Dual-context multicategories as models for implicit computational complexity

Fedor Fomenko

Doctor of Philosophy

Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh

2006

Dual-context multicategories as models for implicit computational complexity

Fedor Fomenko

Doctor of Philosophy

Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh

2006

Abstract

In this thesis we study *dual-context* type systems and their models. A dual-context type system is one in which the context of a term is split into two parts, a *normal* part and a *safe* part. This separation allows one to put different kinds of usage restrictions on the two types of variable. For example, in implicit computational complexity, such a separation is used to implement resource-free characterisations of complexity classes. A similar separation between two kinds of variable occurs in type systems for linear logic, where different structural operations are permitted in the normal and safe parts of the context.

We start by defining two basic dual-context calculi $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$ of typed terms, built using two kinds of free variables and dual-typed function symbols. In the $\mathbb{IIL}(\Sigma)$ calculus we use safe variables in a 'linear' fashion, forbidding any weakening and contraction, while, in the $\mathbb{III}(\Sigma)$ calculus, contraction and weakening are allowed for both safe and normal variables.

We then consider models for $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$ with basic equational judgements. Rather than following the traditional approach of encoding dual-contexts through additional structure on a category, we consider *dual-context multicategories* in which dual-contexts are built into the definition. The main advantage this approach is that it covers a wider class of models, including some particularly natural models from the field of implicit computational complexity.

Next we enrich our basic calculi with different type constructions such as products and sums and provide their multicategorical interpretation. We consider a dual-context list type constructor together with a type-theoretic analogue of the *safe recursion* of Bellantoni and Cook's system $\mathfrak{B}$, which characterises polynomial time computability on natural numbers. We define an interpretation for such dual-context lists using *safe list objects*. We show that the polytime computable functions are exactly the functions definable in every dual-context multicategory with safe binary list object.

Finally, motivated by the standard approach to the categorical interpretation of primitive recursion using the notion of initial algebra, we develop a notion of *safe initial algebra*, which generalises the safe recursion scheme and provides us with insights about the choice of initial functions in system $\mathfrak{B}$.

Acknowledgements

I am hugely grateful to my supervisor Alex Simpson who spent countless hours of his time discussing and reading this thesis, shared his insights and helped me to overcome difficulties. My second supervisor John Longley was invaluable source of moral support and encouragement and helped me to shape a better understanding of some of the problems I was trying to solve in this thesis. I am also very grateful to Gordon Plotkin who gave many helpful comments on the subject of the thesis. I would like to thank Masahito Hasegawa for invitation to RIMS Kyoto University where some of the fundamental ideas of this thesis were conceived.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Fedor Fomenko*)

To my wife and parents for their trust and support

Table of Contents

Chapter 1

Introduction

A type system provides a formal set of rules for dividing programs into classes called *types*. A statement that a given program belongs to a given type can be formally written as a *typing judgement* of the form $\Gamma \vdash t : \sigma$, where Γ is a *context*, t is a *term*, representing the program and σ is a type.

Usually the context for a term is a sequence of typings of the form $x_1 : \sigma_1, \ldots, x_n : \sigma_n$ where each typing $x_i : \sigma_i$ assigns the type σ_i to the variable x_i. Variables themselves are terms, which can be typed using the following rule:

$$\frac{}{\Gamma, x_i : \sigma_i, \Delta \vdash x_i : \sigma_i}$$

There are special rules whose purpose is to allow different manipulations on the context of a term. Such rules are called *structural* and play an important role. For example, the following rule allows one to introduce a new typing (such an operation is called *weakening*):

$$\frac{x_1 : \sigma_1, \ldots, x_n : \sigma_n \vdash t : \rho}{x_1 : \sigma_1, \ldots, x_n : \sigma_n, y : \tau \vdash t : \rho}$$

Other examples of structural rules include the *exchange rule* which allows the permutation of typings in a context and the *contraction rule* for replacing a variable in a term by another variable of the same type. There are interesting type systems in which some of these structural rules are forbidden or restricted in some way. One well-known example of such a type system is associated with Girard's linear logic [Gir87].

In this thesis we shall be studying type systems with contexts which are divided into two parts. Such contexts will be called *dual-contexts*. A typing judgement in such type systems will have the following form:

$$x_1 : \sigma_1, \ldots, x_n : \sigma_n ; y_1 : \tau_1, \ldots, y_m : \tau_m \vdash t : \rho$$

where the separator ; is used to divide the dual-context into two parts.

Situations when it is convenient to divide variables into two classes with different properties are quite common. For example, a number of type systems with two kinds of variables were studied in the logic programming community [Mil94, HM94, PH94, HPW96].

In a dual-context setting we can consider type systems with different structural rules allowed for the different kinds of variables. We shall see several examples of such systems in this thesis. In this chapter we shall discuss why are we interested in dual-context type systems, give a few examples of such systems and outline our approach and main results.

1.1 Background

In this section we will present the relevant background for our research. We start with a brief description of a dual-context type system for the intuitionistic linear logic which is called DILL. This type system will be our main motivational example. Then we shall describe a functional system with a restricted recursion scheme, called system $\mathfrak{B}$which characterizes polynomial time computability on the natural numbers. The restrictions used to achieve it are formulated using the separation of variables into two kinds. One of the type systems studied in this thesis will be developed following the ideas behind the system $\mathfrak{B}$.

1.1.1 Dual Intuitionistic Linear Logic

In this section we shall briefly describe a type system in which dual-contexts are used explicitly. This type system is called DILL which stands for 'dual intuitionistic linear logic' and is a dual-context natural deduction formulation of intuitionistic linear logic [Gir87]. It was developed by Barber and Plotkin [Bar96b, Bar96a].

The set of types in DILL consist of some basic types, unit type I, monoidal product types $\sigma \otimes \tau$, linear function types $\sigma \multimap \tau$ and exponential types $!\sigma$. A dual-context is a pair of sequences of typings, written as $x_1 : \sigma_1, \ldots, x_n : \sigma_n ; y_1 : \tau_1, \ldots, y_m : \tau_m$. The variables $x_1, \ldots, x_n$ are called *intuitionistic* and the variables $y_1, \ldots, y_m$ are called *linear*.

A typing judgement of DILL has the following form:

$$x_1 : \sigma_1, \ldots, x_n : \sigma_n ; y_1 : \tau_1, \ldots, y_m : \tau_m \vdash t : \rho$$

We shall be mostly interested in the different structural rules, which are admissible in DILL, since these rules illustrate the use of dual-contexts. We start with the rules which are used for typing of intuitionistic and linear variables. Note that in the intuitionistic case the typing context does not contain any linear variables:

$$\frac{}{\Gamma, x:\sigma; \vdash x:\sigma}\ \textbf{Int-Ax} \qquad \frac{}{\Gamma; y:\tau \vdash y:\tau}\ \textbf{Lin-Ax}$$

For the intuitionistic variables the following weakening and contraction rules are admissible in DILL:

$$\frac{\Gamma; \Delta \vdash t:\rho}{\Gamma, x:\sigma; \Delta \vdash t:\rho} \qquad \frac{\Gamma, x:\sigma, z:\sigma; \Delta \vdash t:\rho}{\Gamma, x:\sigma; \Delta \vdash t[x/z]:\rho}$$

where $t[x/z]$ denotes a substitution of the variable x for the variable z. The usage of the linear variables is more restrictive and the weakening and the contraction rules for the linear variables are not admissible.

We have seen how the substitution operation was used in the contraction rule above. In fact the contraction rule is a special case of a general substitution rule, which describes how the operation of substituting of a term for a free variable in another term is typed. Since we have two kinds of variables in the dual-context setting we also have two kinds of substitutions. The substitution for a linear variable in a term is typed by the following rule:

$$\frac{\Gamma; \Delta_1 \vdash s:\sigma \qquad \Gamma; \Delta_2, y:\sigma \vdash t:\tau}{\Gamma; \Delta_1 \# \Delta_2 \vdash t[s/y]:\tau}$$

where $\Delta_1 \# \Delta_2$ is an arbitrary sequence of typings obtained by concatenation of typings from Δ_1 and Δ_2 in any order. As a special case of linear substitution the following rule is admissible:

$$\frac{\Gamma; x:\sigma, \Delta \vdash t:\rho}{\Gamma, x:\sigma; \Delta \vdash t:\rho}$$

This rule allows one to change the kind of variable from safe to normal.

Substitution for an intuitionistic variable is described by the following rule:

$$\frac{\Gamma; \vdash s:\sigma \qquad \Gamma, x:\sigma; \Delta \vdash t:\tau}{\Gamma; \Delta \vdash t[s/x]:\tau}$$

Note that the term s must not contain any linear variables. This is a fundamental restriction which can be understood intuitively as follows. Intuitionistic variables can be used liberally in a term while the usage of linear variables is limited. If we allow substitution of a term which contains linear variables for an intuitionistic variable then it would be possible to violate the linearity restrictions.

One of the benefits of DILL compared to the single-context formulations of **ILL** see e.g. [BBHdP93b, BBHdP93a] is the handling of the exponential !, which makes the substitution rules given above admissible. See [Bar96b] for a thorough comparison of the two approaches.

A sound and complete interpretation of DILL can be given using *linear/non-linear models* [Ben95] which consist of a cartesian closed category $\mathbf{C}$, a symmetric monoidal closed category $\mathbf{M}$ and a pair of strong monoidal functors $F : \mathbf{C} \to \mathbf{M}$ and $G : \mathbf{M} \to \mathbf{C}$, which form a monoidal adjunction $G \vdash F$. Such models generalise previous notions of models of linear logic [Bie94, See89, Laf88].

The interpretation works as follows. Assume a function which maps every basic type σ to an object $[\![\sigma]\!] \in |\mathbf{M}|$. This function can be extended to give an interpretation of non-basic types using the structure of linear/non-linear model. Then every DILL term $x_1 : \sigma_1, \ldots, x_n : \sigma_n ; y_1 : \tau_1, \ldots, y_m : \tau_m \vdash t : \rho$ can be interpreted as a morphism

$$FG([\![\sigma_1]\!]) \otimes \cdots \otimes FG([\![\sigma_n]\!]) \otimes [\![\tau_1]\!] \cdots \otimes [\![\tau_m]\!] \xrightarrow{[\![t]\!]} [\![\rho]\!]$$

Another possible interpretation of DILL uses *linear categories* [BBHdP93a], which consist of a symmetric monoidal category $\mathbf{L}$ together with a symmetric monoidal comonad $! : \mathbf{L} \to \mathbf{L}$ satisfying several additional conditions. Benton [Ben95] showed that linear/non-linear models and linear categories are equivalent.

The important features of DILL can be summarized as follows:

- The terms of DILL contain two kinds of free variables, which are typed in the separate parts of contexts.
- The substitution in DILL is restricted - no terms which contained free linear variables were allowed to be substituted for the intuitionistic variables in other terms.
- The weakening and the contraction rules are admissible only for the intuitionistic variables.

In this thesis we shall describe a syntactic framework for describing a general dual-context type system with the same of restriction on the substitution but a different set of structural rules.

1.1.2 Type systems for describing feasible computations

In this section we shall see another example of a dual-context system. This system will exhibit several features of DILL, which we described in the previous section.

In particular, two kinds of variables will be considered and the substitution will be restricted in the same way. One difference compared to DILL will be that weakening and contraction will be allowed for both kinds of variables. Another distinction is that this system is untyped. Before we describe this system, we briefly describe it's origin.

Traditional models of computability such as Turing machines or recursive functions describe computations which may not terminate in general. Even when a computation is terminating it may require enormous amount of time or memory in order to produce a result even for inputs of small size. So such traditional models are too powerful to describe precisely *feasible* or practically possible computations.

A number of restrictions have been suggested for delineating computations which can be regarded as feasible. The most direct approach is to restrict the execution time of a Turing machine by a polynomial. In such model a machine working on any input must terminate in a number of steps which is less than a value of a of some polynomial in the length of input written in binary notation. Any machine for which there exist such polynomial is called *polynomial time computable* (polytime computable for short). The class of polytime computations has some interesting properties and is an important topic of research in complexity theory.

A number of very different models were proposed for describing the notion of general computation. It was shown that all of them are equivalent which suggests a thesis that they adequately formalize the notion of computation. It is therefore natural to ask if we can find an alternative equivalent models for describing polytime computations. Such models would suggest that polytime computability is a good formalization of feasible computations.

Let us consider the class of computable functions called *primitive recursive functions*. This class can be described as the smallest class of functions containing some initial functions and closed under function composition and the primitive recursion scheme, which allows one to define a new function f from given functions g and h by:

$$f(0,x_1,\ldots,x_n) = g(x_1,\ldots,x_n)$$
$$f(n+1,x_1,\ldots,x_n) = h(n,x_1,\ldots,x_n,f(n,x_1,\ldots,x_n))$$

In this setting we can take the length of $f(a_1,\ldots,a_n)$ written in binary notation as lower bound for the execution time of the function on inputs $a_1,\ldots,a_n$. This length will be denoted as $|f(\bar{x})|$ and we will write $|\bar{x}|$ for $|x_1|+\cdots+|x_n|$. We can find a primitive recursive function f for which there does not exist a polynomial p such that $|f(\bar{x})| \leq p(|\bar{x}|)$. So the model of primitive recursive functions is too powerful to describe exactly

the polytime functions.

Cobham [Cob65] suggested replacing the primitive recursion scheme by the following *bounded recursion on notation* scheme, which defines the function f from the given functions g, h and k:

$$f(0,\bar{x}) = g(\bar{x})$$
$$f(n+1,\bar{x}) = h(n,\bar{x},f(\lfloor n/2 \rfloor,\bar{x})$$
$$f(n,\bar{x}) \leq k(n,\bar{x})$$

The function f is defined by this scheme if the bounding condition $f(n,\bar{x}) \leq k(n,\bar{x})$ is satisfied for all n. Cobham proved that the class of polytime computable functions is the smallest class of functions containing various initial functions and closed under function composition and the bounded recursion on notation schemes. Among the basic functions are projections, binary successor functions, parity, integer division by 2, conditional, the constant 0 functions and the smash function $x \odot y = 2^{|x|\cdot|y|}$. The role of the smash function is to provide large enough bounds in order to get the bounded recursion of the ground.

Cobham's scheme provides an equivalent model for describing polytime computations but is difficult to work with. One has to come up with a bounding function in order to use the bounded recursion scheme. Verifying that a particular definition is valid is in general undecidable. Cook and Urquhart [CU93] suggested changing the semantics of bounded recursion to cut off at size overflow:

$$f(x,\bar{y}) = \min\{k(x,\bar{y}), h(x,\bar{y},f(\lfloor x/2 \rfloor,\bar{y}))\}$$

A more principled approach, avoiding boundary functions entirely, was proposed by Bellantoni and Cook [BC92], who considered functions on the natural numbers with two kinds of inputs. Such functions are denoted as $f(x_1,\ldots,x_n\,;y_1,\ldots,y_m)$. The variables $x_1,\ldots,x_n$ are called *normal* and the variables $y_1,\ldots,y_m$ are called *safe*.

Definition 1.1.1 (Bellantoni-Cook)**.**
The class $\mathfrak{B}$ is the smallest class of functions on the natural numbers with two kinds of inputs containing:

- The zero function 0
- Projections $\pi_j^{n,m}(x_1,\ldots,x_n;x_{n+1},\ldots,x_{n+m}) = x_j$
- The binary notation successor functions $s_i(;a) = 2a+i$, for $i \in \{0,1\}$

- The binary notation predecessor function $p(;x) = \lfloor x/2 \rfloor$
- The conditional function

$$C(;x,y,z) = \begin{cases} y & \text{if } x \text{ is even,} \\ z & \text{otherwise} \end{cases}$$

and closed under:

- Given functions h, $s_1,\ldots,s_n$ and $t_1,\ldots,t_m$ in $\mathfrak{B}$ the following *safe composition* scheme defines a function in $\mathfrak{B}$ as well:

$$f(\bar{x};\bar{y}) = h(\bar{s}(\bar{x};);\bar{t}(\bar{x};\bar{y}))$$

  where $\bar{s}(\bar{x};)$ is an abbreviation for $s_1(\bar{x};),\ldots,s_n(\bar{x};)$.

- Given functions g, h_0 and h_1 in $\mathfrak{B}$ the following *safe recursion on notation* scheme defines a function in $\mathfrak{B}$ as well:

$$\begin{aligned} f(0,\bar{x};\bar{y}) &= g(\bar{x};\bar{y}) \\ f(z,\bar{x};\bar{y}) &= h_0(z,\bar{x};\bar{y},f(\lfloor z/2 \rfloor,\bar{x};\bar{y})) \quad z>0 \text{ and } z \text{ is even} \\ f(z,\bar{x};\bar{y}) &= h_1(z,\bar{x};\bar{y},f(\lfloor z/2 \rfloor,\bar{x};\bar{y})) \quad z>0 \text{ and } z \text{ is odd} \end{aligned}$$

The main theorem about the system $\mathfrak{B}$ can be stated as follows:

Theorem 1.1.2 (Bellantoni-Cook). *For any $f(\bar{x};\bar{y}) \in \mathfrak{B}$ the following holds:*

- *f is bounded by a monotone polynomial p_f*

$$|f(\bar{x};\bar{y})| \leq p_f(|\bar{x}|) + \max_j |y_j|$$

- *f is polytime computable*
- *If $g : \mathbb{N}^n \to \mathbb{N}$ is a polytime computable function then $g(x_1,\ldots,x_n;) \in \mathfrak{B}$*

It is worth emphasizing the following 'structural' features of the system $\mathfrak{B}$:

1. Using the safe composition and projection functions we can perform the following manipulations with an $f \in \mathfrak{B}$:

- **Add dummy normal and safe inputs**
  Given $f(x;y)$ we can define the following function:

  $$f'(x,u;y,z) = f(\pi_1^{2,0}(x,u;)\pi_3^{2,2}(x,u;y,z))$$

  The introduction of a safe dummy variable also works if f has only normal inputs

  $$f'(x;y) = f(\pi_1^{1,0}(x;);)$$

  Similarly for f with no inputs at all $f'(\bar{x};\bar{y}) = f(;)$ is a valid function definition by safe composition.

- **Copy normal and safe inputs**
  Given $f(u,v;z,w)$ we can define the following function:

  $$f'(u;z) = f(\pi_1^{1,0}(u;),\pi_1^{1,0}(u;);\pi_2^{2,0}(u;z),\pi_2^{2,0}(u;z))$$

- **Shift normal inputs to safe inputs**
  Given $f(;y)$ we can define the following function:

  $$f'(y;) = f(\pi_1^{1,0}(y;);)$$

  Note that the restriction on safe composition prevents shifting safe inputs to normal.

2. The restrictions on safe composition prevent the substitution of functions depending on safe variables for normal variables.

3. The successor functions, the predecessor function and the conditional function are all defined with safe inputs. Since we can shift normal inputs to safe we can also define same function with normal inputs.

4. In the safe recursion scheme the argument of recursion must be normal and the recursive value is substituted into the safe position. This feature together with the restriction on composition (point 2 above) prevents nested recursions.

Another characterisation of polytime computability on the natural numbers was given by Leivant and Marion [Lei93, LM93]. Instead of just two sorts of inputs in functions they assigned a natural number called **tier** to every input of a function and to the function itself. A function of tier k can be substituted for an input of tier l only

if $l \leq k$. In their version of recursion on notation scheme a tier of a function defined by recursion should be lower than a tier of the recursion variable. All the polytime computable functions on the natural numbers can be defined using the zero function, projections and successors available on all tiers and the composition and the recursion schemes just outlined. Hofmann [Hof97] showed that this approach is a generalisation of Bellantoni and Cook's approach. Otto [Ott95] defined a categorical framework based on Leivant's system, in which polytime computable and Kalmar's elementary functions on the natural numbers could be characterised.

The system $\mathfrak{B}$ presented in this section allows one to describe polytime computations on the natural numbers without any references to polynomial bounds and is easy to work with. It shares two features with DILL type system described earlier. First, the functions definable in the system $\mathfrak{B}$ have two kinds of inputs. Second, the function composition scheme is restricted in such a way that no function with safe inputs can be plugged into a normal input of another function. The main structural difference compared to DILL is that it is possible to duplicate safe arguments and introduce additional safe arguments in function definitions.

1.2 Contributions and outline of the thesis

In this section we shall briefly describe our contributions and outline the structure of the rest of the thesis. We start with a brief description of two dual-context calculi of terms. Then we shall outline our approach to semantics of dual-context-calculi based on the notion of *dual-context multicategory*. Finally consider possible extensions of dual-context calculus and in particular extension with the *safe natural numbers type*.

1.2.1 Basic dual-context calculi

In this section we shall give a brief description of dual-context calculi we are interested in. Earlier in this chapter we presented DILL, a dual-context type system with a sophisticated set of types, including monoidal products, linear functions and exponentials. In the first half of this thesis we shall consider much simpler dual-context systems, which have only the basic types.

Similar to DILL, our calculi will operate with typing judgements of the following form:

$$x_1 : \sigma, \ldots, x_n : \sigma_n ; y_1 : \tau, \ldots, y_m : \tau_m \vdash t : \rho$$

where σ_i, τ_j and ρ are basic types and t is a term. Following Bellantoni and Cook we shall refer to the variables $x_1, \ldots, x_n$ as *normal* and to the variables $y_1, \ldots, y_m$ as *safe*.

Pre-terms of our calculi will consist of two kinds. First, the terms of the form x_i, where x_i is a variable. Such terms can be typed using the axiom rules, which are similar to the axiom rules of DILL. Second, the terms of the form $f(t_1, \ldots, t_n; s_1, \ldots, s_m)$, where t_i and s_j are terms and f is a *function symbol*. The first n arguments of f will be called normal and the remaining m arguments will be called safe. Each argument is assigned a type. Function symbols come from a *signature*, which also contains a set of basic types. Each dual-context calculus we define in this thesis will be given for a particular signature.

Since our dual-context calculi have only basic types, we will focus our attention on the structural properties of such calculi. A first important structural property is the form of term substitution that can be typed. In DILL any term which depends on linear variables can not be substituted for intuitionistic variables. On the other hand substituting a term depending on intuitionistic variables for a linear variable is perfectly legal. We have seen the same kind of restriction in Bellantoni and Cook's safe composition scheme. In the dual-context calculi we consider in this thesis, substitution will be restricted in the same way. This restriction can be motivated as follows. Safe variables are usually used in a special way in a dual-context calculus. For example, a calculus may have a limited set of structural rules for safe variables, so that any argument which is passed via a safe variable can not be duplicated. If substitution is not restricted then it would be possible to violate such kind of limitations of usage.

