contraction and instantiation]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(c(\overline{x})))\Rightarrow(\sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x}))) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by implication left and weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x})))$ and $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x}))$ is true by hypothesis.
$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$ is proved as follows:

$$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$$

$$\Leftarrow \text{[using the equivalence]}$$

$$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x})) \vdash P' \Leftrightarrow Q$$

$$\Leftarrow \text{ [by weakening]}$$

$$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$$

which is exactly the induction hypothesis.
If we apply a rule $l \rightarrow r$ $\mathsf{if}$ $c$ in reverse way, we have

$$P = P[\sigma(r)]_{\omega} \xleftarrow[\mathcal{R}]{Th_{\approx},\Gamma} P[\sigma(l)]_{\omega} = P'$$

for some occurrence $\omega$ in $P'$ such that $Prot(P, \omega, \approx)$ is false and some substitution $\sigma$ such that $\sigma(l) = P'_{|\omega}$ and $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c)$

By induction hypothesis we get $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$.
Let us now derive a proof of $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$:

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by contraction and instantiation]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(c(\overline{x})))\Rightarrow(\sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x}))) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [by implication left and weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x})))$ and $\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash \sigma(c(\overline{x}))$ is true by hypothesis.
$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$ is proved as follows:

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x})) \vdash P \Leftrightarrow Q$

$\Leftarrow$ [using the equivalence]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma, \sigma(l(\overline{x})) \Leftrightarrow \sigma(r(\overline{x})) \vdash P' \Leftrightarrow Q$

$\Leftarrow$ [by weakening]

$\mathcal{T}_{\mathcal{RE}}, Th_{\approx}, \Gamma \vdash P' \Leftrightarrow Q$

which is exactly the induction hypothesis.

(B) For every proposition $P$ in $\mathcal{T}_{\mathcal{RE}}$, we have $Th_{\approx} \vdash_{\sim} P$.
There are two kinds of propositions in $\mathcal{T}_{\mathcal{RE}}$.

- If $P$ has the form $\forall\overline{x}(c \Rightarrow (p \Leftrightarrow q))$, we have to prove that we have $Th_{\approx} \vdash_{\sim} \forall\overline{x}(c \Rightarrow (p \Leftrightarrow q))$.

  To achieve this, we apply the corresponding axiom or rule in $\mathcal{R}$ $\mathcal{E}$, which is written as follows in sequent calculus modulo:

  $Th_{\approx} \vdash_{\sim} \forall\overline{x}(c \Rightarrow (p \Leftrightarrow q))$

  $\Leftarrow$ [by weakening]

  $\vdash_{\sim} \forall\overline{x}(c \Rightarrow (p \Leftrightarrow q))$

  $\Leftarrow$ [by freeing and implication right]

  $c(\overline{y}) \vdash_{\sim} p(\overline{y}) \Leftrightarrow q(\overline{y})$

  $\Leftarrow$ [by and right and twice implication right]

  $c(\overline{y}), p(\overline{y}) \vdash_{\sim} q(\overline{y})$ and $c(\overline{y}), q(\overline{y}) \vdash_{\sim} p(\overline{y})$

  Which are true by application of Axiom modulo.

- If $P$ has the form $\forall\overline{x}(c \Rightarrow (p \approx q))$, we have to prove that we have $Th_{\approx} \vdash_{\sim} \forall\overline{x}(c \Rightarrow (p \approx q))$.

  To achieve this, we again apply the corresponding axiom or rule in $\mathcal{R}$ $\mathcal{E}$, which is done using $Th_{\approx}$ and written as follows in sequent calculus modulo:

$$
\begin{aligned}
& Th_{\approx} \vdash_{\sim} \forall \overline{x}(c \Rightarrow (g \approx d)) \\
& \Leftarrow \text{ [by freeing and implication right]} \\
& Th_{\approx}, c(\overline{y}) \vdash_{\sim} g(\overline{y}) \approx d(\overline{y}) \\
& \Leftarrow \text{ [by weakening]} \\
& \forall x\,(x \approx x), c(\overline{y}) \vdash_{\sim} g(\overline{y}) \approx d(\overline{y}) \\
& \Leftarrow \text{ [by instantiation]} \\
& g(\overline{y}) \approx g(\overline{y}), c(\overline{y}) \vdash_{\sim} g(\overline{y}) \approx d(\overline{y})
\end{aligned}
$$

Which is true by application of Axiom modulo.

□

## C Proof of lemma 4.2

**Lemma 4.2** Given a user theory $Th_u$ and a Noetherian ordering $\prec$ on a set $\tau_x$, the relations described in figure 5 hold.

### Proof of "(3) if (4)"

In $HOL_{\lambda\sigma}$ we have:

$$
\begin{aligned}
& \forall R \forall \tau\,(Noeth(R,\tau) \Rightarrow \forall P\,(NoethInd(P,R,\tau))), Th_u \\
& \vdash^{\lambda\sigma} \forall x\, \forall y\,(x \in \tau_x \Rightarrow (y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))) \qquad (3)\\
& \Longleftrightarrow \\
& \forall R \forall \tau\,(Noeth(R,\tau) \Rightarrow \forall P\,(NoethInd(P,R,\tau))), Th_u \\
& \vdash^{\lambda\sigma} \forall x\,(x \in \tau_x \Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))) \\
& \Longleftarrow \text{[by instantiation]} \\
& \forall R \forall \tau\,(Noeth(R,\tau) \Rightarrow \forall P\,(NoethInd(P,R,\tau))), Th_u \\
& \vdash^{\lambda\sigma} X \in \tau_x \Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \\
& \Longleftarrow \text{[by } \textit{implication right}\text{]} \\
& \forall R \forall \tau\,(Noeth(R,\tau) \Rightarrow \forall P\,(NoethInd(P,R,\tau))), Th_u, X \in \tau_x \\
& \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \qquad (4)
\end{aligned}
$$

## Proof of "(4) if (3)"

In $HOL_{\lambda\sigma}$ we have:

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x\\ &\vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))\end{aligned} \tag{4}$$

$\Longleftarrow$ [using *cut*]

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x\\ &\vdash^{\lambda\sigma} \forall x\,(x \in \tau_x \Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))), \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))\end{aligned} \tag{C.1}$$

and

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x,\\ &\forall x\,(x \in \tau_x \Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))) \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))\end{aligned} \tag{C.2}$$

We easily prove:

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x,\\ &\forall x\,(x \in \tau_x \Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))) \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))\end{aligned} \tag{$C$.2}$$

$\Longleftarrow$ [by instantiation]

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x,\\ &X \in \tau_x \Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))\end{aligned}$$

$\Longleftarrow$ [by *implication left*]

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x\\ &\vdash^{\lambda\sigma} X \in \tau_x, \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))\end{aligned}$$

and

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x,\\ &\forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))\end{aligned}$$

So we continue with:

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x\\ &\vdash^{\lambda\sigma} \forall x\,(x \in \tau_x \Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))), \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))\end{aligned} \tag{$C$.1}$$

$\Longleftarrow$ [by weakening]

$$\begin{aligned}&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x\\ &\vdash^{\lambda\sigma} \forall x\,(x \in \tau_x \Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y))))\end{aligned}$$

$\Longleftrightarrow$

$$\begin{array}{l} \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u \\ \vdash^{\lambda\sigma} \forall x\, \forall y\,(x \in \tau_x \Rightarrow (y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))) \end{array} \tag{3}$$

## Proof of "(4) if (5)"

In $HOL_{\lambda\sigma}$ we have, expliciting the application of the induction principle, and putting the obtained induction hypothesis into form:

$$\begin{array}{l} \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x \\ \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \end{array} \tag{4}$$

$$\Longleftarrow \left[\begin{array}{l} \text{by contraction and instantiation,} \\ \text{the induction relation and set are chosen} \end{array}\right]$$

$$\begin{array}{l} \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x, \\ \mathit{Noeth}(\prec, \tau_x) \Rightarrow \forall P\,(\mathit{NoethInd}(P,\prec,\tau_x)) \\ \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \end{array}$$