Even with restricted substitution, we have many different choices for the possible admissible structural rules. In this thesis we will consider two basic dual-context calculi:

1. The $\mathbb{III}(\Sigma)$ calculus in which weakening and contraction for both the safe and the normal variables will be admissible. This calculus will have the same structural properties as Bellantoni and Cook's system $\mathfrak{B}$.

2. The $\mathbb{IIL}(\Sigma)$ calculus in which weakening and contraction will be admissible for the normal variables only. This calculus will be essentially the structural core of DILL.

After considering two examples of basic dual-context calculi with restricted substitution and different structural rules, we develop a framework in which both of these

calculi and many others can be described. The main idea is to consider structural rules as basic rather than as admissible rules.

We observe that many structural rules can be described via *renamings*, which are functions between the dual-contexts, mapping variables in one dual-context to variables in another dual-context while preserving types of variables. The renamings can be thought of as primitive substitutions where only variables can be substituted. A class of such renamings, which satisfy some closure conditions, can be used to describe a set of admissible structural rules in a dual-context calculus.

Our syntactic framework for describing dual-context calculi will be based on the dual-context calculus $\mathbb{LL}(\Sigma)$ in which both the safe and the normal variables will be treated linearly. Adding to such calculus structural rules, described by a particular set of renamings we obtain another dual-context calculus. For example, one can construct in this way two calculi which are equivalent to the previously considered $\mathbb{II}(\Sigma)$ and $\mathbb{IL}(\Sigma)$. Many other variations are possible too.

1.2.2 Semantics of basic dual-context calculi

Our approach to semantics of dual-context type systems is different from the one taken in DILL and is based on a modified notion of *multicategory*. One reason for such a choice is the fact that the semantical model of DILL does not scale down very well to our calculi, which have no type constructors.

To explain our approach we consider how it works in a single-context case. Consider a *many-sorted signature* Σ which contains a set of basic types $|\Sigma|$ and a set of typed function symbols $f : \sigma_1 \dots \sigma_n \to \tau$, where $\sigma_1, \dots, \sigma_n$ and τ are basic types from $|\Sigma|$. Contexts over this signature are just sets of variable typings of the form $x_1 : \sigma_1, \dots, x_n : \sigma_n$. Pre-terms are either variables or function applications $f(t_1, \dots, t_n)$ where $t_1, \dots, t_n$ are pre-terms and $f \in \Sigma$ is a function symbol.

Given a signature Σ the calculus $I(\Sigma)$ has the following typing rules [Jac01]:

$$\frac{}{x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash x_i : \sigma_i} \qquad \frac{\Gamma \vdash t_1 : \sigma_1 \quad \dots \quad \Gamma \vdash t_n : \sigma_n}{\Gamma \vdash f(t_1, \dots, t_n) : \tau}$$

It can be shown that the standard structural rules, namely exchange, weakening and contraction, are admissible in this calculus.

The standard categorical semantics of $\mathbf{I}(\Sigma)$ can be given using a category with finite products $\mathbf{C}$. We start with an interpretation of the basic types as objects of $\mathbf{C}$. Since contexts are sets of typings we need to assume an ordering on the variables, so that

contexts can be turned into sequences of typings. Then we can interpret any context $x : \sigma_1, \ldots x_n : \sigma_n$ as an object $[\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]$. Given a term t, typed by the following judgement $x : \sigma_1, \ldots x_n : \sigma_n \vdash t : \tau$, we interpret it as a morphism $[\![\sigma]\!] \times \cdots \times [\![\sigma_n]\!] \to [\![\tau]\!]$.

The disadvantage of this approach is that it requires a cartesian category thus limiting the class of structures in which $\mathbf{I}(\Sigma)$ can be interpreted.

Instead of assuming finite products, a more direct approach is to use a *multicategory* [Lam89, Her00, Lei04]. In Appendix A we give a definition of cartesian multicategory. Like an ordinary category, a cartesian multicategory has of a set of object but instead of morphisms with objects as domains and co-domains it has *multiarrows* which have sequences of objects as domains and objects as co-domains. The multiarrows are denoted as $a_1 \ldots a_n \to b$. Identity morphism are replaced with *projection* multiarrows of the form $a_1 \ldots a_n \to a_i$ and the composition operation has the following signature:

$$\mathbf{M}(a_1 \ldots a_n, b) \times \prod_{i=1}^{n} \mathbf{M}(b_1 \ldots b_m, a_i) \to \mathbf{M}(b_1 \ldots b_m, b)$$

where $\mathbf{M}(a_1 \ldots a_n, b)$ denotes set of all the multiarrows with the sequence $a_1 \ldots a_n$ as domain and object b as co-domain. The composition is written as $f \circ \langle g_1, \ldots, g_n \rangle$.

We also need *symmetry maps* of the form $\cdot\sigma : \mathbf{M}(a_1 \ldots a_n, b) \to \mathbf{M}(a_{\sigma(1)} \ldots a_{\sigma(n)}, b)$ for each object b, each sequence of objects $a_1 \ldots a_n$ and each permutation σ.

Given a cartesian multicategory, we interpret the basic types as objects of multicategory. Assuming an ordering on the variables we can model any term $x_1 : \sigma_1, \ldots x_n : \sigma_n \vdash t : \tau$ as a multiarrow $[\![\sigma_1]\!] \ldots [\![\sigma_n]\!] \xrightarrow{[\![t]\!]} [\![\tau]\!]$. Showing the soundness of such an interpretation is easy. We can modify the standard completeness proof for this setting and construct the *term multicategory*, which is possible even without products, since contexts are considered as primitive and we do not need products in order to model them.

In the dual-context setting the story is very similar. Barber's approach involving linear/non-linear models can be modified to provide a sound interpretation of the $\mathbb{IL}(\Sigma)$ calculus. A categorical semantics for the $\mathbb{II}(\Sigma)$ calculus can also be done with appropriate modifications. But such categorical interpretation will suffer from the same drawback we described earlier. It requires a category with sophisticated structure thus limiting our choice of models.

In this thesis we will develop multicategorical semantics for dual-context calculi based on the notion of *dual-context multicategory*, which has multiarrows with dual-contexts as domains. First, we will consider two variants of *dual-context multicategories*, which will be used for giving a sound and complete interpretation of the $\mathbb{II}(\Sigma)$

and the $\mathbb{IL}(\Sigma)$ calculi. Then we will consider a general notion of *multicategories with renaming* and show that the first two definitions are instances of it. Such multicategories with renamings can be used to give a sound and complete interpretation of any dual-context calculus described by our syntactic framework.

Advantages of using dual-context multicategories

The multicategorical approach to the semantics of the dual-context types systems has several advantages. Such an approach provides a complete interpretation of the basic dual-context calculi unlike the traditional approach which is more suitable for the dual-context type systems with tensor products and exponentials. Moreover, for such extended type systems it is possible to show that our multicategorical approach includes the traditional approach as a special case. Thus the multicategorical approach is more general than the categorical one. Indeed the multicategorical approach allows one to interpret basic dual-context calculi in a broader range of models. In particular, for the interpretation of the single object variant of $\mathbb{III}(\Sigma)$ calculus we can use a set of polytime computable functions on natural numbers, which satisfy the Bellantoni and Cook's invariant:

$$|f(\bar{x};\bar{y})| \leq p_f(|\bar{x}|) + \max_j |y_j|$$

where p_f is a monotone polynomial and $|-|$ denotes length in binary notation. Such an idea will be the basis for important examples of dual-context multicategories considered in this thesis, specifically multicategories **B** and **H** in chapter 3.

1.2.3 Extending basic dual-context calculi

The language of simple dual-context calculi is very basic. It can only describe terms which are either variables or applications. We can extend it with different type constructors, like products and disjoint sums. Functions and exponential types are not covered in this thesis and remain one of the possible directions of future work.

We will perform the extensions mainly for the $\mathbb{III}(\Sigma)$ calculus since this calculus is particularly interesting for us because of its relation with the system $\mathfrak{B}$.

We start by extending the $\mathbb{III}(\Sigma)$ calculus with products. In the dual-context setting several different variants of products can be defined. We consider *normal* and *safe* products and provide a systematic way of building a sound multicategorical interpreta-

tion for these two extensions. In the similar fashion we extend $\mathbb{III}(\Sigma)$ with two different variants of disjoint sums and provide a sound multicategorical interpretation for them.

Then we shall consider an extension of the $\mathbb{III}(\Sigma)$ calculus with the dual-context safe list type with an elimination rule describing safe recursion of Bellantoni and Cook. We will show the polynomial time computable functions on natural numbers can be defined using such an extension of the $\mathbb{III}(\Sigma)$ calculus. In this way we will build a typed variant of Bellantoni and Cook's system $\mathfrak{B}$. A sound multicategorical interpretation of safe dual-context list type will be given using the notion of *safe list object.*

The safe recursion of system $\mathfrak{B}$ can be regarded as one of many different possibilities of extending the primitive recursion scheme to a dual-context setting. Bellantoni and Cook showed that the restrictions of the safe recursion scheme were sufficient in order to avoid the definability of non-polytime computable functions. Is this just a coincidence or does safe recursion have deep fundamental significance? In the case of primitive recursion, it has been long known that it is not an ad-hoc definition mechanism, but one that arises naturally in several different contexts. For example in logic, the primitive recursive functions are exactly the definable functions in several dialects of arithmetic, including $I\Sigma_1$ (arithmetic with induction over Σ^0_1 formulae). In category theory, which is a quite different setting, a simple parameterized version of Lawvere's notion of natural number object again yields exactly the primitive recursive functions in categories with finite products [LS86, Rom89]. The notion of parameterized natural number object can also be defined for categories with monoidal product [RP89].

The account of primitive recursion through parameterized natural number object is particularly attractive. Parameterized natural number objects can be recast as parameterized initial algebras for the strong endofunctor $\mathbf{1} + (-)$ which suggests a general method of adapting primitive recursion to other algebraic datatypes, as parameterized initial algebras for other endofunctors. This route has been followed extensively, for example in [CS92, CS95, Hag87a, Hag87b, Jac95].

To end the thesis, we consider a similar possibility for safe recursion. We define a notion of safe endofunctor on a dual-context multicategory and we show how safe list object can be viewed as a suitable kind of initial algebra for an appropriate strong endofunctor. Thus safe recursion does potentially generalise naturally to other endofunctors and hence other data-types.

1.2.4 Chapter Outline

This section provides a brief outline of content of remaining chapters:

- In Chapter 2 we define basic-dual-context calculi $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$ and study their structural properties. Then we present a framework for describing dual-context calculi, which consists of the $\mathbb{LL}(\Sigma)$ calculus and renamings. We show how equivalent formulations of the $\mathbb{III}(\Sigma)$ and the $\mathbb{IIL}(\Sigma)$ calculi can be obtained in this framework.

- In Chapter 3 we develop semantics of the basic dual-context systems defined in the previous chapter. We start with modifying the linear/non-linear models of DILL to give sound interpretation of the $\mathbb{III}(\Sigma)$ calculus. We then define notions of II-multicategory and IL-multicategory and give sound and complete interpretations of $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$. We present examples of II- and IL-multicategories, which are built out of polytime computable functions, satisfying some additional conditions. After this we develop a notion of *R-multicategory*, which generalizes II- and IL-multicategories.

- In Chapter 4 we extend $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$ with products, sums and terminal objects. In the dual-context setting these constructions have several variations, for which we give sound interpretation using dual-context multicategories with additional structure.

- In Chapter 5 we extend $\mathbb{III}(\Sigma)$ with a safe binary list datatype and provide a sound interpretation of this extension in an II-multicategory equipped with a safe binary list object. We show that every function from the system $\mathfrak{B}$ can be represented in an II-multicategory with safe binary list object and that the polytime computable functions are exactly the normal functions representable in every II-multicategory with safe list object.

- In Chapter 6 we define the notion of strong endofunctor on II-multicategory. Then a notion of the safe initial algebra for such endofunctors is considered. We show that such safe initial algebra generalises the notion of safe binary list object.

Chapter 2

Basic dual-context type systems

In this chapter we present a syntactic framework which describes a wide class of dual-context type systems. We start with the basic calculus of terms in which weakening and contraction is allowed for both the safe and the normal variables. Then we consider a modified variant in which weakening and contraction are allowed only for the normal variables. Finally we present a dual-context calculus with renamings which can be instantiated to obtain an equivalent formulations many different dual-context calculi.

2.1 Dual-typed signatures and dual-contexts

This section contains preliminary syntactic definitions which will be used in dual-context type systems. We start with dual-typed many-sorted signatures.

A standard many-sorted signature Σ contains a set of sort symbols $|\Sigma|$ and a set function symbols each of which is assigned a type of the form $\sigma_1 \dots \sigma_n \to \tau$, where $\sigma_1, \dots, \sigma_n$ and τ are sorts from $|\Sigma|$.

We shall be working instead with function symbols with types of the form

$$\sigma_1 \dots \sigma_n \,;\, \tau_1 \dots \tau_m \to \rho$$

Such function symbols will be called 'dual-typed'.

Definition 2.1.1 (Dual-typed many-sorted signature)**.**
A **dual-typed many-sorted signature** Σ is a pair (S, F) where S is a set of symbols called **basic types** and F is a set of dual-typed function symbols

$$f : \sigma_1 \dots \sigma_n \,;\, \tau_1 \dots \tau_m \to \rho$$

where $\sigma_1, \dots, \sigma_n, \tau_1, \dots, \tau_m, \rho$ are basic types from S.

Remark 2.1.2. Since all the signatures used in this thesis are going to be dual-typed many-sorted we shall refer to them as just 'signatures' dropping the 'dual-typed many-sorted' part.

We now assume a countably infinite set of variables *Var* ranged over by $x, y, \ldots$. Given a signature Σ, a **typing** is a pair $x : \sigma$ where $x \in Var$ is a variable and $\sigma \in |\Sigma|$ is a basic type from Σ.

A **dual-context** (d-context for short) is a pair of *sets* of typings which will be written as $[x_1 : \sigma_1, \ldots, x_n : \sigma_n ; y_1 : \tau_1, \ldots, y_m : \tau_m]$ subject to the condition that any variable can appear in a d-context in at most one typing. D-contexts will be ranged over using capital Greek letters such as $\Gamma ; \Delta$ or $\Pi ; \Theta$. Quite often we shall use just the abbreviation $\Gamma ; \Delta$ assuming that it stands for the d-context $[x_1 : \sigma_1, \ldots, x_n : \sigma_n ; y_1 : \tau_1, \ldots, y_m : \tau_m]$.

The variables $x_1, \ldots, x_n$ from the left part of a d-context will be called **normal** and the variables $y_1, \ldots, y_m$ will be called **safe**. A d-context without safe variables will be called a **normal d-context**. We shall use the notation $\Gamma(x)$ for the type of a normal variable x from the d-context $\Gamma ; \Delta$ and $\Delta(y)$ for the type of a safe variable y.

Given two d-contexts with disjoint sets of variables

$$\Gamma_1 ; \Delta_1 = [v_1 : \sigma_1, \ldots, v_n : \sigma_n ; u_1 : \tau_1, \ldots, u_m : \tau_m]$$
$$\Gamma_2 ; \Delta_2 = [x_1 : \rho_1, \ldots, x_k : \rho_k ; y_1 : \xi_1, \ldots, y_l : \xi_l]$$

we can union them together to get a new d-context

$$\Gamma_1, \Gamma_2 ; \Delta_1, \Delta_2 = [v_1 : \sigma_1, \ldots, v_n : \sigma_n, x_1 : \rho_1, \ldots, x_k : \rho_k ; u_1 : \tau_1, \ldots, u_m : \tau_m, y_1 : \xi_1, \ldots, y_l : \xi_l]$$

We shall use the notation $\Gamma_1, \ldots, \Gamma_n ; \Delta_1, \ldots, \Delta_n$ for denoting the union of n mutually disjoint d-contexts.

Given a signature Σ we define **pre-terms** over Σ as follows:

$$t ::= x \mid f(t_1, \ldots, t_n ; u_1, \ldots, u_m)$$

where $t_1, \ldots, t_n, u_1, \ldots, u_m$ are pre-terms, x is a variable and $f \in \Sigma$ is a function symbol with type $\sigma_1, \ldots, \sigma_n ; \tau_1, \ldots, \tau_m \to \rho$.

We can substitute variables in a pre-term with other pre-terms using simultaneous substitution denoted by $t[s_1/x_1, \ldots, s_k/x_k]$, where $t, s_1, \ldots, s_k$ are pre-terms and

$x_1,\ldots,x_k$ are distinct variables. Substitutions will be abbreviated using small Greek letters α,β,σ. Simultaneous substitution is defined as follows:

$$t[\alpha] = \begin{cases} x & \text{if } t = x \text{ and } x \neq x_i \text{ for every } 0 \leq i \leq n \\ s_i & \text{if } t = x_i \\ f(u_1[\alpha],\ldots,u_k[\alpha]\,;\, v_1[\alpha],\ldots,v_l[\alpha]) & \text{if } t = f(u_1,\ldots,u_k\,;\, v_1,\ldots,v_l) \end{cases}$$

where $\alpha = [s_1/x_1,\ldots,s_k/x_k]$.

Since we do not have any binding constructions any variable in a pre-term is free. We write $\mathsf{FV}(t)$ for the set of free variables.

Lemma 2.1.3. *The simultaneous substitution satisfies the following properties:*

$$t[\,] \equiv t$$
$$t[\alpha, x/x] \equiv t[\alpha]$$
$$t[\alpha, u/x] \equiv t[\alpha] \quad \textit{if } x \notin \mathit{FV}(t)$$
$$t[u_1/x_1,\ldots,u_n/x_n][\alpha] \equiv t[u_1[\alpha]/x_1,\ldots,u_n[\alpha]/x_n] \quad \textit{if } \mathit{FV}(t) = \{x_1,\ldots,x_n\}$$

Proof. Induction on the structure of t. □

2.2 Basic dual-context calculus $\mathbb{III}(\Sigma)$

In this section we shall present the basic dual-context calculus $\mathbb{III}(\Sigma)$ for a given signature Σ and study its properties. A typing judgement of $\mathbb{III}(\Sigma)$ will have the following form:

$$\Gamma\,;\,\Delta \vdash t : \sigma$$

where $\Gamma\,;\,\Delta$ is a d-context over Σ, t is a pre-term over Σ and $\sigma \in |\Sigma|$ is a basic type.

Definition 2.2.1.
The rules of $\mathbb{III}(\Sigma)$ are as follows:

$$\frac{}{\Gamma\,;\,\Delta, y:\tau \vdash y:\tau}\ \textbf{II-S-Ax} \qquad \frac{}{\Gamma, x:\sigma\,;\,\Delta \vdash x:\sigma}\ \textbf{II-N-Ax}$$

$$\frac{\begin{array}{c}\text{for any symbol } f:\sigma_1\ldots\sigma_n\,;\,\tau_1\ldots\tau_m\to\rho\in\Sigma \\ \Gamma\,;\vdash t_1:\sigma_1 \quad \ldots \quad \Gamma\,;\vdash t_n:\sigma_n \\ \Gamma\,;\,\Delta\vdash s_1:\tau_1 \quad \ldots \quad \Gamma\,;\,\Delta\vdash s_m:\tau_m\end{array}}{\Gamma\,;\,\Delta\vdash f(t_1,\ldots,t_n\,;\,s_1,\ldots,s_m):\rho}\ \textbf{II-Sort}$$

A pre-term t is an $\mathbb{III}(\Sigma)$-**term** of type σ in a d-context $\Gamma;\Delta$ if $\Gamma;\Delta\vdash t:\sigma$ can be derived in $\mathbb{III}(\Sigma)$. Any $\mathbb{III}(\Sigma)$-terms of the form $\Gamma;\vdash t:\sigma$ will be called **normal**.

Note the following important features of the **II-Sort** rule:

- The terms $t_1,\ldots,t_n$ are required to be normal.
- For a function symbol $f:\sigma_1,\ldots,\sigma_n;\to\rho$ the corresponding **II-Sort** rule is

$$\frac{\Gamma;\vdash t_1:\sigma_1 \quad \ldots \quad \Gamma;\vdash t_n:\sigma_n}{\Gamma;\Delta\vdash f(t_1,\ldots,t_n;):\rho}$$

  where Δ is arbitrary such that $\Gamma;\Delta$ is a valid d-context. Similarly for a function symbol f with the type $;\to\rho$ the **II-Sort** rule looks as follows

$$\frac{}{\Gamma;\Delta\vdash f(;):\rho}$$

  where $\Gamma;\Delta$ is an arbitrary d-context. Normally we shall treat such a zero-argument function as a constant and write simply f instead of $f(;)$.

The following proposition lists two important properties of the $\mathbb{III}(\Sigma)$ calculus. The first one is the unique derivation property which enables us to drop any distinctions between an $\mathbb{III}(\Sigma)$-term and it's derivation. The second property states that every free variable in an $\mathbb{III}(\Sigma)$-term must be typed in the d-context of that term. Since we have an implicit weakening built into the **II-Sort** rule we can encounter situations when a variable appears in the d-context $\Gamma;\Delta$ but not among free variables of the term t such that $\Gamma;\Delta\vdash t:\sigma$. Also we may see multiple appearances of the same variable in an $\mathbb{III}(\Sigma)$-term.

Proposition 2.2.2.

- *Any typing judgement* $\Gamma;\Delta\vdash t:\sigma$ *which can be derived in* $\mathbb{III}(\Sigma)$ *has a unique derivation.*
- *If a judgement* $\Gamma;\Delta\vdash t:\sigma$ *can be derived in* $\mathbb{III}(\Sigma)$*, then every free variable of* t *appears in the d-context* $\Gamma;\Delta$

Proof. Induction on the structure of the pre-term t. □

We shall now investigate the structural rules, which are admissible in the $\mathbb{III}(\Sigma)$ calculus. We start with typing rules for substitution since many other structural rules can be seen as special cases of substitution.

In the dual-context setting we have two kinds of substitutions. We can substitute for a safe variable as follows:

$$\frac{\Gamma;\Delta\vdash t:\sigma \qquad \Gamma;y:\sigma,\Delta\vdash s:\tau}{\Gamma;\Delta\vdash s[t/y]:\tau}$$

This substitution rule is admissible as we shall see below. On the other hand, the following substitution rule for a normal variable is not admissible in $\mathbb{III}(\Sigma)$:

$$\frac{\Gamma;\Delta\vdash t:\sigma \qquad \Gamma,x:\sigma;\Delta\vdash s:\tau}{\Gamma;\Delta\vdash s[t/x]:\tau}$$

The reason for that is the restrictions we pointed out in the **II-Sort** rule. Only the following restricted substitution is admissible for normal variables:

$$\frac{\Gamma;\vdash t:\sigma \qquad \Gamma,x:\sigma;\Delta\vdash s:\tau}{\Gamma;\Delta\vdash s[t/x]:\tau}$$

Such 'unary' substitution rules are consequences of the following more general 'simultaneous' substitution rule:

$$\frac{\begin{array}{c}x_1:\sigma_1,\ldots,x_n:\sigma_n;y_1:\tau_1,\ldots,y_m:\tau_m\vdash t:\rho\\ \Gamma;\vdash u_1:\sigma_1 \qquad \ldots \qquad \Gamma;\vdash u_n:\sigma_n\\ \Gamma;\Delta\vdash v_1:\tau_1 \qquad \ldots \qquad \Gamma;\Delta\vdash v_m:\tau_m\end{array}}{\Gamma;\Delta\vdash t[u_1/x_1,\ldots,u_n/x_n,v_1/y_1,\ldots,v_m/y_m]:\rho}\ \textbf{II-Subst}$$

This rule has two special cases which are worth mentioning. The first one is when the term t is normal. In this case the **II-Subst** rule looks as follows:

$$\frac{\begin{array}{c}x_1:\sigma_1,\ldots,x_n:\sigma_n;\vdash t:\rho\\ \Gamma;\vdash u_1:\sigma_1 \qquad \ldots \qquad \Gamma;\vdash u_n:\sigma_n\end{array}}{\Gamma;\Delta\vdash t[u_1/x_1,\ldots,u_n/x_n]:\rho}$$

where Δ is arbitrary such that $\Gamma;\Delta$ is a valid d-context. Another special case is when both n and m are equal to 0:

$$\frac{;\vdash t:\rho}{\Gamma;\Delta\vdash t:\rho}$$

where $\Gamma;\Delta$ is any valid d-context.

Comparing the substitution typed by the **II-Subst** rule with the safe composition scheme from the Bellantoni and Cook's system $\mathfrak{B}$ (see the definition 1.1.1 on page 7) we can see that they are identical. We shall thus refer to such substitutions as **safe**.

In order to prove the admissibility of the **II-Subst** rule we need the following lemma, which shows that we can perform weakening of safe contexts for normal terms.

Proposition 2.2.3. *If* $\Gamma;\vdash t:\rho$ *then* $\Gamma;\Delta\vdash t:\rho$ *for any* Δ.

Proof. By induction on the derivation of $\Gamma;\vdash t:\rho$:

- The **II-S-Ax** case does not apply and the **II-N-Ax** case is trivial.
- In the **II-Sort** case we have $t = f(u_1,\ldots,u_n;v_1,\ldots,v_m)$. Assume that $m \neq 0$. Then t was derived using the following terms:

$$\begin{array}{llll} \Gamma;\vdash u_1:\sigma_1 & \ldots & \Gamma;\vdash u_n:\sigma_n \\ \Gamma;\vdash v_1:\tau_1 & \ldots & \Gamma;\vdash v_m:\tau_m \end{array}$$

  By applying the induction hypothesis to $v_1,\ldots,v_m$ we get

$$\Gamma;\Delta\vdash v_1:\tau_1 \quad \ldots \quad \Gamma;\Delta\vdash v_m:\tau_m$$

  and by the **II-Sort** rule we get

$$\Gamma;\Delta\vdash f(u_1,\ldots,u_n;v_1,\ldots,v_m):\rho$$

  This proof works also for the case $m = 0$.

□

We can now show the admissibility of the **II-Subst** rule.

Proposition 2.2.4. ***II-Subst*** *is admissible in* $\mathbb{III}(\Sigma)$.

Proof. The proof will be by induction on the derivation of $\Pi;\Theta\vdash t:\rho$ where $\Pi;\Theta = [x_1:\sigma_1,\ldots,x_n:\sigma_n;y_1:\tau_1,\ldots,y_m:\tau_m]$. We shall abbreviate the substitution as follows:

$$\alpha = u_1/x_1,\ldots,u_n/x_n$$
$$\beta = v_1/y_1,\ldots,v_m/y_m$$

- In the **II-S-Ax** case assume $t = y_j$. By the definition of the substitution $y_j[\alpha,\beta] = v_j$ and know that $\Gamma;\Delta\vdash v_j:\tau_j$.
- In the **II-N-Ax** case assume $t = x_i$. By the definition of the substitution $x_i[\alpha,\beta] = u_i$. We know that $\Gamma;\vdash u_i:\sigma_i$. Then by Lemma 2.2.3 we get $\Gamma;\Delta\vdash u_i:\sigma_i$.
- In the **II-Sort** case suppose that $t = f(s_1,\ldots,s_l;t_1,\ldots,t_k)$. Assume $k \neq 0$. We have the following $\mathbb{III}(\Sigma)$-terms:

$$\begin{array}{llll} \Pi;\vdash s_1:\lambda_1 & \ldots & \Pi;\vdash s_l:\lambda_l \\ \Pi;\Theta\vdash t_1:\theta_1 & \ldots & \Pi;\Theta\vdash t_k:\theta_k \end{array}$$

Applying the definition of the substitution we have

$$f(s_1,\ldots,s_l\,;t_1,\ldots,t_k)[\alpha,\beta] = f(s_1[\alpha,\beta],\ldots,s_l[\alpha,\beta]\,;t_1[\alpha,\beta],\ldots,t_k[\alpha,\beta])$$

Since $\Gamma\,;\vdash s_i:\lambda_i$ then s_i does not contain variables $y_1,\ldots,y_m$ so $s_i[\alpha,\beta]=s_i[\alpha]$. Using the induction hypothesis we get

$$\begin{array}{llll}\Gamma\,;\vdash s_1[\alpha]:\lambda_1 & \ldots & \Gamma\,;\vdash s_l[\alpha]:\lambda_l\\ \Gamma\,;\Delta\vdash t_1[\alpha,\beta]:\theta_1 & \ldots & \Gamma\,;\Delta\vdash t_k[\alpha,\beta]:\theta_k\end{array}$$

Applying the **II-Sort** rule we get

$$\Gamma\,;\Delta\vdash f(s_1[\alpha],\ldots,s_l[\alpha]\,;t_1[\alpha,\beta],\ldots,t_k[\alpha,\beta]):\rho$$

□

In a sense the **II-Subst** rule describes all the structural rules of our system. This was our main motivation for introducing it instead of usual unary substitution rule. The following proposition shows how other common structural rules can be derived from the **II-Subst** rule:

Proposition 2.2.5. *The following rules are admissible in* $\mathbb{III}(\Sigma)$*:*

$$\frac{\Gamma\,;x:\sigma,\Delta\vdash t:\rho}{\Gamma,x:\sigma\,;\Delta\vdash t:\rho}\ \textbf{Shift}$$

$$\frac{\Gamma\,;\Delta\vdash t:\rho}{\Gamma\,;\Delta,y:\sigma\vdash t:\rho}\ \textbf{S-Weak}\qquad \frac{\Gamma\,;\Delta\vdash t:\rho}{\Gamma,x:\sigma\,;\Delta\vdash t:\rho}\ \textbf{N-Weak}$$

$$\frac{\Gamma\,;\Delta,y:\tau,z:\tau\vdash t:\rho}{\Gamma\,;\Delta,y:\tau\vdash t[y/z]:\rho}\ \textbf{S-Contr}\qquad \frac{\Gamma,x:\sigma,z:\sigma\,;\Delta\vdash t:\rho}{\Gamma,x:\sigma\,;\Delta\vdash t[x/z]:\rho}\ \textbf{N-Contr}$$

Proof. All these rules turn out to be the special cases of the **II-Subst** rule. For each rule we show how the appropriate premises of **II-Subst** rule can be derived either from the premises of the structural rule or the axioms and then show the desired conclusion of the structural rule by application of the **II-Subst** rule. Assume that

$$[\Gamma\,;\Delta] = [x_1:\sigma_1,\ldots,x_n:\sigma_n\,;y_1:\tau_1,\ldots,y_m:\tau_m]$$

- The **Shift** rule allows us to move variable typing from the safe part of a d-context to the normal part. Note that the opposite operation is not admissible.

We are given the term $\Gamma; x : \sigma, \Delta \vdash t : \rho$. By the **II-N-Ax** rule we obtain

$$\Gamma, x : \sigma; \vdash x_i : \sigma_i \quad 1 \leq i \leq n$$
$$\Gamma, x : \sigma; \Delta \vdash x : \sigma$$

By the **II-S-Ax** rule we get

$$\Gamma, x : \sigma; \Delta \vdash y_k : \tau_k \quad 1 \leq k \leq m$$

So by the **II-Subst** rule we get

$$\Gamma, x : \sigma; \Delta \vdash t[x_1/x_1, \ldots, x_n/x_n, x/x, y_1/y_1, \ldots, y_m/y_m] : \rho$$

which is the same as

$$\Gamma, x : \sigma; \Delta \vdash t : \rho$$

- The **S-Weak** rule allows us to introduce a fresh safe variable into the safe part of the context of a term. We are given the term $\Gamma; \Delta \vdash t : \rho$. By applying the **II-S-Ax** and **II-N-Ax** rules we get the following terms:

$$\Gamma; \vdash x_i : \sigma_i \qquad 1 \leq i \leq n$$
$$\Gamma; \Delta, y : \tau \vdash y_k : \tau_k \quad 1 \leq k \leq m$$

We can now apply the **II-Subst** rule and obtain

$$\Gamma; \Delta, y : \tau \vdash t[x_1/x_1, \ldots, x_n/x_n, y_1/y_1, \ldots, y_m/y_m] : \rho$$

In case of empty Δ using the **II-Subst** rule we get

$$\Gamma; y : \tau \vdash t[x_1/x_1, \ldots, x_n/x_n] : \rho$$

If Γ and Δ are empty then we have $; y : \tau \vdash t[] : \rho$ by the **II-Subst** rule.

- The **N-Weak** rule allows us to introduce a fresh normal variable into the normal part of the context of a term. We are given the term $\Gamma; \Delta \vdash t : \rho$. By the **II-S-Ax** and **II-N-Ax** rules we get the following terms:

$$\Gamma, x : \sigma; \vdash x_i : \sigma_i \qquad 1 \leq i \leq n$$
$$\Gamma, x : \sigma; \Delta \vdash y_k : \tau_k \quad 1 \leq k \leq m$$

We can apply the **II-Subst** rule and obtain

$$\Gamma, x : \sigma; \Delta \vdash t[x_1/x_1, \ldots, x_n/x_n, y_1/y_1, \ldots, y_m/y_m] : \rho$$

If both Γ and Δ are empty we can still use the **II-Subst** rule to get the term $x : \sigma; \vdash t[] : \rho$.

- The **S-Contr** rule allows us to drop a safe variable from a context of a term, replacing all occurrences of this variable in the term by another safe variable of the same type. We are given the term $\Gamma;\Delta,y:\tau,z:\tau\vdash t:\rho$. By the **II-S-Ax** and **II-N-Ax** rules we get the following terms:

$$\begin{aligned}&\Gamma;\vdash x_i:\sigma_i && 1\leq i\leq n\\ &\Gamma;\Delta,y:\tau\vdash y_k:\tau_k && 1\leq k\leq m\\ &\Gamma;\Delta,y:\tau\vdash y:\tau\end{aligned}$$

  Applying the **II-Subst** rule we get

$$\Gamma;\Delta,y:\tau\vdash t[x_1/x_1,\ldots,x_n/x_n,y_1/y_1,\ldots,y_m/y_m,y/y,y/z]:\rho$$

  which by the properties of the substitution is the same as

$$\Gamma;\Delta,y:\tau\vdash t[y/z]:\rho$$

- The **N-Contr** rule allows us to drop a normal variable from a context of a term, replacing all occurrences of this variable in the term by another normal variable of the same type. We are given the term $\Gamma,x:\sigma,z:\sigma;\Delta\vdash t:\rho$. Using the **II-S-Ax** and **II-N-Ax** rules we get the following terms:

$$\begin{aligned}&\Gamma,x:\sigma;\vdash x_i:\sigma_i && 1\leq i\leq n\\ &\Gamma,x:\sigma;\vdash x:\sigma\\ &\Gamma,x:\sigma;\Delta\vdash y_k:\tau_k && 1\leq k\leq m\end{aligned}$$

  We can apply the **II-Subst** and get the term

$$\Gamma,x:\sigma;\Delta\vdash t[x_1/x_1,\ldots,x_n/x_n,x/x,x/z,y_1/y_1,\ldots,y_m/y_m]:\rho$$

  which by the properties of the substitution is the same as

$$\Gamma,x:\sigma;\Delta\vdash t[x/z]:\rho$$

□

We have shown that weakening and contraction are admissible in $\mathbb{III}(\Sigma)$ for the both parts of d-context. This explains our choice of name for $\mathbb{III}(\Sigma)$, where $\mathbb{III}$ stands for 'Intuitionistic-Intuitionistic' and describes how two parts of dual-contexts are treated in this system. In the next section we shall define another basic dual-context calculus, in which weakening and contraction will be allowed only for the normal variables and it will be called $\mathbb{IIL}(\Sigma)$ ('Intuitionistic-Linear').

Remark 2.2.6. We have not said anything about the exchange rules which allow permutations of typings in a dual-context. Since our dual-contexts are pairs of sets of typings we get permutations of safe and normal typings for free. If we were to define dual-contexts as pairs of sequences for example as Barber does in [Bar96b] then the exchange rules would still be admissible in the $\mathbb{III}(\Sigma)$ calculus.

The basic dual-context calculus we have defined so far has the following features:

1. Two kinds of variables in terms, which are typed using dual-contexts.
2. Restricted substitution - only normal terms can be substituted for normal variables.
3. Weakening and contraction are available for both normal and safe variables.

Our goal is to define a general syntactic framework for describing a general dual-context calculus with restricted substitution but with a different set of admissible structural rules. We have seen that the weakening rule and the contraction rule are the special cases of the **II-Subst** rule. Now we are going to consider an alternative notion, which also can be used for describing structural rules:

Definition 2.2.7 (II-Renaming)**.**
Given a pair of d-contexts

$$[\Gamma\,;\Delta] = [x_1 : a_1, \ldots, x_n : a_n\,; y_1 : b_1, \ldots, y_m : b_m]$$
$$[\Pi\,;\Theta] = [u_1 : c_1, \ldots, u_k : c_k\,; v_1 : d_1, \ldots, v_l : d_l]$$

an **II-renaming** ε is a function between the sets of variables

$$\varepsilon : \{x_1, \ldots, x_n, y_1, \ldots, y_m\} \to \{u_1, \ldots, u_k, v_1, \ldots, v_l\}$$

which satisfies the following properties:

- $\varepsilon(x_i) \in \Pi$ and $\Gamma(x_i) = \Pi(\varepsilon(x_i))$ (normal variables are mapped by ε to normal variables with the same types)
- $\varepsilon(y_j) \in \Theta$ and $\Delta(y_j) = \Theta(\varepsilon(x_i))$ (safe variables are mapped to safe variables with the same types)

We shall denote renaming as $\varepsilon : [\Gamma\,;\Delta] \to [\Pi\,;\Theta]$.

Note that several different variables can be mapped to the same variables and some variables may have no variables mapped onto them.

Given an $\mathbb{III}(\Sigma)$-term $\Gamma\,;\Delta \vdash t : \rho$ and an II-renaming $\varepsilon : [\Gamma\,;\Delta] \to [\Pi\,;\Theta]$ an application of ε to t will be denoted as $\Pi\,;\Theta \vdash t \cdot \varepsilon : \rho$ where

$$t \cdot \varepsilon = t[\varepsilon(x_1)/x_1, \ldots, \varepsilon(x_n)/x_n, \varepsilon(y_1)/y_1, \ldots, \varepsilon(y_m)/y_m]$$

For every II-renaming $\varepsilon : [\Gamma\,;\Delta] \to [\Pi\,;\Theta]$ we consider the following typing rule:

$$\frac{\Gamma\,;\Delta \vdash t : \rho}{\Pi\,;\Theta \vdash t \cdot \varepsilon : \rho}\ \mathbf{Ren}(\varepsilon)$$

Different structural rules can be described as instances of the $\mathbf{Ren}(\varepsilon)$ rule for a particular II-renaming. Consider the safe weakening rule for example:

$$\frac{x_1 : \sigma_1, \ldots, x_n : \sigma_n\,;\, y_1 : \tau_1, \ldots, y_m : \tau_m \vdash t : \rho}{x_1 : \sigma_1, \ldots, x_n : \sigma_n\,;\, y_1 : \tau_1, \ldots, y_m : \tau_m, y : \tau \vdash t : \rho}$$

It can be described by the following inclusion renaming:

$$\kappa : \{x_1, \ldots, x_n, y_1, \ldots, y_m\} \to \{x_1, \ldots, x_n, y_1, \ldots, y_m, y\}$$

Proposition 2.2.8. *For any II-renaming ε the $\mathbf{Ren}(\varepsilon)$ rule is admissible in $\mathbb{III}(\Sigma)$.*

Proof. By the **II-N-Ax** rule we have $\Pi\,;\vdash \varepsilon(x_i) : \sigma_i$ since $\alpha(x_i) \in \Pi$ and α respects types. Similarly by the **II-S-Ax** rule we have $\Pi\,;\Theta \vdash \varepsilon(y_j) : \tau_j$. Using the **II-Subst** rule we get $\Pi\,;\Theta \vdash t \cdot \varepsilon : \rho$. The cases with an empty Δ and an empty $[\Gamma\,;\Delta]$ are similar. □

Since renamings are functions we can compose them and get an II-renaming. Also an identity map is obviously an II-renaming:

Proposition 2.2.9. *The set of all II-renamings has the following properties:*

- *Given two II-renamings*

  $$\alpha : [\Gamma_1\,;\Delta_1] \to [\Gamma_2\,;\Delta_2] \qquad \beta : [\Gamma_2\,;\Delta_2] \to [\Gamma_3\,;\Delta_3]$$

  their composition, defined as function composition

  $$\beta \circ \alpha : [\Gamma_1\,;\Delta_1] \to [\Gamma_3\,;\Delta_3]$$

  is also an II-renaming.

- *The identity mapping between on a d-context $[\Gamma\,;\Delta]$ is an II-renaming.*

Proof. Straightforward check. □

Renamings will be used in section 4 of this chapter as a part of a framework for describing a dual-context calculus with safe substitution and an arbitrary set of structural rules.

The $\mathbb{III}(\Sigma)$ calculus can be equipped with a basic equational logic. Such logic will be simple since $\mathbb{III}(\Sigma)$ does not have any type constructors.

Definition 2.2.10.
An **equation-in-context** over a signature Σ can be written as $\Gamma\,;\Delta \vdash t_1 = t_2 : \sigma$ where t_1, t_2 are $\mathbb{III}(\Sigma)$-terms of type σ in the dual-context $\Gamma\,;\Delta$. A dual-context **algebraic theory** *Th* for a signature Σ consists of a set of equations-in-context over Σ, which are called the **axioms** of *Th*. The set of **theorems** of a theory *Th* consists of all equations-in-context which can be proved from the axioms of *Th* using the following rules:

$$\frac{}{\Gamma\,;\Delta \vdash t = t : \sigma}\ \textbf{Ref} \qquad \frac{\Gamma\,;\Delta \vdash t_1 = t_2 : \sigma}{\Gamma\,;\Delta \vdash t_2 = t_1 : \sigma}\ \textbf{Sym}$$

$$\frac{\Gamma\,;\Delta \vdash t_1 = t_2 : \sigma \qquad \Gamma\,;\Delta \vdash t_2 = t_3 : \sigma}{\Gamma\,;\Delta \vdash t_1 = t_3 : \sigma}\ \textbf{Trans}$$

$$\frac{\begin{array}{c}\Gamma\,;\Delta \vdash p_1 = p_2 : \rho \\ \Pi\,;\vdash u_1 = v_1 : \sigma_1 \quad \ldots \quad \Pi\,;\vdash u_n = v_n : \sigma_n \\ \Pi\,;\Theta \vdash s_1 = t_1 : \tau_1 \quad \ldots \quad \Pi\,;\Theta \vdash s_m = t_m : \tau_m\end{array}}{\Pi\,;\Theta \vdash p_1[\alpha] = p_2[\beta] : \rho}\ \textbf{II-Sub}$$

where α abbreviates the substitution $u_1/x_1, \ldots, u_n/x_n, s_1/y_1, \ldots, s_m/y_m$ and β abbreviates the substitution $v_1/x_1, \ldots, v_n/x_n, t_1/y_1, \ldots, t_m/y_m$.

The rules **Ref**, **Sym** and **Trans** assert that the equality $=$ is a reflexive, symmetric and transitive relation. The rule **II-Sub** asserts the safe substitution preserves equality.

In this section we have presented a simple dual-context calculus of terms built using dual-typed functional symbols from a signature Σ. The admissible substitution in this calculus is restricted allowing only normal terms to be substituted for normal variables. We have shown that weakening and contraction for both the safe and the normal variables are admissible in $\mathbb{III}(\Sigma)$. In the second half of this thesis we shall extend this calculus with inductive data-types and relate it to the system $\mathfrak{B}$.

2.3 Basic dual-context calculus $\mathbb{IL}(\Sigma)$

In this section we shall present another variant of simple dual-context calculus of terms built using functional symbols from a signature Σ. This calculus will be called $\mathbb{IL}(\Sigma)$ ('Intuitionistic-Linear') and will have the admissible substitution rule restricted in the same way as the substitution in the $\mathbb{IL}(\Sigma)$ calculus. One major difference between $\mathbb{IL}(\Sigma)$ and $\mathbb{III}(\Sigma)$ will be that only normal weakening and contraction rules will be admissible in $\mathbb{IL}(\Sigma)$.

The $\mathbb{IL}(\Sigma)$ calculus essentially described the structural core of the system DILL from [Bar96b]. We present it here for a number of reasons. Firstly, Barber does not consider this system explicitly, defining instead a generalised version with separate intuitionistic and linear types. Secondly, we believe that this calculus is interesting as it can be potentially used as a foundation for a dual-context reformulation of the polytime type systems of Hofmann [Hof99] and Schwichtenberg [SB02]. Thirdly, comparing this calculus with $\mathbb{III}(\Sigma)$ helps us to understand better how the subtle differences in the typing rules lead to different sets of admissible structural rules. This will be utilised in the next section in which we consider a framework for describing a general dual-context calculus with safe substitution.

The development of the $\mathbb{IL}(\Sigma)$ calculus follows closely that of $\mathbb{III}(\Sigma)$ so we shall be brief and only concentrate on the differences between the two.