$\Longleftarrow$ [by *implication left*]

$$\begin{array}{l} \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x \\ \vdash^{\lambda\sigma} \mathit{Noeth}(\prec,\tau_x), \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \end{array} \tag{C.3}$$

and

$$\begin{array}{l} \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x, \forall P\,(\mathit{NoethInd}(P,\prec,\tau_x)) \\ \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \end{array} \tag{C.4}$$

Since we suppose $\prec$ noetherian on $\tau_x$, we indeed have:

$$\begin{array}{l} \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x \\ \vdash^{\lambda\sigma} \mathit{Noeth}(\prec,\tau_x), \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \end{array} \quad (C.3)$$

So we continue with:

$$\begin{array}{l} \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x, \forall P\,(\mathit{NoethInd}(P,\prec,\tau_x)) \\ \vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \end{array} \quad (C.4)$$

$\Longleftarrow$ [by instantiation and renaming of bound (quantified) variables]

$\forall R \forall \tau\, (Noeth(R, \tau) \Rightarrow \forall P\, (NoethInd(P, R, \tau))), Th_u, X \in \tau_x,$
$\forall x\, ((x \in \tau_x \wedge \forall \underline{x}\, ((\underline{x} \in \tau_x \wedge \underline{x} \prec x) \Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(\underline{x}, y) \Rightarrow Q(\underline{x}, y)))))$
$\Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(x, y) \Rightarrow Q(x, y))))$
$\Rightarrow \forall x\, (x \in \tau_x \Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(x, y) \Rightarrow Q(x, y))))$
$\vdash^{\lambda\sigma} \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$
$\Longleftrightarrow$
$\forall R \forall \tau\, (Noeth(R, \tau) \Rightarrow \forall P\, (NoethInd(P, R, \tau))), Th_u, X \in \tau_x,$
$\forall x\, ((x \in \tau_x \wedge \forall \underline{x} \forall y\, ((\underline{x} \in \tau_x \wedge \underline{x} \prec x \wedge y \in \tau_y \wedge C(\underline{x}, y)) \Rightarrow Q(\underline{x}, y)))$
$\Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(x, y) \Rightarrow Q(x, y))))$
$\Rightarrow \forall x\, (x \in \tau_x \Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(x, y) \Rightarrow Q(x, y))))$
$\vdash^{\lambda\sigma} \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$
$\Longleftarrow$ [by *implication left*]
$\forall R \forall \tau\, (Noeth(R, \tau) \Rightarrow \forall P\, (NoethInd(P, R, \tau))), Th_u, X \in \tau_x$
$\vdash^{\lambda\sigma} \forall x\, ((x \in \tau_x \wedge \forall \underline{x} \forall y\, ((\underline{x} \in \tau_x \wedge \underline{x} \prec x \wedge y \in \tau_y \wedge C(\underline{x}, y)) \Rightarrow Q(\underline{x}, y)))$ (C.5)
$\Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(x, y) \Rightarrow Q(x, y)))), \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$
and
$\forall R \forall \tau\, (Noeth(R, \tau) \Rightarrow \forall P\, (NoethInd(P, R, \tau))), Th_u, X \in \tau_x,$
$\forall x\, (x \in \tau_x \Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(x, y) \Rightarrow Q(x, y))))$ (C.6)
$\vdash^{\lambda\sigma} \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$

We easily prove

$\forall R \forall \tau\, (Noeth(R, \tau) \Rightarrow \forall P\, (NoethInd(P, R, \tau))), Th_u, X \in \tau_x,$
$\forall x\, (x \in \tau_x \Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(x, y) \Rightarrow Q(x, y))))$ $(C.6)$
$\vdash^{\lambda\sigma} \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$
$\Longleftarrow$ [by weakening]
$X \in \tau_x, \forall x\, (x \in \tau_x \Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(x, y) \Rightarrow Q(x, y))))$
$\vdash^{\lambda\sigma} \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$
$\Longleftarrow$ [by instantiation]
$X \in \tau_x, X \in \tau_x \Rightarrow \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$
$\vdash^{\lambda\sigma} \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$
$\Longleftarrow$ [by *implication left*]
$X \in \tau_x \vdash^{\lambda\sigma} X \in \tau_x, \forall y\, (y \in \tau_y \Rightarrow (C(X, y) \Rightarrow Q(X, y)))$

et

$$X \in \tau_x, \forall y\, (y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))$$
$$\vdash^{\lambda\sigma} \forall y\, (y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))$$

So we continue with:

$$\begin{aligned}
&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x \\
&\vdash^{\lambda\sigma} \forall x\,((x \in \tau_x \wedge \forall \underline{x} \forall y\,((\underline{x} \in \tau_x \wedge \underline{x} \prec x \wedge y \in \tau_y \wedge C(\underline{x},y)) \Rightarrow Q(\underline{x},y))) \qquad (C.5) \\
&\Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))), \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \\
&\Longleftarrow [\text{by weakening}] \\
&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u \\
&\vdash^{\lambda\sigma} \forall x\,((x \in \tau_x \wedge \forall \underline{x} \forall y\,((\underline{x} \in \tau_x \wedge \underline{x} \prec x \wedge y \in \tau_y \wedge C(\underline{x},y)) \Rightarrow Q(\underline{x},y))) \\
&\Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(x,y) \Rightarrow Q(x,y)))) \\
&\Longleftarrow [\text{by instantiation}] \\
&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u \\
&\vdash^{\lambda\sigma} (X \in \tau_x \wedge \forall \underline{x} \forall y\,((\underline{x} \in \tau_x \wedge \underline{x} \prec x \wedge y \in \tau_y \wedge C(\underline{x},y)) \Rightarrow Q(\underline{x},y))) \\
&\Rightarrow \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \\
&\Longleftarrow [\text{by } \textit{implication right}] \\
&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \\
&(X \in \tau_x \wedge \forall \underline{x} \forall y\,((\underline{x} \in \tau_x \wedge \underline{x} \prec x \wedge y \in \tau_y \wedge C(\underline{x},y)) \Rightarrow Q(\underline{x},y))) \\
&\vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \\
&\Longleftarrow [\text{by } \textit{and left}] \\
&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x, \\
&\forall \underline{x} \forall y\,((\underline{x} \in \tau_x \wedge \underline{x} \prec x \wedge y \in \tau_y \wedge C(\underline{x},y)) \Rightarrow Q(\underline{x},y)) \qquad (5) \\
&\vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))
\end{aligned}$$

## Proof of “(5) if (4)”

In $HOL_{\lambda\sigma}$ we have:

$$\begin{aligned}
&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x, \\
&\forall \underline{x} \forall y\,((\underline{x} \in \tau_x \wedge \underline{x} \prec x \wedge y \in \tau_y \wedge C(\underline{x},y)) \Rightarrow Q(\underline{x},y)) \qquad (5) \\
&\vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y))) \\
&\Longleftarrow [\text{by weakening}] \\
&\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, X \in \tau_x \qquad (4) \\
&\vdash^{\lambda\sigma} \forall y\,(y \in \tau_y \Rightarrow (C(X,y) \Rightarrow Q(X,y)))
\end{aligned}$$

**Remark C.1** *Notice that in the proof above, we make use of the higher-order capabilities of $HOL_{\lambda\sigma}$ to instanciate $P$ and $R$.*

## D Proof of lemma 4.3

**Lemma 4.3** Given a user theory $Th_u$ and a Noetherian ordering $\prec$ on a set $\tau_1$, and given a subset $\tau_2$ of $\tau_1$, the relations described in figure 6 hold.