Definition 2.3.1.
Given a signature Σ, the $\mathbb{IL}(\Sigma)$ calculus has the following typing rules:

$$\frac{}{\Gamma; y:\tau \vdash y:\tau}\ \textbf{IL-S-Ax} \qquad \frac{}{\Gamma, x:\sigma; \vdash x:\sigma}\ \textbf{IL-N-Ax}$$

$$\frac{\begin{array}{c}\text{for any symbol } f:\sigma_1\ldots\sigma_n;\tau_1\ldots\tau_m\to\rho\in\Sigma \\ \Gamma;\vdash t_1:\sigma_1 \quad \ldots \quad \Gamma;\vdash t_n:\sigma_n \\ \Gamma;\Delta_1\vdash s_1:\tau_1 \quad \ldots \quad \Gamma;\Delta_m\vdash s_m:\tau_m\end{array}}{\Gamma;\Delta_1,\ldots,\Delta_m\vdash f(t_1,\ldots,t_n;s_1,\ldots,s_m):\rho}\ \textbf{IL-Sort}$$

where $\Delta_1,\ldots,\Delta_m$ are mutually disjoint.

Definition 2.3.2 ($\mathbb{IL}(\Sigma)$-Term)**.**
A pre-term t is a $\mathbb{IL}(\Sigma)$-**term** of type σ in a d-context $[\Gamma;\Delta]$ if $\Gamma;\Delta\vdash t:\sigma$ can be derived in $\mathbb{IL}(\Sigma)$.

Let us compare these typing rules with the typing rules of $\mathbb{III}(\Sigma)$ given in definition 2.2.1 on page 18. Note that in the **IL-S-Ax** rule the dual-context in the conclusion

contains only one safe variable and that in the **IL-N-Ax** rule the term x is normal. The **IL-Sort** rule is still 'safe' in the sense that only normal terms $t_1,\ldots,t_n$ can be used when constructing the term $f(t_1,\ldots,t_n\,;s_1,\ldots,s_m)$. Comparing **IL-Sort** with **II-Sort** we see that safe parts of d-contexts in the terms $s_1,\ldots s_m$ are no longer shared.

For a function symbol with normal arguments only $f:\sigma_1,\ldots,\sigma_n\,;\to\rho$ the **IL-Sort** rule looks as follows:

$$\frac{\Gamma\,;\vdash t_1:\sigma_1 \qquad \ldots \qquad \Gamma\,;\vdash t_n:\sigma_n}{\Gamma\,;\vdash f(t_1,\ldots,t_n\,;):\rho}$$

Comparing this rule with the corresponding case of the **II-Sort** rule we can see that there is no implicit weakening with an arbitrary Δ. For a constant function symbol the **IL-Sort** rule looks as follows:

$$\frac{}{\Gamma\,;\vdash f(\,;):\rho}$$

where $\Gamma\,;$ is an arbitrary normal d-context. So in this case **IL-Sort** contains implicit normal weakening.

$\mathbb{IL}(\Sigma)$ also has the unique derivation property and the free variable property. In addition every safe variable from a d-context must appear exactly once in the term, thus no repetition or dropping of safe variables is allowed.

Proposition 2.3.3. $\mathbb{IL}(\Sigma)$ *calculus has the following properties:*

- ***Unique derivation property***
  If t is an $\mathbb{IL}(\Sigma)$-term of type σ in a d-context $[\Gamma\,;\Delta]$ then there is a unique derivation of $\Gamma\,;\Delta\vdash t:\sigma$ in $\mathbb{IL}(\Sigma)$.

- ***Free variable property***
  If $\Gamma\,;\Delta\vdash t:\sigma$ then every variable from $FV(t)$ is typed in $[\Gamma\,;\Delta]$.

- ***Safe variable property***
  If $\Gamma\,;\Delta\vdash t:\sigma$ then every variable $y\in\Delta$ belongs to $FV(t)$ and appears exactly once in the term t

Proof. By induction on the structure of t. □

Simultaneous substitution in $\mathbb{IL}(\Sigma)$ can be typed by the following rule:

$$\frac{\begin{array}{c}x_1:\sigma_1,\ldots,x_n:\sigma_n\,;y_1:\tau_1,\ldots,y_m:\tau_m\vdash t:\rho\\ \Gamma\,;\vdash u_1:\sigma_1 \qquad \ldots \qquad \Gamma\,;\vdash u_n:\sigma_n\\ \Gamma\,;\Delta_1\vdash v_1:\tau_1 \qquad \ldots \qquad \Gamma\,;\Delta_m\vdash v_m:\tau_m\end{array}}{\Gamma\,;\Delta_1,\ldots,\Delta_m\vdash t[u_1/x_1,\ldots,u_n/x_n,v_1/y_1,\ldots,v_m/y_m]:\rho}\ \textbf{IL-Subst}$$

where $\Delta_1,\ldots,\Delta_m$ are mutually disjoint. Comparing it with the **II-Subst** rule on page 20 we see terms $v_1,\ldots v_m$ do not share safe parts of d-contexts. In the case when m is equal to zero the **IL-Subst** rule has the following form:

$$\frac{x_1:\sigma_1,\ldots,x_n:\sigma_n;\vdash t:\rho \qquad \Gamma;\vdash u_1:\sigma_1 \quad \ldots \quad \Gamma;\vdash u_n:\sigma_n}{\Gamma;\vdash t[u_1/x_1,\ldots,u_n/x_n]:\rho}$$

and when both m are n are equal to zero then the **IL-Subst** rule looks as follows:

$$\frac{;\vdash t:\rho}{\Gamma;\vdash t:\rho}$$

Proposition 2.3.4. *The **IL-Subst** rule is admissible in* $\mathbb{IL}(\Sigma)$.

Proof. By induction on the derivation of t. Denote the substitution by $[\alpha,\beta]$.

- The **IL-S-Ax** and **IL-N-Ax** cases are trivial. Note that we no longer need lemma for weakening of normal terms.
- In the **IL-Sort** the term t is $f(p_1,\ldots,p_k;s_1,\ldots,s_l)$. Using the induction hypothesis we obtain $p_1[\alpha],\ldots,p_k[\alpha]$ and $s_1[\alpha,\beta_1],\ldots,s_l[\alpha,\beta_l]$ where $\beta=\beta_1,\ldots,\beta_l$. Then by the applying the **IL-Sort** rule we get $\Gamma;\Delta\vdash t[\alpha,\beta]:\rho$.

□

The **IL-Subst** rule can be used to derive a number of usual structural rules. Some of the rules which were admissible in $\mathbb{III}(\Sigma)$ can no longer be derived because of the differences between the two calculi:

Proposition 2.3.5. *The following structural rules are admissible in* $\mathbb{IL}(\Sigma)$*:*

$$\frac{\Gamma;\Delta\vdash t:\rho}{\Gamma,x:\sigma;\Delta\vdash t:\rho}\ \textbf{N-Weak} \qquad \frac{\Gamma,x:\sigma,z:\sigma;\Delta\vdash t:\rho}{\Gamma,x:\sigma;\Delta\vdash t[x/z]:\rho}\ \textbf{N-Contr}$$

$$\frac{\Gamma;x:\sigma,\Delta\vdash t:\rho}{\Gamma,x:\sigma;\Delta\vdash t:\rho}\ \textbf{Shift}$$

Proof. Similar to the corresponding proof in the $\mathbb{III}(\Sigma)$ calculus. □

The fact that $\mathbb{IL}(\Sigma)$ has fewer admissible structural rules than $\mathbb{III}(\Sigma)$ is reflected in the following notion of IL-renaming, which satisfies stronger properties compared with II-renaming: renamings.

Definition 2.3.6 (IL-Renaming).
Given a pair of d-contexts

$$[\Gamma;\Delta] = [x_1 : a_1, \ldots, x_n : a_n ; y_1 : b_1, \ldots, y_m : b_m]$$
$$[\Pi;\Theta] = [u_1 : c_1, \ldots, u_k : c_k ; v_1 : d_1, \ldots, v_l : d_l]$$

an **IL-renaming** ε is a function between the sets of variables

$$\varepsilon : \{x_1, \ldots, x_n, y_1, \ldots, y_m\} \to \{u_1, \ldots, u_k, v_1, \ldots, v_l\}$$

which satisfies the following properties:

- $\varepsilon(x_i) \in \Pi$ and $\Gamma(x_i) = \Pi(\varepsilon(x_i))$ (normal variables are mapped by ε to normal variables with the same types)
- $\varepsilon(y_j) \in \Theta$ and $\Delta(y_j) = \Theta(\varepsilon(x_i))$ (safe variables are mapped to safe variables with the same types)
- ε restricted to $y_1, \ldots, y_m$ is a bijection (m=l and every safe variable is mapped to exactly one safe variable)

Proposition 2.3.7. *For every IL-renamings* ε *the* **Ren**(ε) *rule is admissible in* $\mathbb{IL}(\Sigma)$.

Proof. **Ren**(ε) is a special case of **IL-Subst**. □

Basic equational logic for $\mathbb{IL}(\Sigma)$ is similar to one for $\mathbb{II}(\Sigma)$ given in definition 2.2.10 on page 27. The only difference is that the **II-Subst** rule is replaced with the following appropriately modified rule:

$$\frac{\begin{array}{c}\Gamma;\Delta \vdash p_1 = p_2 : \rho \\ \Pi;\vdash u_1 = v_1 : \sigma_1 \quad \ldots \quad \Pi;\vdash u_n = v_n : \sigma_n \\ \Pi;\Theta_1 \vdash s_1 = t_1 : \tau_1 \quad \ldots \quad \Pi;\Theta_m \vdash s_m = t_m : \tau_m\end{array}}{\Pi;\Theta_1,\ldots,\Theta_m \vdash p_1[\alpha] = p_2[\beta] : \rho} \textbf{IL-Subst}$$

where α abbreviates the substitution $u_1/x_1, \ldots, u_n/x_n, s_1/y_1, \ldots, s_m/y_m$ and β abbreviates the substitution $v_1/x_1, \ldots, v_n/x_n, t_1/y_1, \ldots, t_m/y_m$.

In this section we have presented simple dual-context calculus $\mathbb{IL}(\Sigma)$ with safe substitution and normal weakening and contraction. We have seen how the changes in axioms and sort rule affect the set of admissible structural rules.

2.4 Basic dual-context calculi with safe substitution

In the previous sections we have introduced two basic dual-context calculi $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$. The main difference between these two systems was in the sets of admissible structural rules as a result of different typing rules.

In this section we shall consider another approach to defining a dual-context calculus with safe substitution. It is based on the idea of adding weakening and contraction rules directly rather than showing that such rules are admissible. We start with a minimal dual-context calculus in which both kinds of variables are treated in the linear fashion. Such a calculus can then be extended with structural rules which are described by a set of renamings. By considering different sets of renamings we can define different dual-context calculi in this way.

The minimal dual-context calculus will be called $\mathbb{LL}(\Sigma)$. It will be developed similarly to $\mathbb{III}(\Sigma)$ with some appropriate modifications. Typing rules of $\mathbb{LL}(\Sigma)$ are given below:

Definition 2.4.1.
The $\mathbb{LL}(\Sigma)$ calculus has the following typing rules:

$$\frac{}{;\, y:\tau \vdash y:\tau}\ \textbf{LL-S-Ax} \qquad \frac{}{x:\sigma;\, \vdash x:\sigma}\ \textbf{LL-N-Ax}$$

$$\frac{\begin{array}{c}\text{for any symbol } f:\sigma_1 \ldots \sigma_n;\, \tau_1 \ldots \tau_m \to \rho \in \Sigma \\ \Pi_1;\, \vdash t_1:\sigma_1 \quad \ldots \quad \Pi_n;\, \vdash t_n:\sigma_n \\ \Gamma_1;\, \Delta_1 \vdash s_1:\tau_1 \quad \ldots \quad \Gamma_m;\, \Delta_m \vdash s_m:\tau_m\end{array}}{\Gamma;\, \Delta \vdash f(t_1,\ldots,t_n;\, s_1,\ldots,s_m):\rho}\ \textbf{LL-Sort}$$

where $\Gamma;\, \Delta = \Pi_1,\ldots,\Pi_n,\Gamma_1,\ldots,\Gamma_m;\, \Delta_1,\ldots,\Delta_m$ is a union of mutually disjoint dual-contexts $[\Pi_1;\,],\ldots,[\Pi_n;\,]$ and $[\Gamma_1;\, \Delta_1],\ldots,[\Gamma_m,;\, \Delta_m]$.

One may wish to compare the rules of $\mathbb{LL}(\Sigma)$ with the rules for $\mathbb{III}(\Sigma)$ given in the definition 2.2.1 on page 18 and the rules for $\mathbb{IIL}(\Sigma)$ from the definition 2.3.1 on page 28 and note that both normal and safe variables are treated in $\mathbb{LL}(\Sigma)$ in the linear fashion.

Proposition 2.4.2. $\mathbb{LL}(\Sigma)$ *calculus has the following properties:*

- ***Unique derivation property***
  If t is an $\mathbb{LL}(\Sigma)$-term of type σ in a d-context $[\Gamma;\, \Delta]$ then there is a unique derivation of $\Gamma;\, \Delta \vdash t:\sigma$ in $\mathbb{LL}(\Sigma)$.

- ***Free variable property***
  If $\Gamma;\Delta\vdash t:\sigma$ *then every variable from* $FV(t)$ *is typed in* $[\Gamma;\Delta]$.

- ***Normal variable property***
  If $\Gamma;\Delta\vdash t:\sigma$ *then every normal variable* $x\in\Gamma$ *belongs to* $FV(t)$ *and appears exactly once in the term* t

- ***Safe variable property***
  If $\Gamma;\Delta\vdash t:\sigma$ *then every safe variable* $y\in\Delta$ *belongs to* $FV(t)$ *and appears exactly once in the term* t

Proof. By induction on the structure of t. □

The admissible simultaneous substitution rule in $\mathbb{LL}(\Sigma)$ looks as follows:

$$\frac{\begin{array}{c}x_1:\sigma_1,\ldots,x_n:\sigma_n;y_1:\tau_1,\ldots,y_m:\tau_m\vdash t:\rho\\ \Gamma_1;\vdash u_1:\sigma_1\quad\ldots\quad\Gamma_n;\vdash u_n:\sigma_n\\ \Pi_1;\Delta_1\vdash v_1:\tau_1\quad\ldots\quad\Pi_m;\Delta_m\vdash v_m:\tau_m\end{array}}{\Gamma;\Delta\vdash t[u_1/x_1,\ldots,u_n/x_n,v_1/y_1,\ldots,v_m/y_m]:\rho}\ \textbf{LL-Subst}$$

where $\Gamma;\Delta=\Gamma_1,\ldots,\Gamma_n,\Pi_1,\ldots,\Pi_m;\Delta_1,\ldots,\Delta_m$. Note that **LL-Subst** does not include implicit safe or normal weakening in the case of n or m being zero. The only admissible structural rule which follows from the **LL-Subst** rule is the following familiar rule:

$$\frac{\Gamma;x:\sigma,\Delta\vdash t:\rho}{\Gamma,x:\sigma;\Delta\vdash t:\rho}\ \textbf{Shift}$$

The basic equational logic of $\mathbb{LL}(\Sigma)$ is similar to the logics of $\mathbb{II}(\Sigma)$ with the following rule replacing the **II-Subst** rule:

$$\frac{\begin{array}{c}\Gamma;\Delta\vdash p_1=p_2:\rho\\ \Pi_1;\vdash u_1=v_1:\sigma_1\quad\ldots\quad\Pi_n;\vdash u_n=v_n:\sigma_n\\ \Omega_1;\Theta_1\vdash s_1=t_1:\tau_1\quad\ldots\quad\Omega_m;\Theta_m\vdash s_m=t_m:\tau_m\end{array}}{\Pi_1,\ldots,\Pi_n,\Omega_1,\ldots,\Omega_m;\Theta_1,\ldots,\Theta_m\vdash p_1[\alpha]=p_2[\beta]:\rho}\ \textbf{LL-Sub}$$

where α abbreviates the substitution $u_1/x_1,\ldots,u_n/x_n,s_1/y_1,\ldots,s_m/y_m$ and β abbreviates the substitution $v_1/x_1,\ldots,v_n/x_n,t_1/y_1,\ldots,t_m/y_m$.

Essentially, the $\mathbb{LL}(\Sigma)$ calculus is the minimal dual-context calculus with safe substitution. We will now consider extensions of this calculus with renaming rules for every II-renaming $\varepsilon:[\Gamma;\Delta]\to[\Pi;\Theta]$:

$$\frac{\Gamma;\Delta\vdash t:\rho}{\Pi;\Theta\vdash t\cdot\varepsilon:\rho}\ \textbf{Ren}(\varepsilon)$$

and show that the resulting calculus is equivalent to $\mathbb{III}(\Sigma)$. This extended calculus will be called $\mathbb{IIIR}(\Sigma)$. The similar result would hold for $\mathbb{IIL}(\Sigma)$ if we extend $\mathbb{LLL}(\Sigma)$ with renaming rules for every IL-renaming.

The basic equational logic of $\mathbb{IIIR}(\Sigma)$ is the same as logic of $\mathbb{LLL}(\Sigma)$ extended with the following rule:

$$\frac{\Gamma\,;\Delta\vdash t_1=t_2:\rho}{\Pi\,;\Theta\vdash t_1\cdot\varepsilon=t_2\cdot\varepsilon:\rho}\ \textbf{Ren}$$

Theorem 2.4.3. *An equation-in-context* $\Gamma\,;\Delta\vdash t=s:\rho$ *can be proved in* $\mathbb{III}(\Sigma)$ *iff it can be proved in* $\mathbb{IIIR}(\Sigma)$.

Proof. We start with $\mathbb{III}(\Sigma)$ to $\mathbb{IIIR}(\Sigma)$ direction and show that $\Gamma\,;\Delta\vdash t:\rho$ can be derived in $\mathbb{IIIR}(\Sigma)$ by induction on the derivation in $\mathbb{III}(\Sigma)$:

- In the **II-S-Ax** case we have $\Gamma\,;\Delta,y:\tau\vdash y:\tau$. By the **LL-S-Ax** we can show $;y:\tau\vdash y:\tau$. Consider a inclusion II-renaming $\varepsilon:[;y:\tau]\to[\Gamma\,;\Delta,y:\tau]$. Using the **Ren** rule for such a renaming we get $\Gamma\,;\Delta,y:\tau\vdash y:\tau$.

- The **II-N-Ax** case is similar to the **II-S-Ax** case.

- In the **II-Sort** rule apply renaming rules to get

$$\begin{array}{lll}\Gamma_1\,;\vdash t_1:\sigma_1 & \ldots & \Gamma_n\,;\vdash t_n:\sigma_n\\ \Pi_1\,;\Delta_1\vdash s_1:\tau_1 & \ldots & \Pi_m\,;\Delta_m\vdash s_m:\tau_m\end{array}$$

  where $\Pi_1,\ldots,\Pi_m,\Gamma_1,\ldots,\Gamma_n$ are mutually disjoint copies of Γ and $\Delta_1,\ldots,\Delta_m$ are mutually disjoint copies of Δ. Then using the **LL-Sort** rule we get

$$\Gamma_1,\ldots,\Gamma_n,\Pi_1,\ldots,\Pi_m\,;\Delta_1,\ldots,\Delta_m\vdash f(t_1,\ldots,t_n\,;s_1,\ldots,s_m):\rho$$

  Then we can apply the renaming $[\Gamma_1,\ldots,\Gamma_n,\Pi_1,\ldots,\Pi_m\,;\Delta_1,\ldots,\Delta_m]\to[\Gamma\,;\Delta]$ which maps Γ_i and Π_j to Γ and Δ_j to Δ and get the desired conclusion.

We have shown that both $\Gamma\,;\Delta\vdash t:\rho$ and $\Gamma\,;\Delta\vdash s:\rho$ can be derived in $\mathbb{IIIR}(\Sigma)$. We now prove that $\Gamma\,;\Delta\vdash t=s:\rho$ in $\mathbb{IIIR}(\Sigma)$ by induction on the derivation in $\mathbb{III}(\Sigma)$:

- **Ref**, **Sym** and **Trans** cases are obvious.

- In the **II-Subst** case by induction hypothesis $\Gamma\,;\Delta\vdash p_1=p_2:\rho$ and the following equations-in-context hold in $\mathbb{IIIR}(\Sigma)$:

$$\begin{array}{lll}\Pi\,;\vdash u_1=v_1:\sigma_1 & \ldots & \Pi\,;\vdash u_n=v_n:\sigma_n\\ \Pi\,;\Theta\vdash s_1=t_1:\tau_1 & \ldots & \Pi\,;\Theta\vdash s_m=t_m:\tau_m\end{array}$$

Using the **Ren** rule we can prove that

$$\Pi_1 ; \vdash u_1 = v_1 : \sigma_1 \quad \ldots \quad \Pi_n ; \vdash u_n = v_n : \sigma_n$$
$$\Gamma_1 ; \Theta_1 \vdash s_1 = t_1 : \tau_1 \quad \ldots \quad \Gamma_m ; \Theta_m \vdash s_m = t_m : \tau_m$$

where $\Pi_1, \ldots, \Pi_n, \Gamma_1, \ldots, \Gamma_m$ are mutually disjoint copies of Π and $\Theta_1, \ldots, \Theta_m$ are mutually disjoint copies of Θ. Then using the **LL-Subst** rule we get

$$\Pi_1, \ldots, \Pi_n, \Gamma_1, \ldots, \Gamma_m ; \Theta_1, \ldots, \Theta_m \vdash p_1[\alpha] = p_2[\beta] : \rho$$

and finally we can use the **Ren** rule to get the desired conclusion

$$\Pi ; \Theta \vdash p_1[\alpha] = p_2[\beta] : \rho$$

In the opposite direction we start with showing that $\Gamma ; \Delta \vdash t : \rho$ can be derived in $\mathbb{III}(\Sigma)$ by induction on the derivation of this judgement in $\mathbb{III}\mathbb{R}(\Sigma)$:

- In the **LL-S-Ax** and the **LL-N-Ax** cases the desired conclusion can be obtained using the **II-S-Ax** and the **II-N-Ax** rules.
- In the **Ren** we use proposition 2.2.8 on page 26 showing that **Ren** rule was admissible in $\mathbb{III}(\Sigma)$ for every II-renaming.
- In the **LL-Sort** case we first apply admissible safe and normal weakening rules to get the show the following judgements:

$$\Gamma ; \vdash t_1 : \sigma_1 \quad \ldots \quad \Gamma ; \vdash t_n : \sigma_n$$
$$\Gamma ; \Delta \vdash s_1 : \tau_1 \quad \ldots \quad \Gamma ; \Delta \vdash s_m : \tau_m$$

  where $[\Gamma ; \Delta] = [\Gamma_1, \ldots, \Gamma_n, \Pi_1, \ldots, \Pi_m ; \Delta_1, \ldots, \Delta_m]$. Then by the **II-Sort** we get the desired conclusion.