### Proof of "(7) if (8)"

In $HOL_{\lambda\sigma}$ we have:

$$\begin{array}{l}\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1)\\ \vdash^{\lambda\sigma} \forall x\,(x \in \tau_1 \Rightarrow (C_1(x) \Rightarrow Q_1(x))) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array} \quad (7)$$

$$\begin{array}{l}\Longleftrightarrow\\ \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1)\\ \vdash^{\lambda\sigma} \forall x\,(x \in \tau_1 \Rightarrow ((C_1(x) \Rightarrow Q_1(x)) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))))\\ \Longleftarrow \text{[by instantiation]}\\ \forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1)\\ \vdash^{\lambda\sigma} X \in \tau_1 \Rightarrow ((C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))\\ \Longleftarrow \text{[by } \textit{implication right}\text{]}\end{array}$$

$$\begin{array}{l}\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1\\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array} \quad (8)$$

### Proof of "(7) if (8)"

In $HOL_{\lambda\sigma}$ we have:

$$\begin{array}{l}\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1\\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array} \quad (8)$$

$$\Longleftarrow \text{[using } \textit{cut}\text{]}$$

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1 \\ \vdash^{\lambda\sigma} \forall x\, (x \in \tau_1 \Rightarrow ((C_1(x) \Rightarrow Q_1(x)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))), \\ (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array} \tag{D.1}$$

and

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\ \forall x\, (x \in \tau_1 \Rightarrow ((C_1(x) \Rightarrow Q_1(x)) \wedge \forall x\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))) \\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array} \tag{D.2}$$

We easily prove:

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\ \forall x\, (x \in \tau_1 \Rightarrow ((C_1(x) \Rightarrow Q_1(x)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))) \\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array} \tag*{$(D.2)$}$$

$\Longleftarrow$ [by instantiation]

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\ X \in \tau_1 \Rightarrow ((C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))) \\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array}$$

$\Longleftarrow$ [by *implication left*]

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\ \vdash^{\lambda\sigma} X \in \tau_1, (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array}$$

and

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\ (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))) \\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array}$$

So we continue with:

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1 \\ \vdash^{\lambda\sigma} \forall x\, (x \in \tau_1 \Rightarrow ((C_1(x) \Rightarrow Q_1(x)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))), \\ (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array} \tag{D.1}$$

$\Longleftarrow$ [by weakening]

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1) \\ \vdash^{\lambda\sigma} \forall x\, (x \in \tau_1 \Rightarrow ((C_1(x) \Rightarrow Q_1(x)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))))\end{array}$$

$\Longleftrightarrow$

$$\begin{aligned}&\forall R\forall\tau\,(\mathit{Noeth}(R,\tau)\Rightarrow\forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x\in\tau_2\Rightarrow x\in\tau_1)\\&\vdash^{\lambda\sigma}\forall x\,(x\in\tau_1\Rightarrow(C_1(x)\Rightarrow Q_1(x)))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\end{aligned}\tag{7}$$

## Proof of "(8) if (9)"

In $HOL_{\lambda\sigma}$ we have:

$$\begin{aligned}&\forall R\forall\tau\,(\mathit{Noeth}(R,\tau)\Rightarrow\forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x\in\tau_2\Rightarrow x\in\tau_1), X\in\tau_1\\&\vdash^{\lambda\sigma}(C_1(X)\Rightarrow Q_1(X))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\end{aligned}\tag{8}$$

$\Longleftarrow$

$$\begin{aligned}&\forall R\forall\tau\,(\mathit{Noeth}(R,\tau)\Rightarrow\forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x\in\tau_2\Rightarrow x\in\tau_1), X\in\tau_1,\\&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\&\vdash^{\lambda\sigma}(C_1(X)\Rightarrow Q_1(X))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\end{aligned}$$

$\Longleftarrow$ [using *cut*]

$$\begin{aligned}&\forall R\forall\tau\,(\mathit{Noeth}(R,\tau)\Rightarrow\forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x\in\tau_2\Rightarrow x\in\tau_1), X\in\tau_1,\\&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\&\vdash^{\lambda\sigma}\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow(C_1(\underline{x})\Rightarrow Q_1(\underline{x}))),\\&(C_1(X)\Rightarrow Q_1(X))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\end{aligned}\tag{D.3}$$

and

$$\begin{aligned}&\forall R\forall\tau\,(\mathit{Noeth}(R,\tau)\Rightarrow\forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x\in\tau_2\Rightarrow x\in\tau_1), X\in\tau_1,\\&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow(C_1(\underline{x})\Rightarrow Q_1(\underline{x}))),\\&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\&\vdash^{\lambda\sigma}(C_1(X)\Rightarrow Q_1(X))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\end{aligned}\tag{D.4}$$

We easily prove:

$$\begin{aligned}&\forall R\forall\tau\,(\mathit{Noeth}(R,\tau)\Rightarrow\forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x\in\tau_2\Rightarrow x\in\tau_1), X\in\tau_1,\\&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\&\vdash^{\lambda\sigma}\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow(C_1(\underline{x})\Rightarrow Q_1(\underline{x}))),\\&(C_1(X)\Rightarrow Q_1(X))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\end{aligned}\tag{D.3}$$

$\Longleftarrow$ [by weakening]

$$\begin{aligned}&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\&\vdash^{\lambda\sigma}\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow(C_1(\underline{x})\Rightarrow Q_1(\underline{x})))\end{aligned}$$

$\Longleftarrow$ [by instantiation]

$$\forall \underline{x}\,((\underline{x} \in \tau_1 \wedge \underline{x} \prec X) \Rightarrow ((C_1(\underline{x}) \Rightarrow Q_1(\underline{x})) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))))$$

$$\vdash^{\lambda\sigma} (\underline{X} \in \tau_1 \wedge \underline{X} \prec X) \Rightarrow (C_1(\underline{X}) \Rightarrow Q_1(\underline{X}))$$

$\Longleftarrow$ [by instantiation]

$$(\underline{X} \in \tau_1 \wedge \underline{X} \prec X) \Rightarrow ((C_1(\underline{X}) \Rightarrow Q_1(\underline{X})) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))$$

$$\vdash^{\lambda\sigma} (\underline{X} \in \tau_1 \wedge \underline{X} \prec X) \Rightarrow (C_1(\underline{X}) \Rightarrow Q_1(\underline{X}))$$

$\Longleftarrow$ [by *implication right*]

$$(\underline{X} \in \tau_1 \wedge \underline{X} \prec X) \Rightarrow ((C_1(\underline{X}) \Rightarrow Q_1(\underline{X})) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))),$$

$$\underline{X} \in \tau_1 \wedge \underline{X} \prec X \vdash^{\lambda\sigma} C_1(\underline{X}) \Rightarrow Q_1(\underline{X})$$

$\Longleftarrow$ [by *implication left*]

$$\underline{X} \in \tau_1 \wedge \underline{X} \prec X \vdash^{\lambda\sigma} \underline{X} \in \tau_1 \wedge \underline{X} \prec X, C_1(\underline{X}) \Rightarrow Q_1(\underline{X})$$

and

$$C_1(\underline{X}) \Rightarrow Q_1(\underline{X}), \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))), \underline{X} \in \tau_1 \wedge \underline{X} \prec X \vdash^{\lambda\sigma} C_1(\underline{X}) \Rightarrow Q_1(\underline{X})$$

So we continue with:

$$\begin{gathered}\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1,\\ \forall \underline{x}\,((\underline{x} \in \tau_1 \wedge \underline{x} \prec X) \Rightarrow (C_1(\underline{x}) \Rightarrow Q_1(\underline{x}))),\\ \forall \underline{x}\,((\underline{x} \in \tau_1 \wedge \underline{x} \prec X) \Rightarrow ((C_1(\underline{x}) \Rightarrow Q_1(\underline{x})) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))))\\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{gathered} \tag{$D$.4}$$

$\Longleftrightarrow$

$$\begin{gathered}\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1,\\ \forall \underline{x}\,((\underline{x} \in \tau_1 \wedge \underline{x} \prec X \wedge C_1(\underline{x})) \Rightarrow Q_1(\underline{x})),\\ \forall \underline{x}\,((\underline{x} \in \tau_1 \wedge \underline{x} \prec X) \Rightarrow ((C_1(\underline{x}) \Rightarrow Q_1(\underline{x})) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))))\\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{gathered}$$

$\Longleftarrow$ [using *cut*]

$$\begin{gathered}\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1,\\ \forall \underline{x}\,((\underline{x} \in \tau_1 \wedge \underline{x} \prec X \wedge C_1(\underline{x})) \Rightarrow Q_1(\underline{x})),\\ \forall \underline{x}\,((\underline{x} \in \tau_1 \wedge \underline{x} \prec X) \Rightarrow ((C_1(\underline{x}) \Rightarrow Q_1(\underline{x})) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))))\\ \vdash^{\lambda\sigma} \forall \underline{y}\,((\underline{y} \in \tau_1 \wedge \underline{y} \prec X) \Rightarrow \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))),\\ (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{gathered} \tag{D.5}$$

and

$$\begin{aligned}
&\forall R\forall\tau\,(\mathit{Noeth}(R,\tau)\Rightarrow\forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x\in\tau_2\Rightarrow x\in\tau_1), X\in\tau_1,\\
&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X\wedge C_1(\underline{x}))\Rightarrow Q_1(\underline{x})),\\
&\forall\underline{y}\,((\underline{y}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))),\\
&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\
&\vdash^{\lambda\sigma}(C_1(X)\Rightarrow Q_1(X))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))
\end{aligned}\tag{D.6}$$

We easily prove:

$$\begin{aligned}
&\forall R\forall\tau\,(\mathit{Noeth}(R,\tau)\Rightarrow\forall P\,(\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\,(x\in\tau_2\Rightarrow x\in\tau_1), X\in\tau_1,\\
&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X\wedge C_1(\underline{x}))\Rightarrow Q_1(\underline{x})),\\
&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\
&\vdash^{\lambda\sigma}\forall\underline{y}\,((\underline{y}\in\tau_1\wedge\underline{y}\prec X)\Rightarrow\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))),\\
&(C_1(X)\Rightarrow Q_1(X))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))
\end{aligned}\tag{D.5}$$

$$\begin{aligned}
&\Longleftarrow\text{[by weakening]}\\
&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\
&\vdash^{\lambda\sigma}\forall\underline{y}\,((\underline{y}\in\tau_1\wedge\underline{y}\prec X)\Rightarrow\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y))))\\
&\Longleftarrow\text{[by instantiation]}\\
&\forall\underline{x}\,((\underline{x}\in\tau_1\wedge\underline{x}\prec X)\Rightarrow((C_1(\underline{x})\Rightarrow Q_1(\underline{x}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))))\\
&\vdash^{\lambda\sigma}(\underline{Y}\in\tau_1\wedge\underline{Y}\prec X)\Rightarrow\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\\
&\Longleftarrow\text{[by instantiation]}\\
&(\underline{Y}\in\tau_1\wedge\underline{Y}\prec X)\Rightarrow((C_1(\underline{Y})\Rightarrow Q_1(\underline{Y}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y))))\\
&\vdash^{\lambda\sigma}(\underline{Y}\in\tau_1\wedge\underline{Y}\prec X)\Rightarrow\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\\
&\Longleftarrow\text{[by \textit{implication right}]}\\
&(\underline{Y}\in\tau_1\wedge\underline{Y}\prec X)\Rightarrow((C_1(\underline{Y})\Rightarrow Q_1(\underline{Y}))\wedge\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))),\\
&\underline{Y}\in\tau_1\wedge\underline{Y}\prec X\vdash^{\lambda\sigma}\forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\\
&\Longleftarrow\text{[by \textit{implication left}]}\\
&\underline{Y}\in\tau_1\wedge\underline{Y}\prec X\\
&\vdash^{\lambda\sigma}\underline{Y}\in\tau_1\wedge\underline{Y}\prec X, \forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))\\
&\text{and}\\
&C_1(\underline{Y})\Rightarrow Q_1(\underline{Y}), \forall y\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y))), \underline{Y}\in\tau_1\wedge\underline{X}\prec Y\\
&\vdash^{\lambda\sigma}\forall x\,(y\in\tau_2\Rightarrow(C_2(y)\Rightarrow Q_2(y)))
\end{aligned}$$

So we continue with:

$$\begin{gathered}
\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), \mathit{Th}_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\
\forall \underline{x}\,((\underline{x} \in \tau_1 \land \underline{x} \prec X \land C_1(\underline{x})) \Rightarrow Q_1(\underline{x})), \\
\forall \underline{y}\,((\underline{y} \in \tau_1 \land \underline{y} \prec X) \Rightarrow \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))), \\
\forall \underline{x}\,((\underline{x} \in \tau_1 \land \underline{x} \prec X) \Rightarrow ((C_1(\underline{x}) \Rightarrow Q_1(\underline{x})) \land \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))) \\
\vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \land \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))
\end{gathered} \tag{D.6}$$

$$\Longleftrightarrow$$

$$\begin{gathered}
\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), \mathit{Th}_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\
\forall \underline{x}\,((\underline{x} \in \tau_1 \land \underline{x} \prec X \land C_1(\underline{x})) \Rightarrow Q_1(\underline{x})), \\
\forall \underline{y} \forall y\,((\underline{y} \in \tau_1 \land \underline{y} \prec X \land y \in \tau_2 \land C_2(y)) \Rightarrow Q_2(y)), \\
\forall \underline{x}\,((\underline{x} \in \tau_1 \land \underline{x} \prec X) \Rightarrow ((C_1(\underline{x}) \Rightarrow Q_1(\underline{x})) \land \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))) \\
\vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \land \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))
\end{gathered}$$

$$\Longleftarrow$$

$$\begin{gathered}
\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), \mathit{Th}_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\
\forall \underline{x}\,((\underline{x} \in \tau_1 \land \underline{x} \prec X \land C_1(\underline{x})) \Rightarrow Q_1(\underline{x})), \forall \underline{y}\,((\underline{y} \in \tau_2 \land \underline{y} \prec X \land C_2(\underline{y})) \Rightarrow Q_2(\underline{y})), \\
\forall \underline{x}\,((\underline{x} \in \tau_1 \land \underline{x} \prec X) \Rightarrow ((C_1(\underline{x}) \Rightarrow Q_1(\underline{x})) \land \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y))))) \\
\vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \land \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))
\end{gathered}$$

$$\Longleftarrow \text{[by weakening]}$$

$$\begin{gathered}
\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), \mathit{Th}_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\
\forall \underline{x}\,((\underline{x} \in \tau_1 \land \underline{x} \prec X \land C_1(\underline{x})) \Rightarrow Q_1(\underline{x})), \forall \underline{y}\,((\underline{y} \in \tau_2 \land \underline{y} \prec X \land C_2(\underline{y})) \Rightarrow Q_2(\underline{y})) \\
\vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \land \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))
\end{gathered} \tag{9}$$

## Proof of "(9) if (8)"

In $HOL_{\lambda\sigma}$ we have:

$$\begin{gathered}
\forall R \forall \tau\,(\mathit{Noeth}(R,\tau) \Rightarrow \forall P\,(\mathit{NoethInd}(P,R,\tau))), \mathit{Th}_u, \forall x\,(x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\
\forall \underline{x}\,((\underline{x} \in \tau_1 \land \underline{x} \prec X \land C_1(\underline{x})) \Rightarrow Q_1(\underline{x})), \forall \underline{y}\,((\underline{y} \in \tau_2 \land \underline{y} \prec X \land C_2(\underline{y})) \Rightarrow Q_2(\underline{y})) \\
\vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \land \forall y\,(y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))
\end{gathered} \tag{9}$$

$$\Longleftarrow \text{[by weakening]}$$

$$\begin{array}{l}\forall R \forall \tau\, (\mathit{Noeth}(R,\tau) \Rightarrow \forall P\, (\mathit{NoethInd}(P,R,\tau))), Th_u, \forall x\, (x \in \tau_2 \Rightarrow x \in \tau_1), X \in \tau_1, \\ \vdash^{\lambda\sigma} (C_1(X) \Rightarrow Q_1(X)) \wedge \forall y\, (y \in \tau_2 \Rightarrow (C_2(y) \Rightarrow Q_2(y)))\end{array} \tag{8}$$

## E Proof of lemma 4.9

We give here several alternate definitions for $<_e$, which we use to prove the main results.

We prove that they are equivalent to definition 4.1.

In all these definitions, let $<$ be a noetherian ordering on terms.