Showing that $\Gamma ; \Delta \vdash t = s : \rho$ can be proved in $\mathbb{III}(\Sigma)$ can be done similarly. □

In this chapter we have defined two basic dual-context calculi for a signature Σ. In both the $\mathbb{III}(\Sigma)$ and the $\mathbb{IIL}(\Sigma)$ calculi only the safe substitution was admissible. The subtle differences in the typing rules of $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$ lead to a different set of admissible structural rules. In the $\mathbb{IIL}(\Sigma)$ calculus weakening and contraction were admissible only in the case of normal variables.

We then proposed another way of defining basic dual-context calculus. Instead of building the structural rules into the basic rules of the calculus, we defined the minimal

dual-context calculus $\mathbb{LL}(\Sigma)$ and added structural rules explicitly to it. The structural rules were described using the notion of renaming.

We have shown that such a framework was suitable for specifying different basic dual-context calculi with safe substitution. We used renamings which mapped normal variables to normal variables and safe variables to safe variable. It is possible to consider different notion of renaming, for example allowing mapping of safe variables to normal or requiring that some additional constraints hold.

Chapter 3

Dual-context multicategories

In this chapter we define three different notions of dual-context multicategories, which extend the usual notion of multicategory with contexts, separated into two parts. We start with a notion of **II**-multicategory and show how the basic equational logic for the calculus $\mathbb{III}(\Sigma)$presented in the previous chapter can be soundly interpreted in such multicategory. We then consider a notion of IL-multicategory, in which the $\mathbb{IIL}(\Sigma)$ calculus can be soundly interpreted. Finally we present a uniform definition of multicategory with renamings, which generalizes II- and IL-multicategories.

3.1 Categorical semantics of $\mathbb{III}(\Sigma)$

In the previous chapter we defined two basic dual-context calculi $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$. For the $\mathbb{III}(\Sigma)$ calculus we have shown that weakening and contraction rules were admissible for safe and normal variables, while $\mathbb{IIL}(\Sigma)$ had only the normal weakening and contraction rules.

In this section we shall define categorical interpretation of the $\mathbb{III}(\Sigma)$ calculus. First we recall how the interpretation of the type system DILL was defined and then modify it appropriately for the $\mathbb{III}(\Sigma)$ case.

Barber [Bar96b] defined the semantics of DILL using a notion of **linear/non-linear models** first proposed by Benton [Ben95]. A linear/non-linear model consist of:

- a cartesian closed category $(\mathbf{C}, \mathbf{1}, \times, \rightarrow)$
- a symmetric monoidal closed category $(\mathbf{S}, I, \otimes, \multimap)$
- a pair of symmetric monoidal functors $(G,n) : \mathbf{S} \rightarrow \mathbf{C}$ and $(F,m) : \mathbf{C} \rightarrow \mathbf{S}$ which form a symmetric monoidal adjunction $G \vdash F$

Given a linear/non-linear model $(\mathbf{C},\mathbf{S},F,G)$ types of DILL are interpreted as objects of S. Dual-contexts are modeled as objects of $\mathbf{S}$:

$$[\![\Gamma;\Delta]\!] \stackrel{\text{def}}{=} FG(a_1)\otimes FG(a_n)\otimes b_1\otimes\cdots\otimes b_m$$

where a_i is an interpretation of the type σ_i and b_j is an interpretation of the type τ_j. Terms of DILL term are interpreted as morphisms of $\mathbf{S}$ with objects representing dual-contexts as domains:

$$[\![\Gamma;\Delta\vdash t:\rho]\!] \stackrel{\text{def}}{=} [\![\Gamma;\Delta]\!] \xrightarrow{[\![t]\!]} [\![\rho]\!]$$

Barber proved that DILL interpretation defined in such a way is sound an complete.

Benton [Ben95] showed that linear/non-linear models are equivalent to *linear categories*, which consist of a symmetric monoidal closed category $\mathbf{S}$ and symmetric monoidal comonad $! : \mathbf{S}S \to \mathbf{S}$ with two natural transformations $e_a : !a \to I$ and $d_a : !a \to !a\otimes !a$, satisfying several non-trivial conditions.

We will modify linear categories for giving an interpretation of the $\mathbb{III}(\Sigma)$ calculus. First, instead of symmetric monoidal category we take a category with finite products. Second, we do not require cartesian closeness.

Definition 3.1.1.
An **II-model** consists of a category with finite products $\mathbf{C}$, a strict product-preserving endofunctor $N : \mathbf{C} \to \mathbf{C}$ and two natural transformations $\varepsilon : N \Rightarrow \mathbf{I}$ and $\delta : N \Rightarrow N^2$ which satisfy the following conditions:

- (N,ε,δ) is a comonad:

$$\begin{array}{ccc} N(a) & \xrightarrow{\delta_a} & N^2(a) \\ {\scriptstyle \delta_a}\downarrow & & \downarrow{\scriptstyle N(\delta_a)} \\ N^2(a) & \xrightarrow[\delta_{N(a)}]{} & N^3(a) \end{array} \qquad \text{(Com-1)}$$

$$\begin{array}{ccccc} N(a) & & \xrightarrow{id_{N(a)}} & & \\ {\scriptstyle \delta_a}\downarrow & & & & \\ N(a) & \xleftarrow{\varepsilon_{N(a)}\ id_{N(a)}} & N^2(a) & \xrightarrow{N(\varepsilon_a)} & N(a) \end{array} \qquad \text{(Com-2)}$$

- Comonad (N,ε,δ) preserves products:

$$\begin{array}{ccc} N(a_1)\times\cdots\times N(a_n) & \xrightarrow{\varepsilon_{a_1}\times\cdots\times\varepsilon_{a_n}} & a_1\times\cdots\times a_n \\ \Big\downarrow{\scriptstyle N(f)} & & \Big\downarrow{\scriptstyle f} \\ N(b) & \xrightarrow{\varepsilon_b} & b \end{array} \qquad \text{(Com-3)}$$

$$\begin{array}{ccc} N(a_1)\times\cdots\times N(a_n) & \xrightarrow{\delta_{a_1}\times\cdots\times\delta_{a_n}} & N^2(a_1)\times\cdots\times N^2(a_n) \\ \Big\downarrow{\scriptstyle N(f)} & & \Big\downarrow{\scriptstyle N^2(f)} \\ N(b) & \xrightarrow{\delta_b} & N^2(b) \end{array} \qquad \text{(Com-4)}$$

Now we define an interpretation of the basic types and the functional symbols from a signature Σ:

Definition 3.1.2.
Given an II-model (C,N,ε,δ) an **II-structure** L for a signature Σ in this II-model is specified by giving an object $[\![\sigma]\!] \in |\mathbf{C}|$ for each basic type $\sigma \in |\Sigma|$ and for each function symbol $f : \sigma_1,\ldots,\sigma_n ; \tau_1,\ldots,\tau_m \to \rho$ a morphism

$$[\![f]\!] : N([\![\sigma_1]\!])\times\cdots\times N([\![\sigma_n]\!])\times[\![\tau_1]\!]\times\ldots[\![\tau_m]\!] \to [\![\rho]\!]$$

We defined dual-contexts as pairs of sets of typings, so we assume a total order on the variable set *Var* and interpret dual-context as objects of the category $\mathbf{C}$:

$$[\![x_1 : \sigma_1,\ldots,x_n : \sigma_n ; y_1 : \tau_1,\ldots,y_m : \tau_m]\!] \stackrel{\text{def}}{=} N([\![\sigma_1]\!])\times\cdots\times N([\![\sigma_n]\!])\times[\![\tau_1]\!]\times\ldots[\![\tau_m]\!]$$

We shall now define an interpretation of $\mathbb{III}(\Sigma)$-terms as morphisms in an II-model (C,N,ε,δ). We shall denote the composition in C by $\circ$. For the projection morphisms in $\mathbf{C}$ we shall use the notation $\pi_{a_i} : a_1\times\cdots\times a_i\times\cdots\times a_n \to a_i$ and given morphisms $f_1 : c\to a_1,\ldots,f_n : c\to a_n$ we denote the morphism obtained by the universal property of products as $\langle f_1,\ldots,f_n\rangle : c\to a_1\times\cdots\times a_n$.

Definition 3.1.3 (Interpretation of $\mathbb{III}(\Sigma)$-terms)**.**
Given an II-structure L in an II-model (C,N,ε,δ) we define an interpretation of $\mathbb{III}(\Sigma)$-

terms $[\![\Gamma ; \Delta \vdash t : \sigma]\!]$ as follows:

$$[\![\Gamma ; \Delta \vdash x_i : \sigma_i]\!] \stackrel{\text{def}}{=} \varepsilon_{[\![\sigma_i]\!]} \circ \pi_{N([\![\sigma_i]\!])}$$
$$[\![\Gamma ; \Delta \vdash y_j : \tau_j]\!] \stackrel{\text{def}}{=} \pi_{[\![\tau_j]\!]}$$
$$[\![\Gamma ; \Delta \vdash f(t_1, \ldots, t_n ; s_1, \ldots, s_m) : \rho]\!] \stackrel{\text{def}}{=} [\![f]\!] \circ \langle W_\Delta([\![t_1]\!]), \ldots, W_\Delta([\![t_n]\!]), [\![s_1]\!], \ldots, [\![s_m]\!] \rangle$$

where $W_\Delta([\![t]\!])$ is defined as follows:

$$W_\Delta([\![t]\!]) = N([\![t]\!]) \circ (\delta_{[\![\sigma_1]\!]} \times \cdots \times \delta_{[\![\sigma_n]\!]}) \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle$$

We shall now prove that this interpretation is sound with respect to the basic equational logic for $\mathbb{III}(\Sigma)$. In $\mathbb{III}(\Sigma)$ the operation of weakening of normal terms with safe variables played an important role. We start with establishing how such weakening is modeled in our interpretation:

Lemma 3.1.4. *Given a normal term* $\Gamma ; \vdash t : \rho$ *the following holds:*

$$[\![\Gamma ; \Delta \vdash t]\!] = [\![\Gamma ; \vdash t]\!] \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle$$

Proof. By induction on the derivation of $\Gamma ; \vdash t : \rho$.

- The **II-S-Ax** case does not apply.
- In the **II-N-Ax** case by the property of projection morphisms we have

$$[\![\Gamma ; \vdash x_i]\!] \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle = \varepsilon_{[\![\sigma_i]\!]} \circ \pi_{N([\![\sigma_i]\!])} \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle = \varepsilon_{[\![\sigma_i]\!]} \circ \pi_{N([\![\sigma_i]\!])} = [\![\Gamma ; \Delta \vdash x_i]\!]$$

- In the **II-Sort** case by the definition of $[\![\ldots]\!]$ and properties of product and composition we have the following:

$$[\![\Gamma ; \vdash f(t_1, \ldots, t_n ; s_1, \ldots, s_m)]\!] \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle =$$
$$[\![f]\!] \circ \langle W([\![t_1]\!]) \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle, \ldots W([\![t_n]\!]) \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle$$
$$[\![s_1]\!] \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle, \ldots, [\![s_m]\!] \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle \rangle \quad (\star)$$

By the definition of W we have

$$W([\![t_i]\!]) \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle = W_\Delta([\![t_i]\!])$$

Using the induction hypothesis we can show

$$[\![\Gamma ; \vdash s_j]\!] \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle = [\![\Gamma ; \Delta \vdash s_j]\!]$$

Using the above two equalities we can rewrite $(\star)$ as follows

$$[\![f]\!] \circ \langle W_\Delta([\![t_1]\!]), \ldots, W_\Delta([\![t_n]\!]), [\![\Gamma ; \Delta \vdash s_1]\!], \ldots, [\![\Gamma ; \Delta \vdash s_m]\!] \rangle$$

which is equal to $[\![\Gamma ; \Delta \vdash f(t_1, \ldots, t_n ; s_1, \ldots, s_m)]\!]$ by the definition of $[\![\ldots]\!]$.

□

We can now proof the main lemma for the coming soundness theorem. This lemma shows how the safe substitution is interpreted:

Lemma 3.1.5. *Given an term* $\Gamma ; \Delta \vdash t : \rho$ *and terms which can be substituted into it*

$$\begin{array}{llll} \Pi ; \vdash u_1 : \sigma_1 & \ldots & \Pi ; \vdash u_n : \sigma_n \\ \Pi ; \Theta \vdash v_1 : \tau_1 & \ldots & \Pi ; \Theta \vdash v_m : \tau_m \end{array}$$

the following holds:

$$[\![\Pi ; \Theta \vdash t[u_1/x_1, \ldots, u_n/x_n, v_1/y_1, \ldots, v_m/y_m]]\!] = \\ [\![t]\!] \circ \langle W_\Theta([\![u_1]\!]), \ldots, W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle$$

Proof. By induction on the derivation of $\Gamma ; \Delta \vdash t : \rho$. We assume that the context $\Pi ; \Theta$ contains typings $u_1 : a_r, \ldots, u_r : a_r ; v_1 : b_1, \ldots, b_s : r_s$ and abbreviate parts of the substitution as $\alpha = [u_1/x_1, \ldots, u_n/x_n]$ and $\beta = [v_1/y_1, \ldots, v_m/y_m]$.

- The **II-S-Ax** case is trivial.
- In the **II-N-Ax** case we need to show

$$[\![\Pi ; \Theta \vdash u_i]\!] = [\![x_i]\!] \circ \langle W_\Theta([\![u_1]\!]), \ldots, W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle \qquad (\star)$$

  Expanding the definition of $[\![x_i]\!]$ and using the properties of projections the right-hand side of $(\star)$ is equal to:

$$\varepsilon_{[\![\sigma_i]\!]} \circ \pi_{N([\![\sigma_i]\!])} \circ \langle W_\Theta([\![u_1]\!]), \ldots, W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle = \varepsilon_{[\![\sigma_i]\!]} \circ W_\Theta([\![u_i]\!])$$

  Expanding the definition of $W_\Theta([\![u_i]\!])$ we get

$$\varepsilon_{[\![\sigma_i]\!]} \circ N([\![u_i]\!]) \circ (\delta_{[\![a_1]\!]} \times \cdots \times \delta_{[\![a_r]\!]}) \circ \langle \pi_{N([\![a_1]\!])}, \ldots, \pi_{N([\![a_k]\!])} \rangle$$

  Using the Com-3 of ε and δ we get

$$[\![u_i]\!] \circ (\varepsilon_{N(a_1)} \times \cdots \times \varepsilon_{N(a_r)}) \circ (\delta_{[\![a_1]\!]} \times \cdots \times \delta_{[\![a_r]\!]}) \circ \langle \pi_{N([\![a_1]\!])}, \ldots, \pi_{N([\![a_k]\!])} \rangle$$

By the associativity of composition this is equal to

$$[\![u_i]\!] \circ (\varepsilon_{N(a_1)} \circ \delta_{[\![a_1]\!]} \times \cdots \times \varepsilon_{N(a_r)} \circ \delta_{[\![a_r]\!]}) \circ \langle \pi_{N([\![a_1]\!])}, \ldots, \pi_{N([\![a_k]\!])} \rangle$$

By the Com-2 of ε and δ this is equal to $[\![u_i]\!] \circ \langle \pi_{N([\![a_1]\!])}, \ldots, \pi_{N([\![a_k]\!])} \rangle$ which is equal to $[\![\Gamma ; \Delta \vdash u_i]\!]$ by lemma 3.1.4.

- In the **II-Sort** case we need to show

$$[\![f(t_1, \ldots, t_k ; s_1, \ldots, s_l)[\alpha, \beta]]\!] = [\![f(t_1, \ldots, t_k ; s_1, \ldots, s_l)]\!] \circ \langle W_\Theta([\![u_1]\!]), \ldots, W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle \quad \text{(Sort-Sem)}$$

Using the properties of substitution and the definition of $[\![\ldots]\!]$ we can write the left-hand side of (Sort-Sem) as follows:

$$[\![f(t_1, \ldots, t_k ; s_1, \ldots, s_l)[\alpha, \beta]]\!] = [\![f(t_1[\alpha], \ldots, t_k[\alpha] ; s_1[\alpha, \beta], \ldots, s_l[\alpha, \beta])]\!] = [\![f]\!] \circ \langle W_\Theta([\![t_1[\alpha]]\!]), \ldots, W_\Theta([\![t_k[\alpha]]\!]), [\![s_1[\alpha, \beta]]\!], \ldots, [\![s_l[\alpha, \beta]]\!] \rangle$$

By the definition of $[\![\ldots]\!]$ the right-hand side of (Sort-Sem) can be rewritten as follows:

$$[\![f(t_1, \ldots, t_k ; s_1, \ldots, s_l)]\!] \circ \langle W_\Theta([\![u_1]\!]), \ldots, W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle = [\![f]\!] \circ \langle W_\Delta([\![t_1]\!]), \ldots, W_\Delta([\![t_k]\!]), [\![s_1]\!], \ldots, [\![s_l]\!] \rangle \circ \langle W_\Theta([\![u_1]\!]), \ldots, W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle$$

Using the induction hypothesis we have

$$[\![s_i[\alpha, \beta]]\!] = [\![s_i]\!] \circ \langle W_\Theta([\![u_1]\!]), \ldots, W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle$$

Then in order to show (Sort-Sem) we need to prove the following

$$W_\Delta([\![t_i]\!]) \circ \langle W_\Theta([\![u_1]\!]), \ldots, W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle = W_\Theta([\![t_i[\alpha]]\!]) \qquad (*)$$

Using the definition of $W_\Delta([\![t_i]\!])$ the left hand-side of (*) is equal to

$$N([\![t_i]\!]) \circ (\delta_{[\![\sigma_1]\!]} \times \cdots \times \delta_{[\![\sigma_n]\!]}) \circ \langle \pi_{N([\![\sigma_1]\!])}, \ldots, \pi_{N([\![\sigma_n]\!])} \rangle \circ \langle W_\Theta([\![u_1]\!]), \ldots W_\Theta([\![u_n]\!]), [\![v_1]\!], \ldots, [\![v_m]\!] \rangle$$

By the properties of composition and projections this is equal to

$$N([\![t_i]\!]) \circ \langle \delta_{[\![\sigma_1]\!]} \circ W_\Theta([\![u_1]\!]), \ldots, \delta_{[\![\sigma_n]\!]} \circ W_\Theta([\![u_n]\!]) \rangle$$

Using the definition of $W_\Theta([\![t_i[\alpha]]\!])$ the right-hand side of (*) is equal to

$$N([\![t_i[\alpha]]\!]) \circ (\delta_{[\![a_1]\!]} \times \cdots \times \delta_{[\![a_r]\!]}) \circ \langle \pi_{N([\![a_1]\!])}, \ldots, \pi_{N([\![a_r]\!])} \rangle$$

Using the induction hypothesis this is equal to

$$N([\![t_i]\!]) \circ \langle N(W([\![u_1]\!])), \ldots, N(W([\![u_n]\!])) \rangle \circ \langle \delta_{[\![a_1]\!]} \circ \pi_{N([\![a_1]\!])}, \ldots, \delta_{[\![a_r]\!]} \circ \pi_{N([\![a_r]\!])} \rangle$$

Expanding the definition of $W(\ldots)$ and using the fact that N is a product preserving functor we get

$$N([\![t_i]\!]) \circ \langle N^2([\![u_1]\!]) \circ (N(\delta_{[\![a_1]\!]}) \times \cdots \times N(\delta_{[\![a_r]\!]})), \ldots, N^2([\![u_n]\!]) \circ (N(\delta_{[\![a_1]\!]}) \times \ldots \times N(\delta_{[\![a_r]\!]})) \rangle \circ \langle \delta_{[\![a_1]\!]} \circ \pi_{N([\![a_1]\!])}, \ldots, \delta_{[\![a_r]\!]} \circ \pi_{N([\![a_r]\!])} \rangle$$

By the associativity of composition this is equal to

$$N([\![t_i]\!]) \circ \langle N^2([\![u_1]\!]) \circ \langle N(\delta_{[\![a_1]\!]}) \circ \delta_{[\![a_1]\!]} \circ \pi_{N([\![a_1]\!])}, \ldots, N(\delta_{[\![a_r]\!]}) \circ \delta_{[\![a_r]\!]} \circ \pi_{N([\![a_r]\!])} \rangle, \ldots, N^2([\![u_n]\!]) \circ \langle N(\delta_{[\![a_1]\!]}) \circ \delta_{[\![a_1]\!]} \circ \pi_{N([\![a_1]\!])}, \ldots, N(\delta_{[\![a_r]\!]}) \circ \delta_{[\![a_r]\!]} \circ \pi_{N([\![a_r]\!])} \rangle \rangle$$

Using the Com-1 property of ε and δ this is equal to

$$N([\![t_i]\!]) \circ \langle N^2([\![u_1]\!]) \circ \langle \delta_{N([\![a_1]\!])} \circ \delta_{[\![a_1]\!]} \circ \pi_{N([\![a_1]\!])}, \ldots, \delta_{N([\![a_r]\!])} \circ \delta_{[\![a_r]\!]} \circ \pi_{N([\![a_r]\!])} \rangle, \ldots, N^2([\![u_n]\!]) \circ \langle \delta_{N([\![a_1]\!])} \circ \delta_{[\![a_1]\!]} \circ \pi_{N([\![a_1]\!])}, \ldots, \delta_{N([\![a_r]\!])} \circ \delta_{[\![a_r]\!]} \circ \pi_{N([\![a_r]\!])} \rangle \rangle$$

By the associativity of composition this is equal to

$$N([\![t_i]\!]) \circ \langle N^2([\![u_1]\!]) \circ (\delta_{N([\![a_1]\!])} \times \cdots \times \delta_{N([\![a_r]\!])}) \circ \langle \delta_{[\![a_1]\!]} \circ \pi_{N([\![a_1]\!])}, \ldots, \delta_{[\![a_r]\!]} \circ \pi_{N([\![a_r]\!])} \rangle, \ldots, N^2([\![u_n]\!]) \circ (\delta_{N([\![a_1]\!])} \times \cdots \times \delta_{N([\![a_r]\!])}) \circ \langle \delta_{[\![a_1]\!]} \circ \pi_{N([\![a_1]\!])}, \ldots \delta_{[\![a_r]\!]} \circ \pi_{N([\![a_r]\!])} \rangle \rangle$$

Using the Com-4 property of ε and δ this is equal to

$$N([\![t_i]\!]) \circ \langle \delta_{\sigma_1} \circ N([\![u_1]\!]) \circ (\delta_{[\![a_1]\!]} \times \cdots \times \delta_{[\![a_r]\!]}) \circ \langle \pi_{N([\![a_1]\!])}, \ldots, \pi_{N([\![a_r]\!])} \rangle, \ldots, \delta_{\sigma_n} \circ N([\![u_n]\!]) \circ (\delta_{[\![a_1]\!]} \times \cdots \times \delta_{[\![a_r]\!]}) \circ \langle \pi_{N([\![a_1]\!])}, \ldots, \pi_{N([\![a_r]\!])} \rangle \rangle$$

which can be rewritten as $N([\![t_i[\alpha]]\!]) \circ (\delta_{[\![a_1]\!]} \times \cdots \times \delta_{[\![a_r]\!]}) \circ \langle \pi_{N([\![a_1]\!])}, \ldots, \pi_{N([\![a_k]\!])} \rangle$
This proves (*) and (Sort-Sem).