**Definition E.1**

$$\begin{array}{ll} & t_1 \approx t_2 >_e t_3 \approx t_4 \\ \Leftrightarrow & \\ & (t_1 > t_3 \vee t_2 > t_3) \wedge (t_1 > t_4 \vee t_2 > t_4) \\ \vee & t_1 \neq t_2 \wedge \bigvee \left\{ \begin{array}{l} t_1 = t_3 \wedge (t_2 > t_4 \vee t_3 = t_4 \vee (t_1 > t_4 \wedge t_1 \# t_2)) \\ t_2 = t_3 \wedge (t_1 > t_4 \vee t_3 = t_4 \vee (t_2 > t_4 \wedge t_1 \# t_2)) \\ t_1 = t_4 \wedge (t_2 > t_3 \vee t_3 = t_4 \vee (t_1 > t_3 \wedge t_1 \# t_2)) \\ t_2 = t_4 \wedge (t_1 > t_3 \vee t_3 = t_4 \vee (t_2 > t_3 \wedge t_1 \# t_2)) \end{array} \right. \end{array}$$

**Definition E.2**

$$\begin{array}{ll}
 & t_1 \approx t_2 >_e t_3 \approx t_4 \\
\Leftrightarrow & \\
 & ( \\
\hline
 & t_1 > t_2 \wedge t_3 > t_4 \ \wedge (t_1 > t_3 \vee (t_1 = t_3 \wedge t_2 > t_4)) \\
\vee & t_1 > t_2 \wedge t_3 < t_4 \ \wedge (t_1 > t_4 \vee (t_1 = t_4 \wedge t_2 > t_3)) \\
\vee & t_1 > t_2 \wedge t_3 = t_4 \ \wedge t_1 \geq t_4^3 \\
\vee & t_1 > t_2 \wedge t_3 \# t_4 \ \wedge t_1 > t_3 \wedge t_1 > t_4 \\
\hline
\vee & t_1 < t_2 \wedge t_3 > t_4 \ \wedge (t_2 > t_3 \vee (t_2 = t_3 \wedge t_1 > t_4)) \\
\vee & t_1 < t_2 \wedge t_3 < t_4 \ \wedge (t_2 > t_4 \vee (t_2 = t_4 \wedge t_1 > t_3)) \\
\vee & t_1 < t_2 \wedge t_3 = t_4 \ \wedge t_2 \geq t_4^3 \\
\vee & t_1 < t_2 \wedge t_3 \# t_4 \ \wedge t_2 > t_3 \wedge t_2 > t_4 \\
\hline
\vee & t_1 = t_2 \wedge t_3 > t_4 \ \wedge t_2^1 > t_3 \\
\vee & t_1 = t_2 \wedge t_3 < t_4 \ \wedge t_2^1 > t_4 \\
\vee & t_1 = t_2 \wedge t_3 = t_4 \ \wedge t_2^1 > t_4^3 \\
\vee & t_1 = t_2 \wedge t_3 \# t_4 \ \wedge t_2^1 > t_3 \wedge t_2^1 > t_4 \\
\hline
\vee & t_1 \# t_2 \wedge t_3 > t_4 \ \wedge (t_1 \geq t_3 \vee t_2 \geq t_3) \\
\vee & t_1 \# t_2 \wedge t_3 < t_4 \ \wedge (t_1 \geq t_4 \vee t_2 \geq t_4) \\
\vee & t_1 \# t_2 \wedge t_3 = t_4 \ \wedge (t_1 \geq t_4^3 \vee t_2 \geq t_4^3) \\
\vee & t_1 \# t_2 \wedge t_3 \# t_4 \ \wedge (t_1 > t_3 \vee t_2 > t_3 \vee t_1 > t_4 \vee t_2 > t_4) \\
 & \qquad\qquad\qquad\quad \wedge (t_1 \geq t_3 \vee t_2 \geq t_3) \wedge (t_1 \geq t_4 \vee t_2 \geq t_4) \\
 & )
\end{array}$$

We start by giving a simplified form of $\gg$, in the case of two incomparable terms, since it is the most difficult case we get with $<_e$.

**Lemma E.1**

$$\begin{array}{ll}
 & t_1 \# t_2 \wedge t_3 \# t_4 \\
\Rightarrow & (\{t_1, t_2\} \gg \{t_3, t_4\} \\
\Leftrightarrow & ((t_1 > t_3 \vee t_2 > t_3 \vee t_1 > t_4 \vee t_2 > t_4) \\
\wedge & (t_1 \geq t_3 \vee t_2 \geq t_3) \\
\wedge & (t_1 \geq t_4 \vee t_2 \geq t_4)))
\end{array}$$

**Proof:** By definition of $\gg$, we have:

$$\begin{aligned}
& t_1 \# t_2 \wedge t_3 \# t_4 \\
\Rightarrow\ & (\{t_1, t_2\} \gg \{t_3, t_4\} \\
\Leftrightarrow\ & ((t_1 > t_3 \vee t_2 > t_3) \wedge (t_1 > t_4 \vee t_2 > t_4)) \\
\vee\ & (t_1 = t_3 \wedge t_2 > t_4) \\
\vee\ & (t_2 = t_3 \wedge t_1 > t_4) \\
\vee\ & (t_1 = t_4 \wedge t_2 > t_3) \\
\vee\ & (t_2 = t_4 \wedge t_1 > t_3))
\end{aligned}$$

Noticing that they do not introduce any inconsistency, we add the cases we need to recombine:

$$\begin{aligned}
& t_1 \# t_2 \wedge t_3 \# t_4 \\
\Rightarrow\ & (\{t_1, t_2\} \gg \{t_3, t_4\} \\
\Leftrightarrow\ & ((t_1 > t_3 \vee t_2 > t_3) \wedge (t_1 > t_4 \vee t_2 > t_4)) \\
\vee\ & ((t_1 = t_3 \vee t_2 = t_3) \wedge t_2 > t_4) \\
\vee\ & ((t_1 = t_3 \vee t_2 = t_3) \wedge t_1 > t_4) \\
\vee\ & ((t_1 = t_4 \vee t_2 = t_4) \wedge t_2 > t_3) \\
\vee\ & ((t_1 = t_4 \vee t_2 = t_4) \wedge t_1 > t_3)
\end{aligned}$$

We can now recombine, to obtain:

$$\begin{aligned}
& t_1 \# t_2 \wedge t_3 \# t_4 \\
\Rightarrow\ & (\{t_1, t_2\} \gg \{t_3, t_4\} \\
\Leftrightarrow\ & (((t_1 = t_3 \vee t_2 = t_3) \wedge (t_1 > t_4 \vee t_2 > t_4)) \\
\vee\ & ((t_1 = t_4 \vee t_2 = t_4) \wedge (t_1 > t_3 \vee t_2 > t_3)) \\
\vee\ & ((t_1 > t_3 \vee t_2 > t_3) \wedge (t_1 > t_4 \vee t_2 > t_4))))
\end{aligned}$$

Still recombining, we get:

$$\begin{aligned}
& t_1 \# t_2 \wedge t_3 \# t_4 \\
\Rightarrow\ & (\{t_1, t_2\} \gg \{t_3, t_4\} \\
\Leftrightarrow\ & (((t_1 > t_4 \vee t_2 > t_4) \wedge (t_1 \geq t_3 \vee t_2 \geq t_3)) \\
\vee\ & ((t_1 > t_3 \vee t_2 > t_3) \wedge (t_1 \geq t_4 \vee t_2 \geq t_4))))
\end{aligned}$$

Now, as we want a conjunctive form, we expand to obtain:

$$\begin{aligned}
& t_1 \mathrel{\#} t_2 \wedge t_3 \mathrel{\#} t_4 \\
\Rightarrow\ & (\{t_1, t_2\} \gg \{t_3, t_4\} \\
\Leftrightarrow\ & ((t_1 > t_3 \vee t_2 > t_3 \vee t_1 > t_4 \vee t_2 > t_4) \\
\wedge\ & (t_1 > t_3 \vee t_2 > t_3 \vee t_1 \geq t_3 \vee t_2 \geq t_3) \\
\wedge\ & (t_1 > t_4 \vee t_2 > t_4 \vee t_1 \geq t_4 \vee t_2 \geq t_4) \\
\wedge\ & (t_1 \geq t_3 \vee t_2 \geq t_3 \vee t_1 \geq t_4 \vee t_2 \geq t_4)))
\end{aligned}$$

A last set of simplifications lets us obtain:

$$\begin{aligned}
& t_1 \mathrel{\#} t_2 \wedge t_3 \mathrel{\#} t_4 \\
\Rightarrow\ & (\{t_1, t_2\} \gg \{t_3, t_4\} \\
\Leftrightarrow\ & ((t_1 > t_3 \vee t_2 > t_3 \vee t_1 > t_4 \vee t_2 > t_4) \\
\wedge\ & (t_1 \geq t_3 \vee t_2 \geq t_3) \\
\wedge\ & (t_1 \geq t_4 \vee t_2 \geq t_4)))
\end{aligned}$$

□

**Lemma E.2** Definition E.2 is equivalent to definition 4.1.

**Proof:** The result easily comes by expanding $t_1 \approx t_2 >_e t_3 \approx t_4$ using definition 4.1. We proceed by cases:

- if $t_1 > t_2$, we have $C(t_1 \approx t_2) = (\{t_1\}, \{t_2\})$
  - · if $t_3 > t_4$, we have $C(t_3 \approx t_4) = (\{t_3\}, \{t_4\})$, and thus

    $$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_1 > t_3 \vee (t_1 = t_3 \wedge t_2 > t_4)$$

  - · if $t_3 < t_4$, we have $C(t_3 \approx t_4) = (\{t_4\}, \{t_3\})$, and thus

    $$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_1 > t_4 \vee (t_1 = t_4 \wedge t_2 > t_3)$$

  - · if $t_3 = t_4 = t_4^3$, we have $C(t_3 \approx t_4) = (\{t_4^3\}, \{\})$, and thus

    $$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_1 \geq t_4^3$$

  - · if $t_3 \mathrel{\#} t_4$, we have $C(t_3 \approx t_4) = (\{t_3, t_4\}, \{\})$, and thus

    $$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_1 > t_3 \wedge t_1 > t_4$$

- if $t_1 < t_2$, we have $C(t_1 \approx t_2) = (\{t_2\}, \{t_1\})$
  - · if $t_3 > t_4$, we have $C(t_3 \approx t_4) = (\{t_3\}, \{t_4\})$, and thus

    $$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_2 > t_3 \vee (t_2 = t_3 \wedge t_1 > t_4)$$

  - if $t_3 < t_4$, we have $C(t_3 \approx t_4) = (\{t_4\}, \{t_3\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_2 > t_4 \vee (t_2 = t_4 \wedge t_1 > t_3)$$

  - if $t_3 = t_4 = t_4^3$, we have $C(t_3 \approx t_4) = (\{t_4^3\}, \{\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_2 \geq t_4^3$$

  - if $t_3 \# t_4$, we have $C(t_3 \approx t_4) = (\{t_3, t_4\}, \{\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_2 > t_3 \wedge t_2 > t_4$$

- if $t_1 = t_2 = t_2^1$, we have $C(t_1 \approx t_2) = (\{t_2^1\}, \{\})$
  - if $t_3 > t_4$, we have $C(t_3 \approx t_4) = (\{t_3\}, \{t_4\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_2^1 > t_3$$

  - if $t_3 < t_4$, we have $C(t_3 \approx t_4) = (\{t_4\}, \{t_3\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_2^1 > t_4$$

  - if $t_3 = t_4 = t_4^3$, we have $C(t_3 \approx t_4) = (\{t_4^3\}, \{\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_2^1 > t_4^3$$

  - if $t_3 \# t_4$, we have $C(t_3 \approx t_4) = (\{t_3, t_4\}, \{\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_2^1 > t_3 \wedge t_2^1 > t_4$$

- if $t_1 \# t_2$, we have $C(t_1 \approx t_2) = (\{t_1, t_2\}, \{\})$
  - if $t_3 > t_4$, we have $C(t_3 \approx t_4) = (\{t_3\}, \{t_4\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_1 \geq t_3 \vee t_2 \geq t_3$$

  - if $t_3 < t_4$, we have $C(t_3 \approx t_4) = (\{t_4\}, \{t_3\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_1 \geq t_4 \vee t_2 \geq t_4$$

  - if $t_3 = t_4 = t_4^3$, we have $C(t_3 \approx t_4) = (\{t_4^3\}, \{\})$, and thus

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow t_1 \geq t_4^3 \vee t_2 \geq t_4^3$$

  - if $t_3 \# t_4$, we have $C(t_3 \approx t_4) = (\{t_3, t_4\}, \{\})$, and we are thus in the context of lemma E.1 and

$$t_1 \approx t_2 >_e t_3 \approx t_4 \Leftrightarrow \{t_1, t_2\} \gg \{t_3, t_4\}$$

We finish the proof by recollecting the results. □

**Lemma E.3** Definition E.1 is equivalent to definition E.2.

**Proof:** This is easily checked by simplifying definition E.2 for each case of the comparison of $t_1$ with $t_2$ and of $t_3$ with $t_4$. □

**Lemma E.4** Definition 4.2 is equivalent to definition E.1.

**Proof:** This is easily checked on definitions E.1 and 4.2. □

**Lemma E.5** When the equations have a common term, we get a simplified definition that we will use to show the main results.

Let $t_1$, $t_2$ and $t$ be terms.

$t_1 \approx t >_e t_3 \approx t \Leftrightarrow t_1 \neq t \wedge (t_1 > t_3 \vee t_3 = t \vee (t_3 < t \wedge t_1 \# t))$

**Proof:** This is easily checked with definition E.1. □

**Lemma 4.9** Consider a goal $\alpha_1 \approx \alpha_2$, and rule $t_1 \approx t_2$.

We have $\sigma(t_1) \approx \sigma(t_2) <_e \alpha_1 \approx \alpha_2$

if the rule is used:

(A) on a strict subterm of $\alpha_1$

(B) after reducing the goal into $\alpha_1' \approx \alpha_2$

- (a) on the term $\alpha_1'$
- (b) on a strict subterm of $\alpha_2$
- (c) on $\alpha_2$ at head position, if $\alpha_1 \not< \alpha_2$ or $\alpha_1 > \sigma(t_2)$

(C) after reducing the goal into $\alpha_1' \approx \alpha_2'$

**Proof:** We follow the cases of the result:

(A) The rule is used on a strict subterm of $\alpha_1$.
By hypothesis, we have:

$$\alpha_1 \approx \alpha_2 = \alpha_1[\sigma(t_1)]_\omega \approx \alpha_2$$

and since $\alpha_1[\sigma(t_1)]_\omega \approx \alpha_2 \rightarrow \alpha_1[\sigma(t_2)]_\omega \approx \alpha_2$, we have:

$$\alpha_1[\sigma(t_1)]_\omega \approx \alpha_2 >_e \alpha_1[\sigma(t_2)]_\omega \approx \alpha_2 \tag{E.1}$$

By the subterm property of the simplification ordering, we have:

$$\alpha_1[\sigma(t_1)]_\omega > \sigma(t_1)$$
$$\alpha_1[\sigma(t_2)]_\omega > \sigma(t_2)$$

Thus

$$\alpha_1 > \sigma(t_1) \text{ by } \alpha_1 = \alpha_1[\sigma(t_1)]_\omega > \sigma(t_1) \tag{E.2}$$

and by lemma E.5 applied to (E.1), we can devise two cases:

- Either $\alpha_1[\sigma(t_1)]_\omega > \alpha_1[\sigma(t_2)]_\omega$.
  In this case we have

$$\alpha_1 > \sigma(t_2) \text{ by } \alpha_1 = \alpha_1[\sigma(t_1)]_\omega > \alpha_1[\sigma(t_2)]_\omega > \sigma(t_2) \tag{E.3}$$