□

We are now ready to state and prove the soundness theorem for the interpretation we defined. We say that an Π-structure L satisfies an equation-in-context $\Gamma;\Delta\vdash t_1 = t_2 : \sigma$ if $[\![\Gamma;\Delta\vdash t_1]\!] = [\![\Gamma;\Delta\vdash t_2]\!]$. Given a dual-context algebraic theory *Th* we say that an Π-structure is a **Th-algebra** if it satisfies all the axioms of *Th*.

Theorem 3.1.6. *Let an Π-structure L in an Π-model (C,N,ε,δ) be a Th-algebra for a dual-context algebraic theory Th. Then L satisfies any theorem of Th:*

$$\Gamma;\Delta\vdash t_1 = t_2 : \sigma \Longrightarrow [\![t_1]\!] = [\![t_2]\!]$$

Proof. By induction on the proof of the equation-in-context $\Gamma;\Delta\vdash t_1 = t_2 : \sigma$

- Axioms of *Th* are satisfied because L is a *Th*-algebra.
- **Ref**, **Com** and **Trans** follow from the properties of equality of morphisms.
- **Π-Subt** follows from the lemma 3.1.5.

□

We have define a sound interpretation of the $\Pi\!\!\Pi(\Sigma)$ calculus. Another important property of an interpretation is completeness with respect to the basic equational logic. Traditionally, completeness is proved by constructing a term model, in which types are objects and terms are morphisms. In our case such term model should have the structure of Π-model. Construction of such term model requires product and exponential types, which are not present in the $\Pi\!\!\Pi(\Sigma)$ calculus. In the next section we propose a different approach to semantics of $\Pi\!\!\Pi(\Sigma)$, for which we provide a completeness proof.

3.2 Dual-context Π-multicategories

In the previous section we defined an interpretation of the $\Pi\!\!\Pi(\Sigma)$ calculus using Π-models and proved the soundness result for it. In this section we propose an alternative approach to the semantics of dual-context systems which we show to be sound and complete.

The main problem of proving completeness of the Π-model interpretation was that the calculus $\Pi\!\!\Pi(\Sigma)$ did not have enough type constructors to obtain a term model using $\Pi\!\!\Pi(\Sigma)$-terms. Such structure was essential for the interpretation of dual-contexts, in particular a comonad was used in the interpretation of normal typings in a d-context.

We propose a different approach to the semantics of $\mathbb{III}(\Sigma)$. We shall introduce dual-contexts directly into our models, instead of modeling dual-contexts using products and comonad structure. The $\mathbb{III}(\Sigma)$ calculus will be interpreted using the notion of a dual-context II-multicategory, which has morphism with dual-contexts as domains. The safe substitution will be interpreted as a composition in such II-multicategory.

The use of multicategories in semantics was pioneered by Lambek [Lam68, Lam69, Lam89]. Leinster and Hermida have used the language of multicategories for expressing the different concepts of higher-dimensional category theory [Lei04, Her00].

In Appendix A we give a definition of multicategory from [Lei04]. The major difference between categories and multicategories is that morphisms in a multicategory have sequences of objects as domains. Such change requires modified composition which is the following operation:

$$\mathbf{M}(a_1 \dots a_n, b) \times \prod_{i=1}^{n} \mathbf{M}(\bar{a}_i, a_i) \to \mathbf{M}(\bar{a}_1 \dots \bar{a}_n, b)$$

where $\mathbf{M}(a_1 \dots a_n, b)$ denotes a set of morphisms from the sequence of objects $a_1 \dots a_n$ to the object b and $\bar{a}_1 \dots \bar{a}_n$ is a concatenation of the sequences $\bar{a}_1 \dots \bar{a}_n$. Lambek [Lam89] used a different unary composition scheme, which required one additional axioms. We believe that using simultaneous composition is more natural.

The axioms of a multicategory are similar to the usual axioms of a category. We have an appropriately modified associativity axioms and two obvious identity axioms. Leinster and Hermida [Lei04, Her00] showed that multicategories are closely related to monoidal categories. More details can be found in Appendix A.

The multicategories can be used in semantics in a similar way as categories. A term $x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash t : \rho$ can be interpreted as a morphism $[\![\sigma_1]\!] \dots [\![\sigma_n]\!] \to [\![\rho]\!]$. Such an approach would be suitable for interpretation of linear calculi but one ingredient is missing. We need some way of interpreting exchange rules, which are usually present in any calculus.

One way of dealing with this issue is to add a symmetric structure to the definition of multicategory, which is given by the following family of maps [Lei04]:

$$\cdot\sigma : \mathbf{M}(a_1 \dots a_n, b) \to \mathbf{M}(a_{\sigma(1)} \dots a_{\sigma(n)}, b)$$

for each permutation σ. Leinster [Lei04] proposed using indexed families rather than sequences as domains of morphisms in a multicategory. In this case we do not need any symmetry maps. Leinster [Lei04] showed that these two approaches are equivalent. More details can be found in Appendix A.

For the interpretation of intuitionistic calculi we need to make more modifications. Composition operation should look as follows:

$$\mathbf{C}(b_1 \dots b_m, c) \times \prod_{i=1}^{m} \mathbf{C}(a_1 \dots a_n, b_i) \to \mathbf{C}(a_1 \dots a_n, c)$$

Note that concatenation of sequences is no longer used. Another change is that instead of usual identity morphisms we require projection morphisms $a_1 \dots a_n \to a_i$. Such multicategories can be related with the cartesian categories. More details can be found in Appendix A.

We are now ready to define multicategories which are suitable for interpreting of dual-context calculi. We shall follow the Leinster's approach and consider morphisms which have dual-contexts as domains, denoted as:

$$[x_1 : a_1, \dots, x_n : a_n \,;\, y_1 : b_1, \dots, y_m : b_m] \to c$$

where $a_1, \dots, a_n$ and $b_1, \dots, b_m$ are object of multicategory. As we did in the previous chapter, in the multicategorical setting we will use the standard abbreviation $[\Gamma \,;\, \Delta]$ to stand for the dual-context $[x_1 : a_1, \dots, x_n : a_n \,;\, y_1 : b_1, \dots, y_m : b_m]$.

Next we need to modify the composition operation. It will be used for interpreting of the safe substitution in $\mathbb{III}(\Sigma)$. Recall that such substitution is typed by the following rule:

$$\frac{\begin{array}{c}\Gamma \,;\, \Delta \vdash t : \rho \\ \Pi \,;\, \vdash u_1 : \sigma_1 \quad \dots \quad \Pi \,;\, \vdash u_n : \sigma_n \\ \Pi \,;\, \Theta \vdash v_1 : \tau_1 \quad \dots \quad \Pi \,;\, \Theta \vdash v_m : \tau_m\end{array}}{\Pi \,;\, \Theta \vdash t[u_1/x_1, \dots, u_n/x_n, v_1/y_1, \dots, v_m/y_m] : \rho}$$

So we need the following operation as composition:

$$\mathbf{M}([\Gamma \,;\, \Delta], c) \times \prod_{i=1}^{n} \mathbf{M}([\Pi \,;\,], a_i) \times \prod_{j=1}^{m} \times \mathbf{M}([\Pi \,;\, \Theta], b_j) \to \mathbf{M}([\Pi \,;\, \Theta], c)$$

We shall use the following notation for denoting composition

$$f \circ_{[\Pi;\Theta]} \langle g_1, \dots, g_n ; h_1, \dots, h_m \rangle$$

omitting the subscript $[\Pi;\Theta]$ when possible. Note that the subscript is essential as m or both n and m can be equal to zeros. In this case we see an implicit weakening built into the composition.

What properties should such composition operation have? One obvious property is the modified associativity, which corresponds to the similar property of the safe substitution:

$$f \circ_{[\Pi;\Theta]} \langle g_1 \circ_{[\Pi;]} \langle \bar{t}\,;\rangle \ldots g_n \circ_{[\Pi;]} \langle \bar{t}\,;\rangle\,;\, h_1 \circ_{[\Pi;\Theta]} \langle \bar{t}\,;\bar{s}\rangle, \ldots h_m \circ_{[\Pi;\Theta]} \langle \bar{t}\,;\bar{s}\rangle\rangle = (f \circ_{[\Gamma;\Delta]} \langle \bar{g}\,;\bar{h}\rangle) \circ_{[\Pi;\Theta]} \langle \bar{t}\,;\bar{s}\rangle$$

Note that such property also covers special cases when implicit weakening is involved.

We also need to modify the identity morphisms. In the dual-context settings we have two kinds of identities, one of which is required to be normal:

$$[x_1 : a_1, \ldots, x_n : a_n\,;\,] \xrightarrow{\pi_{x_i}} a_1 \qquad 1 \le i \le n$$

$$[x_1 : a_1, \ldots, x_n : a_n\,;\, y_1 : b_1, \ldots, y_m : b_m] \xrightarrow{\sigma_{y_j}} b_j \qquad 1 \le j \le m$$

If we compose any multiarrow $[x_1 : a_1, \ldots, x_n : a_n\,;\, y_1 : b_1, \ldots, y_m : b_m] \xrightarrow{f} c$ with the identities the following should hold:

$$f \circ_{[\Pi;\Theta]} \langle \pi_{x_1}, \ldots, \pi_{x_n}\,;\, \sigma_{y_1}, \ldots, \sigma_{y_m}\rangle = f$$

Other properties describing the composition with identities are as follows:

$$\pi_{x_i} \circ_{[\Pi;]} \langle f_1, \ldots, f_n\,;\,\rangle = f_i$$

$$\sigma_{y_j} \circ_{[\Pi;\Theta]} \langle f_1, \ldots, f_n\,;\, g_1, \ldots, g_m\rangle = g_j$$

Combining all the above we give the following definition:

Definition 3.2.1 (**Π**-multicategory)**.**
An **Π-multicategory M** consists of

- a set of objects $\mathbf{M}_0$
- for each object c and each d-context $[\Gamma\,;\,\Delta]$ over $\mathbf{M}_0$ a set $\mathbf{M}([\Gamma\,;\,\Delta], a)$, whose elements will be called multiarrows and denoted as $[\Gamma\,;\,\Delta] \xrightarrow{f} a$
- for each object c, each d-context $[\Gamma\,;\,\Delta]$ and each d-context $[\Pi\,;\,\Theta]$ a composition operation:

$$\mathbf{M}([\Gamma\,;\,\Delta], c) \times \prod_{i=1}^{n} \mathbf{M}([\Pi\,;\,], a_i) \times \prod_{j=1}^{m} \mathbf{M}([\Pi\,;\,\Theta], b_j) \to \mathbf{M}(\Pi\,;\,\Theta, c)$$

- for each d-context $[\Gamma\,;\Delta]$ and each x_i a **normal identity** multiarrow $[\Gamma\,;\,] \xrightarrow{\pi_{x_i}} a_i$ and for each y_j a **safe identity** multiarrow $[\Gamma\,;\Delta] \xrightarrow{\sigma_{y_j}} b_j$

satisfying the following equalities:

- **Associative law**

$$f \circ_{[\Pi;\Theta]} \langle g_1 \circ_{[\Pi;]} \langle \bar{t}\,;\,\rangle \ldots g_n \circ_{[\Pi;]} \langle \bar{t}\,;\,\rangle\,;\, h_1 \circ_{[\Pi;\Theta]} \langle \bar{t}\,;\bar{s}\rangle, \ldots h_m \circ_{[\Pi;\Theta]} \langle \bar{t}\,;\bar{s}\rangle\rangle = (f \circ_{[\Gamma;\Delta]} \langle \bar{g}\,;\bar{h}\rangle) \circ_{[\Pi;\Theta]} \langle \bar{t}\,;\bar{s}\rangle \quad \text{(Assoc)}$$

- **Identity laws**

$$\sigma_{y_j} \circ_{[\Gamma;\Delta]} \langle g_1, \ldots, g_n\,;\, h_1, \ldots, h_m\rangle = h_j \quad \text{(S-Id)}$$

$$\pi_{x_i} \circ_{[\Gamma;]} \langle g_1, \ldots, g_n\,;\,\rangle = g_i \quad \text{(N-Id)}$$

$$f \circ_{[\Gamma;\Delta]} \langle \pi_{x_1}, \ldots, \pi_{x_n}\,;\, \sigma_{y_1}, \ldots, \sigma_{y_m}\rangle = f \quad \text{(Id)}$$

Remark 3.2.2. An alternative definition of II-multicategory with unary composition can be given. One would need two different operations, one for each side of dual-context.

The II-multicategories defined in this section can be related to the II-models we used in the previous section. In Appendix B we outline how every II-model can be represented as II-multicategory. Going in the opposite direction and representing every II-multicategory as II-model requires some additional structure on II-multicategory which we study in the next chapter.

We now consider one important example of an II-multicategory, which will play an important role in this thesis:

Definition 3.2.3.
The II-multicategory **B** is defined as follows:

- The set of objects consists of one element - set of all natural numbers $\mathbb{N}$.

- The set of multiarrows $\mathbf{B}_1([x_1 : \mathbb{N}, \ldots, x_n : \mathbb{N}\,;\, y_1 : \mathbb{N}, \ldots, y_m : \mathbb{N}], \mathbb{N})$ consists of all the functions $f : \mathbb{N}^n \times \mathbb{N}^m \to \mathbb{N}$ such that f is polytime computable and is bounded by a monotone polynomial p_f:

$$|f(\bar{x}\,;\bar{y})| \leq p_f(|\bar{x}|) + \max_j |y_j| \quad \text{(B-Bound)}$$

where $|\_|$ denotes a length of a binary representation of a number. When $|\_|$ is applied to a vector of numbers $a_1 \dots a_n$ it means $\sum_{i=1}^{n} |a_i|$. We shall call such p_f the **size-bounding polynomial**.

- The normal identity multiarrow $\pi_{x_i}(x_1, \dots, x_n ;)$ is given by the projection x_i and the safe identity multiarrow $\sigma_{y_j}(x_1, \dots, x_n ; y_1, \dots, y_m)$ is the projection y_j.
- The composition operation $f \circ_{[\Gamma;\Delta]} \langle g_1, \dots, g_n ; h_1, \dots, h_m \rangle$ is defined using the usual function composition as $f(g_1(\bar{x}), \dots, g_n(\bar{x}), h_1(\bar{x}, \bar{y}), \dots, h_m(\bar{x}, \bar{y}))$.

Proposition 3.2.4. *B is an II-multicategory.*

Proof. The composition is associative since it is based on the usual function composition. If functions f, $\bar{g}$ and $\bar{h}$ are polytime computable then their composition is also polytime computable. Let the bounding polynomials for f, g_i, h_j are p, q_i and r_j respectively. Then the bounding polynomial for the composition can be obtained as follows:

$$
\begin{aligned}
|f \circ \langle g_1, \dots, g_n ; h_1, \dots, h_m \rangle(\bar{x} ; \bar{y})| &\leq p(|\bar{g}|) + \max_j(|h_j|) \\
&\leq p(q_1(|\bar{x}|) + \dots + q_n(|\bar{x}|)) + \max_j(r_j(|\bar{x}|) + \max_j(|y_j|)) \leq \\
&\quad p(q_1(|\bar{x}|) + \dots + q_n(|\bar{x}|)) + r_1(|\bar{x}|) + \dots + r_m(|\bar{x}|) + \max_j(|y_j|
\end{aligned}
$$

Take $p(q_1(x) + \dots + q_n(x)) + r_1(x) + \dots + r_m(x)$ as the bounding polynomial for the composition $f \circ_{[\Pi;\Theta]} \langle \bar{g} ; \bar{h} \rangle$.

The identities π_{x_i} and σ_{y_j} are polytime computable with the bounding polynomials $p_\pi(x) = x$ and $p_\sigma(x) = 0$. The composition and identities satisfy all the desired properties. □

Definition 3.2.5.

Given an II-model $(\mathbf{C}, ! : C \to C)$ define *underlying* II-multicategory $\hat{\mathbf{C}}$ as follow:

- A set of objects of $\hat{\mathbf{C}}$ is the same as $|\mathbf{C}|$.
- For each dual-sequence of objects $a_1 \dots a_n ; b_1 \dots b_m$ and each object c a set of multiarrows

$$\hat{\mathbf{C}}(a_1 \dots a_n ; b_1 \dots b_m, c) = \mathbf{C}(!a_1 \times \dots \times !a_n \times b_1 \times \dots \times b_m, c)$$

- For each dual-sequence of objects $a_1 \dots a_n ; b_1 \dots b_m$ and each object a_i a normal projection multiarrow $\pi_{a_i} = p_{a_i} ; \varepsilon_{a_i}$

- For each dual-sequence of objects $a_1 \dots a_n; b_1 \dots b_m$ and each object b_j a safe projection multiarrow $\sigma_{b_j} = p_{b_j}$

- For each dual-sequence of objects $a_1 \dots a_n; b_1 \dots b_m$, each object c and each dual-sequence of objects $d_1 \dots d_k; e_1 \dots e_l$ a composition operation

$$f \circ \langle g_1, \dots, g_n \,;\, h_1, \dots, h_m \rangle = \langle N(g_1), \dots, N(g_n), h_1, \dots, h_n \rangle; f$$

where $N(g_i) = \langle p_{!a_1}; \delta_{!a_1}, \dots, p_{!a_n}; \delta_{!a_n} \rangle; !g_i$.

Proposition 3.2.6. $\hat{\mathbf{C}}$ *is an II-multicategory.*

Proof. Tedious but routine verification of required conditions.

Another important example of an II-multicategory will be considered in the next section, in which we define a sound and complete interpretation of the $\mathbb{III}(\Sigma)$ calculus using II-multicategories.

3.3 Multicategorical semantics of $\mathbb{III}(\Sigma)$

In the previous section we defined II-multicategories, which have dual-contexts as domains of multiarrows. This feature and the special composition of II-multicategories allow us to construct an interpretation of the $\mathbb{III}(\Sigma)$ calculus and show that it is sound and complete. We shall start with a standard definition, which describes how elements of a signature Σ can be interpreted in an II-multicategory.

Definition 3.3.1.
An **II-structure** $S(\Sigma)$ for a signature Σ in an II-multicategory $\mathbf{M}$ is specified by giving an object $[\![\sigma]\!] \in \mathbf{M}_0$ for each basic type $\sigma \in |\Sigma|$ and a multiarrow

$$[x_1 : [\![\sigma_1]\!], \dots, x_n : [\![\sigma_n]\!] \,;\, y_1 : [\![\tau_1]\!], \dots, y_m : [\![\tau_m]\!]] \xrightarrow{[\![f]\!]} [\![\rho]\!]$$

for each functional symbol $f : \sigma_1, \dots, \sigma_n \,;\, \tau_1, \dots, \tau_m \to \rho$ in Σ.

Given a II-structure in a II-multicategory $\mathbf{M}$ dual-contexts from $\mathbb{III}(\Sigma)$ can be modeled as dual-contexts in $\mathbf{M}$ with types replaced by their interpretations. We will often use the same abbreviations for the dual-contexts and their interpretations.

Each $\mathbb{III}(\Sigma)$-term $\Gamma \,;\, \Delta \vdash t : \rho$ will be interpreted as a multiarrow in $\mathbf{M}$:

$$[x_1 : [\![\sigma_1]\!], \dots, x_n : [\![\sigma_n]\!] \,;\, y_1 : [\![\tau_1]\!], \dots, y_m : [\![\tau_m]\!]] \xrightarrow{[\![t]\!]} [\![\rho]\!]$$

Definition 3.3.2.

Given an II-structure in an II-multicategory $\mathbf{M}$ we define $[\![\Gamma\,;\Delta\vdash t:\sigma]\!]$ as follows:

$$[\![\Gamma\,;\Delta\vdash x_i:\sigma_i]\!] \stackrel{\text{def}}{=} \pi_{x_i}\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle$$

$$[\![\Gamma\,;\Delta\vdash y_j:\tau_j]\!] \stackrel{\text{def}}{=} \sigma_{y_j}$$

$$[\![\Gamma\,;\Delta\vdash f(t_1,\ldots,t_n\,;s_1,\ldots,s_m):\rho]\!] \stackrel{\text{def}}{=} [\![f]\!]\circ_{[\Gamma;\Delta]}\langle[\![t_1]\!],\ldots,[\![t_n]\!]\,;[\![s_1]\!],\ldots,[\![s_m]\!]\rangle$$

We start proving the soundness of such an interpretation with the following lemma, which shows how weakening of normal terms is interpreted:

Lemma 3.3.3. *For any term* $\Gamma\,;\vdash t:\rho$ *the following holds:*

$$[\![\Gamma\,;\Delta\vdash t]\!] = [\![\Gamma\,;\vdash t]\!]\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle$$

Proof. By induction on the derivation of $\Gamma\,;\vdash t:\rho$:

- The **II-S-Ax** case does not apply.
- In the **II-N-Ax** case by the definition of $[\![\ldots]\!]$ we have:

  $$[\![x_i]\!][\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle = (\pi_{x_i}\circ_{[\Gamma;]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle)\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle$$

  which is by the associativity of the composition equal to $\pi_{x_i}\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle$, which is $[\![\Gamma\,;\Delta\vdash x_i]\!]$.

- In the **II-Sort** case by the definition of $[\![\ldots]\!]$ we have

  $$[\![f(u_1,\ldots,u_n\,;v_1,\ldots,v_m)]\!]\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle = ([\![f]\!]\circ_{[\Gamma;]}\langle[\![u_1]\!],\ldots,[\![u_n]\!]\,;\\ [\![v_1]\!],\ldots,[\![v_m]\!]\rangle)\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle$$

  By the associativity of the composition this can be rewritten as

  $$[\![f]\!]\circ_{[\Gamma;\Delta]}\langle[\![u_1]\!]\circ_{[\Gamma;]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle,\ldots,[\![u_n]\!]\circ_{[\Gamma;]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle\,;\\ [\![v_1]\!]\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle,\ldots,[\![v_m]\!]\circ_{[\Gamma;\Delta]}\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\rangle\rangle$$

  By the induction hypothesis and the definition of $[\![\ldots]\!]$ this is equal to

  $$[\![f]\!]\circ_{[\Gamma;\Delta]}\langle[\![u_1]\!],\ldots,[\![u_n]\!]\,;[\![v_1]\!],\ldots,[\![v_m]\!]\rangle = [\![\Gamma\,;\Delta\vdash f(u_1,\ldots,u_n\,;v_1,\ldots,v_m)]\!]$$

□

The next lemma proves that the safe substitution in the $\mathbb{III}(\Sigma)$ can be interpreted using the composition of II-multicategory:

Lemma 3.3.4. *Given the* $\mathbb{III}(\Sigma)$*-term* $\Gamma\,;\Delta \vdash t : \rho$ *and the following* $\mathbb{III}(\Sigma)$*-terms:*

$$\begin{array}{llll} \Pi\,;\vdash u_1 : \sigma_1 & \ldots & \Pi\,;\vdash u_n : \sigma_n \\ \Pi\,;\Theta \vdash v_1 : \tau_1 & \ldots & \Pi\,;\Theta \vdash v_m : \tau_m \end{array}$$

the following holds: $[\![t[\alpha,\beta]]\!] = [\![t]\!] \circ_{[\Pi;\Theta]} \langle [\![u_1]\!],\ldots,[\![u_n]\!]\,;[\![v_1]\!],\ldots,[\![v_m]\!]\rangle$*, where* α *abbreviates* $u_1/x_1,\ldots,u_n/x_n$ *and* β *abbreviates* $v_1/y_1,\ldots,v_m/y_m$*.*

Proof. By induction on the structure of t. We will abbreviate the family $[\![u_1]\!],\ldots,[\![u_n]\!]$ as $[\![\bar{u}]\!]$ and the family $[\![v_1]\!],\ldots,[\![v_m]\!]$ as $[\![\bar{v}]\!]$.

- In the **II-S-Ax** case by the properties of substitution $y_i[\alpha,\beta] = v_i$. On the other hand by the S-Id property and the definition of $[\![\ldots]\!]$ we have

$$[\![y_i]\!] \circ_{[\Pi;\Theta]} \langle [\![\bar{u}]\!]\,;[\![\bar{v}]\!]\rangle = \sigma_i \circ_{[\Pi;\Theta]} \langle [\![\bar{u}]\!]\,;[\![\bar{v}]\!]\rangle = [\![v_i]\!]$$

- In the **II-N-Ax** case by the properties of the substitution $x_i[\alpha,\beta] = u_i$. On the other hand by the definition of $[\![\ldots]\!]$, associativity of the composition and the N-Id property we get:

$$(\pi_{x_i} \circ_{[\Gamma;\Delta]} \langle \pi_{x_1},\ldots,\pi_{x_n}\,;\rangle) \circ_{[\Pi;\Theta]} \langle [\![\bar{u}]\!]\,;[\![\bar{v}]\!]\rangle = \pi_{x_i} \circ_{[\Pi;\Theta]} \langle \pi_{x_1} \circ_{[\Pi;]} \langle [\![\bar{u}]\!]\,;\rangle,\ldots,$$
$$\pi_{x_n} \circ_{[\Pi;]} \langle [\![\bar{u}]\!]\,;\rangle\,;\rangle = \pi_{x_i} \circ_{[\Pi;\Theta]} \langle [\![\bar{u}]\!]\,;\rangle$$

  By the Id property we have $[\![u_i]\!] = [\![u_i]\!] \circ_{[\Pi;]} \langle \pi_{x_1},\ldots,\pi_{x_n}\,;\rangle$ so we have:

$$\pi_{x_i} \circ_{[\Pi;\Theta]} \langle [\![\bar{u}]\!]\,;\rangle = \pi_{x_i} \circ_{[\Pi;\Theta]} \langle [\![u_1]\!] \circ_{[\Pi;]} \langle \pi_{x_1},\ldots,\pi_{x_n}\,;\rangle,\ldots,$$
$$[\![u_n]\!] \circ_{[\Pi;]} \langle \pi_{x_1},\ldots,\pi_{x_n}\,;\rangle\,;\rangle$$

  Using associativity and N-Id this can be written as

$$(\pi_{x_i} \circ_{[\Pi;]} \langle [\![u_1]\!],\ldots,[\![u_n]\!]\,;\rangle) \circ_{[\Pi;\Theta]} \langle \pi_{x_1},\ldots,\pi_{x_n}\,;\rangle = [\![u_i]\!] \circ_{[\Pi;\Theta]} \langle \pi_{x_1},\ldots,\pi_{x_n}\,;\rangle$$

  which by the lemma 3.3.3 is equal to $[\![\Pi\,;\Theta \vdash u_i : \sigma_i]\!]$.

- In the **II-Sort** by the properties of the substitution we have:

$$f(s_1,\ldots,s_k\,;t_1,\ldots,t_l)[\alpha,\beta] = f(s_1[\alpha],\ldots,s_k[\alpha]\,;t_1[\alpha,\beta],\ldots,t_l[\alpha,\beta])$$

On the other hand we have:

$$[\![f(s_1,\ldots,s_k\,;t_1,\ldots,t_l)]\!]\circ_{[\Pi;\Theta]}\langle[\![\bar{u}]\!]\,;[\![\bar{v}]\!]\rangle = ([\![f]\!]\circ_{[\Gamma;\Delta]}\langle[\![s_1]\!],\ldots,[\![s_k]\!]\,;[\![t_1]\!],\ldots,[\![t_l]\!]\rangle)\circ_{[\Pi;\Theta]}\langle[\![\bar{u}]\!]\,;[\![\bar{v}]\!]\rangle$$

which by the associativity of composition is equal to

$$[\![f]\!]\circ_{[\Pi;\Theta]}\langle[\![s_1]\!]\circ_{[\Pi;]}\langle[\![\bar{u}]\!]\,;\rangle,\ldots,[\![s_k]\!]\circ_{[\Pi;]}\langle[\![\bar{u}]\!]\,;\rangle\,;\ [\![t_1]\!]\circ_{[\Pi;\Theta]}\langle[\![\bar{u}]\!]\,;[\![\bar{v}]\!]\rangle,\ldots,[\![t_l]\!]\circ_{[\Pi;\Theta]}\langle[\![\bar{u}]\!]\,;[\![\bar{v}]\!]\rangle\rangle$$

Using the induction hypothesis this can be rewritten as

$$[\![f]\!]\circ_{[\Pi;\Theta]}\langle[\![s_1[\alpha]]\!],\ldots,[\![s_k[\alpha]]\!]\,;[\![t_1[\alpha,\beta]]\!],\ldots,[\![t_l[\alpha,\beta]]\!]\rangle = [\![f(s_1[\alpha],\ldots,s_k[\alpha]\,;t_1[\alpha,\beta],\ldots,t_l[\alpha,\beta])]\!]$$

□

Given an equation-in-context $\Gamma\,;\Delta\vdash t_1 = t_2 : \sigma$ any II-structure $S(\Sigma)$ gives rise to the two multiarrows $[\![t_1]\!],[\![t_2]\!] : [\![\Gamma\,;\Delta]\!]\to[\![\sigma]\!]$. We say that an II-structure $S(\Sigma)$ **satisfies** this equation-in-context if these two multiarrows are equal. If *Th* is a dual-context algebraic theory over a signature Σ then we say that $S(\Sigma)$ is a **Th-algebra** if it satisfies all the axioms of *Th*. The following theorem shows that any *Th*-algebra satisfies all the theorems of *Th*:

Theorem 3.3.5 (Soundness)**.** *Let* $\mathbf{M}$ *be an* $\mathbf{II}$*-multicategory and* $S(\Sigma)$ *be a Th-algebra in* $\mathbf{M}$ *for a dual-context algebraic theory Th. Then* $S(\Sigma)$ *satisfies any theorem of Th*

$$\Gamma\,;\Delta\vdash t_1 = t_2 : \sigma \Longrightarrow [\![t_1]\!] = [\![t_2]\!]$$

Proof. By induction on the derivation of $\Gamma\,;\Delta\vdash t_1 = t_2 : \sigma$

- Axioms of *Th* are satisfied because $S(\Sigma)$ is a *Th*-algebra.
- **Ref**, **Com** and **Trans** follow from the properties of equality of multiarrows in a **II**-multicategory.
- **II-Sub** follows from the lemma 3.3.4.

□

We will now prove the completeness of our multicategorical interpretation of $\mathbb{III}(\Sigma)$. At the heart of such proof is the definition of classifying II-multicategory built out of types and terms of $\mathbb{III}(\Sigma)$.

Assuming that *Th* is a dual-context algebraic theory over the signature Σ we shall write $t_1 =_{Th} t_2$ if $\Gamma;\Delta \vdash t_1 = t_2 : \rho$ is a theorem of *Th*. In this case we say that the terms t_1 and t_2 are **Th-provably equal**. The **Ref**, **Sym** and **Trans** rules of basic equational logic of $\mathbb{III}(\Sigma)$ ensure that the relation $=_{Th}$ is an equivalence relation.

Definition 3.3.6.
Given a signature Σ a **classifying II-multicategory $\mathbf{Cl}_\Sigma$** will be defined as follows:

- The objects of $\mathbf{Cl}_\Sigma$ are the basic types from $|\Sigma|$.
- For each d-context $[\Gamma;\Delta]$ the set of multiarrows $\mathbf{Cl}_\Sigma([\Gamma;\Delta],\rho)$ will be the set of all equivalence classes of $\mathbb{III}(\Sigma)$-terms $\Gamma;\Delta \vdash t : \rho$ under the relation $=_{Th}$. We will denote such equivalence classes as $[\Gamma;\Delta \vdash t : \rho]$.
- Given a multiarrow $[\Gamma;\Delta \vdash t : \rho]$ and a family of multiarrows

$$\begin{array}{lll} [\Pi;\vdash u_1 : \sigma_1] & \ldots & [\Pi;\vdash u_n : \sigma_n] \\ [\Pi;\Theta \vdash v_1 : \tau_1] & \ldots & [\Pi;\Theta \vdash v_m : \tau_m] \end{array}$$

  the composition $t \circ_{[\Pi;\Theta]} \langle u_1,\ldots,u_n ; v_1,\ldots,v_m\rangle$ is defined as equivalence class of the term $\Pi;\Theta \vdash t[u_1/x_1,\ldots,u_n/x_n,v_1/y_1,\ldots,v_m/y_m] : \rho$.
- A safe identity $\sigma_{y_j} : [\Gamma;\Delta] \to \tau_j$ is the equivalence class $[\Gamma;\Delta \vdash y_j : \tau_j]$ and a normal identity $\pi_{x_i} : [\Gamma;\Delta] \to \sigma_i$ is the equivalence class $[\Gamma;\vdash x_i : \sigma_i]$.

Proposition 3.3.7. $\mathbf{Cl}_\Sigma$ *is an II-multicategory.*

Proof. We need to check that all the axioms of II-multicategory are satisfied:

- For the identity laws of II-multicategory we have:

$$\sigma_{y_i} \circ_{[\Gamma;\Delta]} \langle s_1,\ldots,s_n ; t_1,\ldots,t_m\rangle = [y_i[s_1/x_1,\ldots,s_n/x_n,t_1/y_1,\ldots,t_m/y_m]] = [t_i]$$
$$\pi_{x_i} \circ_{[\Gamma;\Delta]} \langle s_1,\ldots,s_n ; \rangle = [x_i[s_1/x_1,\ldots,s_n/x_n,t_1/y_1]] = [s_i]$$
$$t \circ_{[\Gamma;\Delta]} \langle \pi_{x_1},\ldots,\pi_{x_n} ; \sigma_{y_1},\ldots,\sigma_{y_m}\rangle = [t[x_1/x_1,\ldots,x_n/x_n,y_1/y_1,\ldots,y_m/y_m]] = [t]$$

- The composition is associative because of the associativity property of the safe substitution.

□

Next we describe the 'generic' II-structure $\mathfrak{I}$ in Cl_Σ. In $\mathfrak{I}$ each basic type σ is interpreted by the object σ and each function symbol $f : \sigma_1, \ldots, \sigma_n ; \tau_1, \ldots, \tau_m \to \rho$ is interpreted as a multiarrow $[\Gamma ; \Delta \vdash f(x_1, \ldots, x_n ; y_1, \ldots, y_m) : \rho]$.

Proposition 3.3.8. *$\mathfrak{I}$ is a Th-algebra.*

Proof. We need to show that any $\mathbb{III}(\Sigma)$-term $\Gamma ; \Delta \vdash t : \rho$ is interpreted by the equivalence class $[\Gamma ; \Delta \vdash t : \rho]$.

- In the **II-N-Ax** case we have:

$$[\![\Gamma ; \Delta \vdash x_i : \sigma_i]\!] = \pi_{x_i} \circ_{[\Gamma;\Delta]} \langle \pi_{x_1}, \ldots, \pi_{x_n} ; \rangle = [x_i[x_1/x_1, \ldots, x_n/x_n]] = [x_i]$$

- In the **II-S-Ax** case we have:

$$[\![\Gamma ; \Delta \vdash y_j : \tau_j]\!] = \sigma y_j \circ_{[\Gamma;\Delta]} \langle \pi_{x_1}, \ldots, \pi_{x_n} ; \sigma_{y_1}, \ldots, \sigma_{y_m} \rangle = [y_j[x_1/x_1, \ldots, x_n/x_n, y_1/y_1, \ldots, y_m/y_m]] = [y_j]$$

- In the **II-Sort** case we have

$$[\![\Pi ; \Theta \vdash f(s_1, \ldots, s_k ; t_1, \ldots, t_l) : \rho]\!] = [\![f]\!] \circ_{[\Pi;\Theta]} \langle [\![s_1]\!], \ldots, [\![s_k]\!] ; [\![t_1]\!], \ldots, [\![t_l]\!] \rangle = [f(x_1, \ldots, x_k ; y_1, \ldots, y_l)[s_1/x_1, \ldots, s_k/x_k, t_1/y_1, \ldots, t_l/y_l]] = [f(s_1, \ldots, s_k ; t_1, \ldots, t_l)]$$

□

Theorem 3.3.9 (Completeness). *If for the given two $\mathbb{III}(\Sigma)$-terms $\Gamma ; \Delta \vdash t_1 : \rho$ and $\Gamma ; \Delta \vdash t_2 : \rho$ their interpretations in any Th-algebra I are equal $[\![t_1]\!]^I = [\![t_2]\!]^I$ then $\Gamma ; \Delta \vdash t_1 = t_2 : \rho$ is a theorem of Th.*

Proof. Since $[\![t_1]\!]^I = [\![t_2]\!]$ in any *Th*-algebra then it must also hold in $\mathfrak{I}$. From this we can conclude that $\Gamma ; \Delta \vdash t_1 = t_2 : \rho$ is a theorem of *Th*. □

We have shown how the $\mathbb{III}(\Sigma)$ calculus can be interpreted in II-multicategories and proved that this interpretation was sound and complete with respect to the basic equational logic of $\mathbb{III}(\Sigma)$. The construction worked because II-multicategories contained dual-contexts as primitive and products and exponential types were not needed in order to construct the classifying multicategory.

After establishing soundness and completeness of the multicategorical interpretation of the $\mathbb{III}(\Sigma)$ calculus we start using the language of $\mathbb{III}(\Sigma)$ for reasoning about II-multicategories. A multiarrow $[\Gamma\,;\Delta] \xrightarrow{f} c$ can be represented as the $\mathbb{III}(\Sigma)$-term

$$[x_1 : a_1, \ldots, x_n : a_n\,; y_1 : b_1, \ldots, y_m : b_m]\; f(x_1, \ldots, x_n\,; y_1, \ldots, y_m) : c$$

The composition $f \circ_{[\Pi;\Theta]} \langle g_1, \ldots, g_n\,; h_1, \ldots, h_m\rangle$will be represented using substitution by the $\mathbb{III}(\Sigma)$-term

$$[\Pi\,;\Theta]\; f(g_1(\bar{x};), \ldots, g_n(\bar{x};)\,; h_1(\bar{x}\,;\bar{y}), \ldots, h_m(\bar{x}\,;\bar{y}))$$

Identities will be represented by the terms $[\Gamma\,;\Delta]\; x_i$ and $[\Gamma\,;\Delta]\; y_j$.

3.4 IL-multicategories

In the previous section we defined the notion of II-multicategories and used it to give the interpretation of the $\mathbb{III}(\Sigma)$ calculus which was sound and complete. In this section we are going to define a similar notion of IL-multicategories and use it for interpretation of the $\mathbb{IIL}(\Sigma)$ calculus. Much of the development will be very similar to the $\mathbb{III}(\Sigma)$ case so we shall be brief.

The semantics of $\mathbb{III}(\Sigma)$defined in the previous section was essentially based on the following two principles:

1. Dual-contexts are included in the definition of II-multicategories as domains of multiarrows.

2. The safe substitution on $\mathbb{III}(\Sigma)$ corresponds to the composition operation in II-multicategory.

3. The axioms of $\mathbb{III}(\Sigma)$ are interpreted using the identities of an II-multicategory.

The axioms and the safe substitution rule of the $\mathbb{IIL}(\Sigma)$ are different from corresponding has slightly different variants of the substitution and identities:

Definition 3.4.1 (**IL**-multicategory)**.**
An **IL-multicategory M** consists of

- a set of objects $\mathbf{M}_0$

- for each object c and each d-context $[\Gamma\,;\Delta]$ over $\mathbf{M}_0$ a set $\mathbf{M}([\Gamma\,;\Delta], a)$, whose elements will be called multiarrows and denoted as $[\Gamma\,;\Delta] \xrightarrow{f} a$

- for each object c, each d-context $[\Gamma;\Delta]$ and each d-context $[\Pi;\Theta]$ a composition operation:

$$\mathbf{M}([\Gamma;\Delta],c)\times\prod_{i=1}^{n}\mathbf{M}([\Pi;],a_i)\times\prod_{j=1}^{m}\mathbf{M}([\Pi;\Theta_j],b_j)\to\mathbf{M}([\Pi;\Theta_1,\ldots,\Theta_m],c)$$

where $\Theta_1,\ldots,\Theta_m$ are assumed to be mutually disjoint. We shall denote composition as $f\circ\langle g_1,\ldots,g_n;h_1,\ldots,h_m\rangle$. In principle we need to subscript $\circ$ with Π since in the case of f being a multiarrow with the empty d-context as domain, the composition of f with the empty family of multiarrows is a multiarrow with $[\Pi;]$ as domain, where $[\Pi;]$ is an arbitrary normal d-context but we decided to disregard this in favor of lighter notation.

- for each d-context $[\Gamma;]$ and each x_i a **normal identity** multiarrow $[\Gamma;]\xrightarrow{\pi_{x_i}}a_i$ and for each d-context $[\Gamma;y:b]$ a **safe identity** multiarrow $[\Gamma;y:b]\xrightarrow{\sigma_y}b$

satisfying the following equalities:

- **Associative law**

$$f\circ\langle g_1\circ\langle\bar{t};\rangle\ldots g_n\circ\langle\bar{t};\rangle;h_1\circ\langle\bar{t};\bar{s}_1\rangle,\ldots,h_m\circ\langle\bar{t};\bar{s}_m\rangle\rangle=$$
$$(f\circ\langle\bar{g};\bar{h}\rangle)\circ\langle\bar{t};\bar{s}_1,\ldots,\bar{s}_m\rangle \quad \text{(Assoc)}$$

- **Identities laws**

$$\pi_{x_i}\circ\langle g_1,\ldots,g_n;\rangle=g_i \quad \text{(IL-N-Id)}$$
$$\sigma_y\circ\langle g_1,\ldots,g_n;h\rangle=h \quad \text{(IL-S-Id)}$$
$$f\circ\langle\pi_{x_1},\ldots,\pi_{x_n};\sigma_{y_1},\ldots,\sigma_{y_m}\rangle=f \quad \text{(IL-Id)}$$

Given a signature Σ the notion of an IL-structure in an IL-multicategory $\mathbf{M}$ is defined in the same way as II-structure, providing the interpretation of the basic types as objects of $\mathbf{M}$ and function symbols as morphisms in $\mathbf{M}$.

Definition 3.4.2.

Given an IL-structure I in an IL-multicategory $\mathbf{M}$ the interpretation of an $\mathbb{IL}(\Sigma)$-term $\Gamma;\Delta\vdash t:\rho$ is defined as follows:

$$[\![\Gamma;\vdash x_i:\sigma_i]\!]\stackrel{\text{def}}{=}\pi_{x_i}$$
$$[\![\Gamma;y:\tau\vdash y:\tau]\!]\stackrel{\text{def}}{=}\sigma_y$$
$$[\![\Gamma;\Delta_1,\ldots,\Delta_m\vdash f(t_1,\ldots,t_n;s_1,\ldots,s_m):\rho]\!]\stackrel{\text{def}}{=}[\![f]\!]\circ\langle[\![t_1]\!],\ldots,[\![t_n]\!];[\![s_1]\!],\ldots,[\![s_m]\!]\rangle$$

The main lemma for establishing the soundness of this interpretation tells that the safe substitution in $\mathbb{IL}(\Sigma)$ is interpreted by the composition in IL-multicategory:

Lemma 3.4.3. *Given an $\mathbb{IL}(\Sigma)$-term $\Gamma\,;\Delta \vdash t : \rho$ and a family of $\mathbb{IL}(\Sigma)$-terms:*

$$\begin{array}{llll} \Pi\,;\vdash u_1 : \sigma_1 & \ldots & \Pi\,;\vdash u_n : \sigma_n \\ \Pi\,;\Theta_{\vdash} v_1 : \tau_1 & \ldots & \Pi\,;\Theta_m \vdash v_m : \tau_m \end{array}$$

we have the following:

$$[\![t[u_1/x_1,\ldots,u_n/x_n,v_1/y_1,\ldots,v_m/y_m]]\!] = [\![t]\!] \circ \langle [\![u_1]\!],\ldots,[\![u_n]\!]\,;[\![v_1]\!],\ldots,[\![v_m]\!]\rangle$$

Proof. Induction on the structure of t, which is similar to the corresponding proof in the II case. □

Given a dual-context theory *Th* for a signature Σ an IL-structure is a *Th*-algebra if satisfies all the axioms of *Th*.