  To sum up we have $\alpha_1 > \sigma(t_1)$ (E.2) and $\alpha_1 > \sigma(t_2)$ (E.3), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
- Or $\alpha_1[\sigma(t_2)]_\omega \leq \alpha_2$.
  In this case we have

$$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 \geq \alpha_1[\sigma(t_2)]_\omega > \sigma(t_2) \tag{E.4}$$

  To sum up we have $\alpha_1 > \sigma(t_1)$ (E.2) and $\alpha_2 > \sigma(t_2)$ (E.4), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

(B) The rule is used after reducing the goal into $\alpha'_1 \approx \alpha_2$,

(a) on the term $\alpha'_1$
Since $\alpha_1 \approx \alpha_2 \rightarrow^n \alpha'_1 \approx \alpha_2$, we have:

$$\alpha_1 \approx \alpha_2 >_e \alpha'_1 \approx \alpha_2 \tag{E.5}$$

by hypothesis, we have:

$$\alpha'_1 \approx \alpha_2 = \alpha'_1[\sigma(t_1)]_\omega \approx \alpha_2$$

and since $\alpha'_1[\sigma(t_1)]_\omega \approx \alpha_2 \rightarrow \alpha'_1[\sigma(t_2)]_\omega \approx \alpha_2$, we have:

$$\alpha'_1[\sigma(t_1)]_\omega \approx \alpha_2 >_e \alpha'_1[\sigma(t_2)]_\omega \approx \alpha_2 \tag{E.6}$$

By the subterm property of the simplification ordering, we have:

$$\alpha'_1[\sigma(t_1)]_\omega \geq \sigma(t_1)$$
$$\alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2)$$

By lemma E.5 applied to (E.5), we have

$$\alpha_1 \neq \alpha_2 \tag{E.7}$$

and three cases:

- The first case is $\alpha_1 > \alpha'_1$.

  In this case we have

  $$\alpha_1 > \sigma(t_1) \text{ by } \alpha_1 > \alpha'_1 = \alpha'_1[\sigma(t_1)]_\omega \geq \sigma(t_1) \tag{E.8}$$

  and by lemma E.5 applied to (E.6), we can devise two subcases:

  · Either $\alpha'_1[\sigma(t_1)]_\omega > \alpha'_1[\sigma(t_2)]_\omega$.
  In this case we have

  $$\alpha_1 > \sigma(t_2) \text{ by } \alpha_1 > \alpha'_1 = \alpha'_1[\sigma(t_1)]_\omega > \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.9}$$

  To sum up we have $\alpha_1 > \sigma(t_1)$ (E.8) and $\alpha_1 > \sigma(t_2)$ (E.9),
  thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

  · Or $\alpha'_1[\sigma(t_2)]_\omega \leq \alpha_2$.
  In this case we have

  $$\alpha_2 \geq \sigma(t_2) \text{ by } \alpha_2 \geq \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.10}$$

  To sum up we have $\alpha_1 > \sigma(t_1)$ (E.8), $\alpha_2 \geq \sigma(t_2)$ (E.10) and $\alpha_1 \neq \alpha_2$ (E.7),
  thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

- The second case would be $\alpha'_1 = \alpha_2$, but it is not possible, because it contradicts (E.6).

- The third case is $\alpha'_1 < \alpha_2 \wedge \alpha_1 \# \alpha_2$.

  In this case we have

  $$\alpha_2 > \sigma(t_1) \text{ by } \alpha_2 > \alpha'_1 = \alpha'_1[\sigma(t_1)]_\omega \geq \sigma(t_1) \tag{E.11}$$

  and by lemma E.5 applied to (E.6), we can devise two cases:

  · Either $\alpha'_1[\sigma(t_1)]_\omega > \alpha'_1[\sigma(t_2)]_\omega$.
  In this case we have

  $$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 > \alpha'_1 = \alpha'_1[\sigma(t_1)]_\omega > \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.12}$$

  To sum up we have $\alpha_2 > \sigma(t_1)$ (E.11) and $\alpha_2 > \sigma(t_2)$ (E.12),
  thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

  · Or $\alpha'_1[\sigma(t_2)]_\omega \leq \alpha_2$.
  In this case we have

  $$\alpha_2 \geq \sigma(t_2) \text{ par } \alpha_2 \geq \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.13}$$

  To sum up we have $\alpha_2 > \sigma(t_1)$ (E.11), $\alpha_2 \geq \sigma(t_2)$ (E.13) and $\alpha_1 \# \alpha_2$ (hypothesis),
  thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

(b) on a strict subterm of $\alpha_2$

Since $\alpha_1 \approx \alpha_2 \to^n \alpha'_1 \approx \alpha_2$, we have:

$$\alpha_1 \approx \alpha_2 >_e \alpha'_1 \approx \alpha_2 \tag{E.14}$$

By hypothesis, we have:

$$\alpha'_1 \approx \alpha_2 = \alpha'_1 \approx \alpha_2[\sigma(t_1)]_\omega$$

and since $\alpha'_1 \approx \alpha_2[\sigma(t_1)]_\omega \to \alpha'_1 \approx \alpha_2[\sigma(t_2)]_\omega$, we have:

$$\alpha'_1 \approx \alpha_2[\sigma(t_1)]_\omega >_e \alpha'_1 \approx \alpha_2[\sigma(t_2)]_\omega \tag{E.15}$$

By the subterm property of the simplification ordering, we have:

$$\alpha_2[\sigma(t_1)]_\omega > \sigma(t_1)$$
$$\alpha_2[\sigma(t_2)]_\omega > \sigma(t_2)$$

thus

$$\alpha_2 > \sigma(t_1) \text{ by } \alpha_2 = \alpha_2[\sigma(t_1)]_\omega > \sigma(t_1) \tag{E.16}$$

and by lemma E.5 applied to (E.15), we can devise two cases:

- $\alpha_2[\sigma(t_1)]_\omega > \alpha_2[\sigma(t_2)]_\omega$.
  In this case we have

  $$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 = \alpha_2[\sigma(t_1)]_\omega > \alpha_2[\sigma(t_2)]_\omega > \sigma(t_2) \tag{E.17}$$

  To sum up we have $\alpha_2 > \sigma(t_1)$ (E.16) and $\alpha_2 > \sigma(t_2)$ (E.17), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
- $\alpha_2[\sigma(t_2)]_\omega \leq \alpha'_1$.
  In this case, by lemma E.5 applied to (E.14), we can devise two cases:
  - $\alpha_1 > \alpha'_1$.
    In this case we have

    $$\alpha_1 > \sigma(t_2) \text{ by } \alpha_1 > \alpha'_1 \geq \alpha_2[\sigma(t_2)]_\omega > \sigma(t_2) \tag{E.18}$$

    To sum up we have $\alpha_2 > \sigma(t_1)$ (E.16) and $\alpha_1 > \sigma(t_2)$ (E.18), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
  - $\alpha'_1 \leq \alpha_2$.
    In this case we have

    $$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 \geq \alpha'_1 \geq \alpha_2[\sigma(t_2)]_\omega > \sigma(t_2) \tag{E.19}$$

    To sum up we have $\alpha_2 > \sigma(t_1)$ (E.16) and $\alpha_2 > \sigma(t_2)$ (E.19), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

(c) on $\alpha_2$ at head position, with $\alpha_1 \not< \alpha_2$ or $\alpha_1 > \sigma(t_2)$

Since $\alpha_1 \approx \alpha_2 \to^n \alpha'_1 \approx \alpha_2$, we have:

$$\alpha_1 \approx \alpha_2 >_e \alpha'_1 \approx \alpha_2 \tag{E.20}$$

by hypothesis, we have:

$$\alpha'_1 \approx \alpha_2 = \alpha'_1 \approx \sigma(t_1)$$

and since $\alpha'_1 \approx \sigma(t_1) \to \alpha'_1 \approx \sigma(t_2)$, we have:

$$\alpha'_1 \approx \sigma(t_1) >_e \alpha'_1 \approx \sigma(t_2) \tag{E.21}$$

By lemma E.5 applied to (E.21), we have:

$$\alpha_1 \neq \alpha_2 \tag{E.22}$$

We can devise two cases:

- $\sigma(t_1) > \sigma(t_2)$.
  In this case we have

  $$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 = \sigma(t_1) > \sigma(t_2) \tag{E.23}$$

  We have to proceed by cases between $\alpha_1$ and $\alpha_2$:
  - $\alpha_1 = \alpha_2$.
    This contradicts (E.20) and thus cannot be the case.
  - $\alpha_1 \# \alpha_2$.
    In this case we have $\alpha_2 = \sigma(t_1)$ (hypothesis) $\alpha_2 > \sigma(t_2)$ (E.23) and $\alpha_1 \# \alpha_2$,
    thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
  - $\alpha_1 > \alpha_2$.
    In this case we have $\alpha_1 > \sigma(t_1)$ and $\alpha_1 > \sigma(t_2)$,
    thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
  - $\alpha_1 < \alpha_2$.
    In this case we have $\alpha_2 > \alpha_1$, $\sigma(t_1) > \sigma(t_2)$, $\alpha_2 = \sigma(t_1)$ and $\alpha_1 > \sigma(t_2)$,
    thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
- Or $\sigma(t_2) \leq \alpha'_1\cdot$
  In this case by lemma E.5 applied to (E.20), we have three cases:
  - $\alpha_1 > \alpha'_1$.
    In this case we have

    $$\alpha_1 > \sigma(t_2) \text{ by } \alpha_1 > \alpha'_1 \geq \sigma(t_2) \tag{E.24}$$

    To sum up we have $\alpha_2 = \sigma(t_1)$, $\alpha_1 > \sigma(t_2)$ (E.24) and $\alpha_1 \neq \alpha_2$ (E.22),
    thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
  - $\alpha'_1 = \alpha_2$.
    This contradicts (E.21) and thus cannot be the case.
  - $\alpha'_1 < \alpha_2 \wedge \alpha_1 \# \alpha_2$.
    In this case we have

    $$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 > \alpha'_1 \geq \sigma(t_2) \tag{E.25}$$

    To sum up we have $\alpha_2 = \sigma(t_1)$ $\alpha_2 > \sigma(t_2)$ (E.25) and $\alpha_1 \# \alpha_2$ (hypothesis),
    thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

(C) The rule is used after reducing the goal into $\alpha'_1 \approx \alpha'_2$.
Since $\alpha_1 \approx \alpha_2 \rightarrow^n \alpha'_1 \approx \alpha'_2$, we have:

$$\alpha_1 \approx \alpha_2 >_e \alpha'_1 \approx \alpha'_2 \tag{E.26}$$

by hypothesis, we have:

$$\alpha'_1 \approx \alpha'_2 = \alpha'_1[\sigma(t_1)]_\omega \approx \alpha'_2$$

and since $\alpha'_1[\sigma(t_1)]_\omega \approx \alpha'_2 \rightarrow \alpha'_1[\sigma(t_2)]_\omega \approx \alpha'_2$, we have:

$$\alpha'_1[\sigma(t_1)]_\omega \approx \alpha'_2 >_e \alpha'_1[\sigma(t_2)]_\omega \approx \alpha'_2 \tag{E.27}$$

By the subterm property of the simplification ordering, we have:

$$\alpha'_1[\sigma(t_1)]_\omega \geq \sigma(t_1)$$
$$\alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2)$$

By definition E.1 applied to (E.26), we can devise three cases:

- $\alpha_1 > \alpha'_1$.

  In this case we have

$$\alpha_1 > \sigma(t_1) \text{ by } \alpha_1 > \alpha'_1 = \alpha'_1[\sigma(t_1)]_\omega \geq \sigma(t_1) \tag{E.28}$$

  and by lemma E.5 applied to (E.27), we can devise two cases:

  · Either $\alpha'_1[\sigma(t_1)]_\omega > \alpha'_1[\sigma(t_2)]_\omega$.

  In this case we have

$$\alpha_1 > \sigma(t_2) \text{ by } \alpha_1 > \alpha'_1 = \alpha'_1[\sigma(t_1)]_\omega > \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.29}$$

  To sum up we have $\alpha_1 > \sigma(t_1)$ (E.28) and $\alpha_1 > \sigma(t_2)$ (E.29), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

  · Or $\alpha'_1[\sigma(t_2)]_\omega \leq \alpha'_2$.

  In this case, by definition E.1 applied to (E.26), we have two cases

  Either $\alpha_1 > \alpha'_2$.
  In this case we have

$$\alpha_1 > \sigma(t_2) \text{ by } \alpha_1 > \alpha'_2 \geq \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.30}$$

  To sum up we have $\alpha_1 > \sigma(t_1)$ (E.28) and $\alpha_1 > \sigma(t_2)$ (E.30), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
  Or $\alpha_2 > \alpha'_2$.
  In this case we have

$$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 > \alpha'_2 \geq \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.31}$$

  To sum up we have $\alpha_1 > \sigma(t_1)$ (E.28) and $\alpha_2 > \sigma(t_2)$ (E.31), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

- $\alpha_2 > \alpha'_1$.
  In this case we have

$$\alpha_2 > \sigma(t_1) \text{ by } \alpha_2 > \alpha'_1 = \alpha'_1[\sigma(t_1)]_\omega \geq \sigma(t_1) \tag{E.32}$$

  and by lemma E.5 applied to (E.27), we can devise two subcases:
  - · Or $\alpha'_1[\sigma(t_1)]_\omega > \alpha'_1[\sigma(t_2)]_\omega$.
    In this case we have

$$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 > \alpha'_1 = \alpha'_1[\sigma(t_1)]_\omega > \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.33}$$

    To sum up we have $\alpha_2 > \sigma(t_1)$ (E.32) and $\alpha_2 > \sigma(t_2)$ (E.33), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
  - · Or $\alpha'_1[\sigma(t_2)]_\omega \leq \alpha'_2$.
    In this case, by definition E.1 applied to (E.26), we have two subcases

    Either $\alpha_1 > \alpha'_2$.
    In this case we have

$$\alpha_1 > \sigma(t_2) \text{ by } \alpha_1 > \alpha'_2 \geq \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.34}$$

    To sum up we have $\alpha_2 > \sigma(t_1)$ (E.32) and $\alpha_1 > \sigma(t_2)$ (E.34), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.
    Or $\alpha_2 > \alpha'_2$.
    In this case we have

$$\alpha_2 > \sigma(t_2) \text{ by } \alpha_2 > \alpha'_2 \geq \alpha'_1[\sigma(t_2)]_\omega \geq \sigma(t_2) \tag{E.35}$$

    To sum up we have $\alpha_2 > \sigma(t_1)$ (E.32) and $\alpha_2 > \sigma(t_2)$ (E.35), thus $\alpha_1 \approx \alpha_2 >_e \sigma(t_1) \approx \sigma(t_2)$.

- $\alpha_1 \neq \alpha_2$.
  We have four subcases: $\alpha_1 = \sigma(t_1)$, $\alpha_1 = \sigma(t_2)$, $\alpha_2 = \sigma(t_1)$ and $\alpha_2 = \sigma(t_2)$. Each of them contradicts the hypothesis and thus cannot happen.

□

## Contents

1 Introduction 1

2 Deduction Modulo 4

2.1 Sequent calculus modulo 4

2.2 Relating deduction modulo and classical deduction: compatibility 6

2.3 Modularity of compatibility 8

2.4 Conditional protected equality 10

2.5 Deduction modulo using conditional equational systems 15

3 $HOL_{\lambda\sigma}$, a first-order expression of higher-order logic 16

4 Proof by induction 19

4.1 Noetherian induction 20

4.2 Inductive consequence and inductive property 21

4.3 Using the induction hypothesis 22

4.4 Internalization of induction hypotheses 23

4.5 Using the membership hypothesis 25

4.6 Simplifying the goal 25

4.7 The induction ordering 26

5 Proving inductive properties by rewriting 28

5.1 Simple examples in Peano arithmetics 29

5.2 The general case 36

6 Conclusion 37

References 38

A Proof of lemma 2.1 42

B Proof of lemma 2.5 50

C Proof of lemma 4.2 55

D Proof of lemma 4.3 61

E Proof of lemma 4.9 67