Theorem 3.4.4 (Soundness)**.** *Let **M** be an **II**-multicategory and $S(\Sigma)$ be a Th-algebra in **M** for a dual-context algebraic theory Th. Then $S(\Sigma)$ satisfies any theorem of Th*

$$\Gamma\,;\Delta \vdash t_1 = t_2 : \sigma \Longrightarrow [\![t_1]\!] = [\![t_2]\!]$$

Proof. By induction on the derivation of $\Gamma\,;\Delta \vdash t_1 = t_2 : \sigma$ with the previous lemma being the only non-trivial case. □

In the opposite direction we have the following completeness theorem:

Theorem 3.4.5 (Completeness)**.** *If for the given two $\mathbb{III}(\Sigma)$-terms $\Gamma\,;\Delta \vdash t_1 : \rho$ and $\Gamma\,;\Delta \vdash t_2 : \rho$ their interpretations in any Th-algebra I are equal $[\![t_1]\!]^I = [\![t_2]\!]^I$ then $\Gamma\,;\Delta \vdash t_1 = t_2 : \rho$ is a theorem of Th.*

Proof. Proof is done by constructing the classifying IL-multicategory out of the basic type and the equivalence classes of terms in the same way as the classifying II-multicategory was built. □

We give one example of IL-multicategory, taken from Hofmann [Hof99]:

Definition 3.4.6 (Multicategory **H**)**.**
The IL-multicategory **H** is defined as follows:

- The set of objects consists of one element - set of all natural numbers $\mathbb{N}$.

- The set of multiarrows $\mathbf{H}_1([x_1 : \mathbb{N}, \ldots, x_n : \mathbb{N}; y_1 : \mathbb{N}, \ldots, y_m : \mathbb{N}], \mathbb{N})$ consists of all the functions $f : \mathbb{N}^n \times \mathbb{N}^m \to \mathbb{N}$ such that f is polytime computable and is bounded by a monotone polynomial p_f:

$$|f(\bar{x}; \bar{y})| \leq p_f(|\bar{x}|) + |\bar{y}| \qquad \text{(H-Bound)}$$

- The normal identity multiarrow $\pi_{x_i}(x_1, \ldots, x_n;)$ is given by the projection x_i and the safe identity multiarrow $\sigma_{y_j}(x_1, \ldots, x_n; y_1, \ldots, y_m)$ is the projection y_j.

- The composition operation $f \circ_\Gamma \langle g_1, \ldots, g_n; h_1, \ldots, h_m \rangle$ is defined using the usual function composition as $f(g_1(\bar{x}), \ldots, g_n(\bar{x}), h_1(\bar{x}, \bar{y}), \ldots, h_m(\bar{x}, \bar{y}))$.

Proposition 3.4.7. *$\boldsymbol{H}$ is an IL-multicategory.*

Proof. The composition is associative since it is based on the usual function composition. If functions f, $\bar{g}$ and $\bar{h}$ are polytime computable then their composition is also polytime computable. Let the bounding polynomials for f, g_i, h_j are p, q_i and r_j respectively. Then the bounding polynomial for the composition can be obtained as follows:

$$|f(g_1, \ldots, g_n, h_1, \ldots, h_m)| \leq p_f(|\bar{g}|) + |\bar{h}| \leq$$
$$p_f(q_1(|\bar{x}|) + \cdots + q_n(|\bar{x}|)) + r_1(|\bar{x}|) + |\bar{y}_1| + \cdots + r_m(|\bar{x}|) + |\bar{y}_m|$$

Take $p(q_1(x) + \cdots + q_n(x)) + r_1(x) + \cdots + r_m(x)$ as the bounding polynomial for the composition. Note that the having IL-style composition is essential in order to obtain the needed bound.

The identities π_{x_i} and σ_{y_j} are polytime computable with the bounding polynomials $p_\pi(x) = x$ and $p_\sigma(x) = 0$. The composition and identities satisfy all the desired properties. □

3.5 R-Multicategories

In the previous sections we have defined the notion of II- and IL-multicategories and have shown how these structures can be used for interpretation of $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$. In this section we shall define a general notion of a dual-context multicategory, which generalizes both II- and IL-multicategories.

We have seen that both $\mathbb{IL}(\Sigma)$ and $\mathbb{III}(\Sigma)$ can be described using the minimal dual-context calculus $\mathbb{LL}(\Sigma)$ extended with appropriate renaming rules. The multicategorical interpretation of such extended $\mathbb{LL}(\Sigma)$ calculus requires the following changed notion of dual-context multicategory:

- Composition will have the following signature:

$$\mathbf{M}([\Gamma\,;\Delta],c)\times\prod_{i=1}^{n}\mathbf{M}([\Lambda_i\,;\,],a_i)\times\prod_{j=1}^{m}\mathbf{M}([\Pi_j\,;\Theta_j],b_j)\to \mathbf{M}([\Lambda_1,\ldots,\Lambda_n,\Pi_1,\ldots,\Pi_m\,;\Theta_1,\ldots,\Theta_m],c)$$

- For identities we take

$$[x:a\,;\,]\xrightarrow{\pi_x}a \qquad\qquad [\,;y:b]\xrightarrow{\sigma_y}b$$

- Composition and identities should satisfy the usual laws:

$$f\circ\langle g_1\circ\langle\bar{t}_1\,;\,\rangle,\ldots,g_n\circ\langle\bar{t}_n\,;\,\rangle\,;h_1\circ\langle\bar{u}_1\,;\bar{s}_1\rangle,\ldots,h_m\circ\langle\bar{u}_m\,;\bar{s}_m\rangle\rangle=$$
$$(f\circ\langle\bar{g}\,;\bar{h}\rangle)\circ\langle\bar{t}_1,\ldots,\bar{t}_n,\bar{u}_1,\ldots,\bar{u}_m\,;\bar{s}_1,\ldots,\bar{s}_m\rangle$$
$$\pi_x\circ\langle g\,;\,\rangle=g$$
$$\sigma_y\circ\langle\,;h\rangle=h$$
$$f\circ\langle\pi_{x_1},\ldots,\pi_{x_n}\,;\sigma_{y_1},\ldots,\sigma_{y_m}\rangle=f$$

- For any renaming $\varepsilon:[\Gamma\,;\Delta]\to[\Pi\,;\Theta]$ in $\mathfrak{R}$ and any object c we need an operation

$$\cdot\varepsilon:\mathbf{M}([\Gamma\,;\Delta],c)\to\mathbf{M}([\Pi\,;\Theta],c)$$

The renaming operation $\cdot\varepsilon$ should satisfy the following laws:

- For the identity renaming $\iota:[\Gamma\,;\Delta]\to[\Gamma\,;\Delta]$ the corresponding renaming operations should be the identity as well:

$$f\cdot\iota=f \qquad \text{(Ren-Id)}$$

- For composable renamings

$$\varepsilon_1:[\Gamma_1\,;\Delta_1]\to[\Gamma_2\,;\Delta_2] \qquad \varepsilon_2:[\Gamma_2\,;\Delta_2]\to[\Gamma_3\,;\Delta_3]$$

the corresponding renaming operations should satisfy

$$f\cdot(\varepsilon_2\circ\varepsilon_1)=(f\cdot\varepsilon_1)\cdot\varepsilon_2 \qquad \text{(Ren-Sup)}$$

- Given a multiarrow $f \in \mathbf{M}([\Gamma\,;\Delta], c)$, where

$$[\Gamma\,;\Delta] = [x_1 : a_1, \ldots, x_n : a_n\,; y_1 : b_1, \ldots, y_m : b_m]$$

which can be composed with the following family:

$$\langle g_{x_1} \cdot \varepsilon_{x_1}, \ldots, g_{x_n} \cdot \varepsilon_{x_n}\,; h_{y_1} \cdot \varepsilon_{y_1}, \ldots, h_{y_m} \cdot \varepsilon_{y_m} \rangle$$

where

$$\begin{aligned} g_{x_i} &: [\Pi_{x_i}\,;\,] \to \Gamma(x_i) & h_{y_j} &: [\Pi_{y_j}\,;\Theta_{y_j}] \to \Delta(y_j) \\ \varepsilon_{x_i} &: [\Pi_{x_i}\,;\,] \to [\Pi'_{x_i}\,;\,] & \varepsilon_{y_j} &: [\Pi_{y_j}\,;\Theta_{y_j}] \to [\Pi'_{y_j}\,;\Theta'_{y_j}] \end{aligned}$$

we require have the following

$$f \circ \langle g_{x_1} \cdot \varepsilon_{x_1}, \ldots, g_{x_n} \cdot \varepsilon_{x_n}\,; h_{y_1} \cdot \varepsilon_{y_1}, \ldots, h_{y_m} \cdot \varepsilon_{y_m} \rangle = (f \circ \langle g_{x_1}, \ldots, g_{x_n}\,; h_{y_1}, \ldots, h_{y_m} \rangle) \cdot \varepsilon \quad \text{(Ren-Comp-1)}$$

where the renaming operation ε corresponds to sum of renamings $\varepsilon_{x_1}, \ldots, \varepsilon_{x_n}$ and $\varepsilon_{y_1}, \ldots, \varepsilon_{y_m}$ such that:

$$\varepsilon : \Pi_{x_1}, \ldots, \Pi_{x_n}, \Pi_{y_1}, \ldots, \Pi_{y_m}\,; \Theta_{y_1}, \ldots, \Theta_{y_m} \to \Pi'_{x_1}, \ldots, \Pi'_{x_n}, \Pi'_{y_1}, \ldots, \Pi'_{y_m}\,; \Theta'_{y_1}, \ldots, \Theta'_{y_m}$$

$$\varepsilon(v) = \begin{cases} \varepsilon_{x_i}(v) & \text{if } v \in [\Pi_{x_i}\,;\,] \\ \varepsilon_{y_j}(v) & \text{if } v \in [\Pi_{y_j}\,;\Theta_{y_j}] \end{cases}$$

- Given a renaming $\varepsilon : [\Gamma\,;\Delta] \to [\Pi\,;\Theta]$

$$\begin{aligned} [\Gamma\,;\Delta] &= [x_1 : a_1, \ldots, x_n : a_n\,; y_1 : b_1, \ldots, y_m : b_m] \\ [\Pi\,;\Theta] &= [u_1 : c_1, \ldots, u_k : c_k\,; v_1 : d_1, \ldots, v_l : d_l] \end{aligned}$$

and a multiarrow $f \in \mathbf{M}([\Gamma\,;\Delta], e)$ and a family of multiarrows composable with $f \cdot \varepsilon$:

$$\begin{aligned} g_{u_i} &: [\Pi_{u_i}\,;\,] \to \Pi(u_i) & 1 \le i \le k \\ h_{v_j} &: [\Pi_{v_j}\,;\Theta_{v_j}] \to \Theta(v_j) & 1 \le j \le l \end{aligned}$$

consider a family

$$\langle g_{\varepsilon(x_1)}, \ldots, g_{\varepsilon(x_n)}\,; h_{\varepsilon(y_1)}, \ldots, g_{\varepsilon(y_n)} \rangle$$

This family can't be composed with f directly because it may contain same multiarrow occurring twice. So we have to apply renamings which will make sure that all the multiarrows have disjoint contexts:

$$g_{x_i} = g_{\varepsilon(x_i)} \cdot \sigma_{x_i} \qquad h_{y_j} = h_{\varepsilon(y_j)} \cdot \sigma_{y_j}$$

where the renamings σ_{x_i} and σ_{y_i} are defined as follows:

$$\sigma_{x_i} : [\Pi_{\varepsilon(x_i)} ;] \to [\Pi_{x_i} ;] \qquad \sigma_{x_i}(v) = v^{x_i}$$
$$\sigma_{y_j} : [\Pi_{\varepsilon(y_j)} ; \Theta_{\varepsilon(y_j)}] \to [\Pi_{y_j} ; \Theta_{y_j}] \qquad \sigma_{y_j}(v) = v^{y_j}$$

We require that

$$(f \cdot \varepsilon) \circ \langle g_{u_1}, \ldots, g_{u_k} ; h_{v_1}, \ldots, h_{v_l} \rangle = (f \circ \langle g_{x_1}, \ldots, g_{x_n} ; h_{y_1}, \ldots, h_{y_m} \rangle) \cdot \sigma \qquad \text{(Ren-Comp-2)}$$

where

$$\sigma : \Pi_{x_1}, \ldots, \Pi_{x_n}, \Pi_{y_1}, \ldots, \Pi_{y_m} ; \Theta_{y_1}, \ldots, \Theta_{y_m} \to \Pi_{u_1}, \ldots, \Pi_{u_k}, \Pi_{v_1}, \ldots, \Pi_{v_l} ; \Theta_{v_1}, \ldots, \Theta_{v_l}$$

$$\sigma(v^{x_i}) = v \in \Pi_{\varepsilon(x_i)}$$

To summarize we have the following definition:

Definition 3.5.1 (R-multicategory)**.**
An **R-multicategory** $\mathbf{M}(\mathfrak{R})$ for a given renaming set $\mathfrak{R}$ consists of

- a set $\mathbf{M}_0$, whose elements are called the **objects** of $\mathbf{M}$
- for each d-context $[\Gamma ; \Delta]$ over $\mathbf{M}_0$ and each object c, a set $\mathbf{M}([\Gamma ; \Delta], c)$
- for each object c, for each d-context

  $$[\Gamma ; \Delta] = [x_1 : a_1, \ldots, x_n : a_n ; y_1 : b_1, \ldots, y_m : b_m]$$

  and each d-context $[\Pi_1, \ldots, \Pi_n, \Phi_1, \ldots, \Phi_m ; \Theta_1, \ldots, \Theta_m]$ a **composition** operation

  $$\mathbf{M}([\Gamma ; \Delta], c) \times \prod_{i=1}^{n} \mathbf{M}([\Pi_1 ;], a_1) \times \prod_{j=1}^{m} \mathbf{M}([\Phi_j ; \Theta_j], b_j) \times \to \mathbf{M}([\Pi_1, \ldots, \Pi_n, \Phi_1, \ldots, \Phi_m ; \Theta_1, \ldots, \Theta_m], c)$$

- for each d-context $[x : a\,;\,]$ a **normal identity** multiarrow

$$[x : a\,;\,] \xrightarrow{\pi_x} a$$

and for each d-context $[\,;\, y_a]$ a **safe identity** multiarrow

$$[\,;\, y : a] \xrightarrow{\sigma_y} a$$

- for every renaming $\sigma : [\Gamma;\Delta] \to [\Pi;\Theta] \in R$ a map

$$\cdot\sigma : M([\Gamma;\Delta], c) \to M([\Pi;\Theta], c)$$

satisfying

- **Associative law**

$$f \circ \langle g_1 \circ \langle \bar{s}_1;\rangle, \ldots, g_n \circ \langle \bar{s}_n\rangle; h_1 \circ \langle \bar{u}_1;\bar{v}_1\rangle, \ldots, h_m \circ \langle \bar{u}_m;\bar{v}_m\rangle\rangle =$$
$$(f \circ \langle g_1, \ldots, g_n; h_1, \ldots, h_m\rangle) \circ \langle \bar{s}_1, \ldots, \bar{s}_n, \bar{u}_1, \ldots, \bar{u}_m; \bar{v}_1, \ldots, \bar{v}_m\rangle \qquad \text{(R-Assoc)}$$

- **Identity laws**

$$f \circ \langle \pi_{x_1}, \ldots, \pi_{x_n}\,;\, \sigma_{y_1}, \ldots, \sigma_{y_m}\rangle = f \qquad \text{(R-Id)}$$
$$\pi_x \circ \langle g\,;\,\rangle = g \qquad \text{(R-N-Id)}$$
$$\sigma_y \circ \langle\,;\, h\rangle = h \qquad \text{(R-S-Id)}$$

where $f : [x_1 : a_1, \ldots, x_n : a_n\,;\, y_1 : b_1, \ldots, y_m : b_m] \to c$

- **Renaming laws** Ren-Comp-1 and Ren-Comp-2

We can construct II-multicategory $\mathbf{M}$ out of R-multicategory $\mathbf{M}(\mathfrak{R}_{II})$, where $\mathfrak{R}_{II}$ is a set of all II-renamings.

- $\mathbf{M}$ has the same objects and multiarrows as $\mathbf{M}(\mathfrak{R}_{II})$.

- Given a multiarrow

$$f : [x_1 : a_1, \ldots, x_n : a_n\,;\, y_1 : b_1, \ldots, y_m : b_m] \to c$$

and a family of multiarrows

$$g_{x_i} : [\Pi\,;\,] \to a_i \qquad 1 \le i \le n$$
$$g_{y_j} : [\Pi\,;\,\Theta] \to b_j \qquad 1 \le j \le m$$

we define

$$f \circ_{[\Pi;\Theta]} \langle g_{x_1}, \ldots, g_{x_n} ; h_{y_1}, \ldots, h_{y_m} \rangle = \\ (f \circ \langle g_{x_1} \cdot \varepsilon_{x_1}, \ldots, g_{x_n} \cdot \varepsilon_{x_n} ; h_{y_1} \cdot \varepsilon_{y_1}, \ldots, h_{y_m} \cdot \varepsilon_{y_m} \rangle) \cdot \sigma$$

where

$$\varepsilon_{x_i} : [\Pi ;] \to [\Pi_{x_i} ;] \qquad \varepsilon_{x_i}(u_j) = u_j^{x_i}$$
$$\varepsilon_{y_j} : [\Pi ; \Theta] \to [\Pi_{y_j} ; \Theta_{y_j}] \qquad \varepsilon_{y_j}(u_i) = u_i^{y_j} \qquad \varepsilon_{y_j}(v_i) = v_i^{y_j}$$
$$\sigma : [\Pi_{x_1}, \ldots, \Pi_{x_n}, \Pi_{y_1}, \ldots, \Pi_{y_m} ; \Theta_{y_1}, \ldots, \Theta_{y_m}] \to [\Pi ; \Theta]$$
$$\sigma(u_j^{x_i}) = u_i \quad \sigma(u_j^{y_i}) = u_j \quad \sigma(v_j^{y_i}) = v_j$$

- The normal identity multiarrows are defined by

$$[\Gamma ;] = [x_1 : a_1, \ldots, x_n : a_n ;] \xrightarrow{\pi_{x_i}} a_i$$
$$\pi_{x_i} = \pi_x \cdot \nu$$
$$\nu : [x : a_i ;] \to [\Gamma ;] \qquad \nu(x) = x_i$$

- The safe identity multiarrows are defined by

$$[\Gamma ; \Delta] = [x_1 : a_1, \ldots, x_n : a_n ; y_1 : b_1, \ldots, y_m : b_m] \xrightarrow{\sigma_{y_j}} b_j$$
$$\sigma_{y_j} = \sigma_y \cdot \mu$$
$$\mu : [; y : b_j] \to [\Gamma ; \Delta] \qquad \mu(y) = y_j$$

Proposition 3.5.2. **M** *is an II-multicategory.*

Proof. See appendix B. □

We can also obtain a R-multicategory out of II-multicategory.

Definition 3.5.3.
Give an II-multicategory **M** we define the R-multicategory $\hat{\mathbf{M}}$ as follows:

- $\hat{\mathbf{M}}$ has the same objects and multiarrows as **M**.

- The composition of a multiarrow

$$f : [x_1 : a_1, \ldots, x_n : a_n ; y_1 : b_1, \ldots, y_m : b_m] \to c$$

and a family of multiarrows

$$g_i : [\Pi_i\,;\,] \to a_i \qquad [\Pi_i\,;\,] = [x_1^i : c_1^i, \ldots, x_{k_i}^i : c_{k_i}^i\,;\,]$$
$$h_j : [\Lambda_j\,;\,\Theta_j] \to b_j \qquad [\Lambda_j\,;\,\Theta_j] = [u_1^j : d_1^j, \ldots, u_{l_j}^j : d_{l_j}^j\,;\, v_1^j : e_1^j, \ldots, v_{r_j}^j : e_{r_j}^j]$$

is defined as

$$f \circ \langle g_1, \ldots, g_n\,;\, h_1, \ldots, h_m\rangle = f \circ_{[\Pi;\Theta]} \langle g_1 \circ_{[\Pi;]} \langle \pi_{x_1^1}, \ldots, \pi_{x_{k_1}^1}\,;\,\rangle, \ldots,$$
$$g_n \circ_{[\Pi;]} \langle \pi_{x_1^n}, \ldots, \pi_{x_{k_n}^n}\,;\,\rangle\,;\, h_1 \circ_{[\Pi;\Theta]} \langle \pi_{u_1^1}, \ldots, \pi_{u_{l_1}^1}\,;\, \sigma_{v_1^1}, \ldots, \sigma_{v_{r_1}^1}\rangle, \ldots,$$
$$h_m \circ_{[\Pi;\Theta]} \langle \pi_{u_1^m}, \ldots, \pi_{u_{l_m}^m}\,;\, \sigma_{v_1^m}, \ldots, \sigma_{v_{r_m}^m}\rangle\rangle$$

where

$$[\Pi\,;\,\Theta] = [\Pi_1, \ldots, \Pi_n, \Lambda_1, \ldots, \Lambda_m\,;\,\Theta_1, \ldots, \Theta_m]$$

- For the safe and normal identities take the corresponding identities of **M**.
- Given an II-renaming

$$\varepsilon : [\Gamma\,;\,\Delta] = [x_1 : \sigma_1, \ldots, x_n : \sigma_n\,;\, y_1 : \tau_1, \ldots, y_m : \tau_m] \to [\Pi\,;\,\Theta]$$

and a multiarrow $f : [\Gamma\,;\,\Delta] \to c$ we define

$$f \cdot \varepsilon = f \circ_{[\Pi;\Theta]} \langle \pi_{\varepsilon(x_1)}, \ldots, \pi_{\varepsilon(x_n)}\,;\, \sigma_{\varepsilon(y_1)}, \ldots, \sigma_{\varepsilon(y_m)}\rangle$$

Proposition 3.5.4. $\hat{\mathbf{M}}$ *is an R-multicategory.*

Proof. See appendix B. □

In this chapter we defined notions of II- and IL-multicategory. The important feature of our approach to multicategories is the use of dual-contexts as domains for multiarrows compared to sequences of objects in the traditional approach. The usage of dual-contexts requires further modifications in the notions of composition of multiarrows and identity multiarrows.

The difference between II- and IL-multicategories is in the allowed structural operations on multiarrows. In the case of II-multicategories the usual intuitionistic structural operations can be performed for both safe and normal variables in dual-contexts, while in the case of IL-multicategories only linear usage of safe variables is allowed.

Dual-context multicategories allow more natural interpretation of dual-context calculi compared to using categories with structure. We defined interpretations of the

$\mathbb{III}(\Sigma)$ and the $\mathbb{IIL}(\Sigma)$ calculi using II- and IL-multicategories and proved that this interpretation was sound and complete.

Finally we outline one possible way of unifying the notions of II- and IL- multicategories using the notion of R-multicategories, in which structural operations on multiarrows are introduced explicitly as opposed to being constructed using composition operation and identity multiarrows.

Chapter 4

Dual-context type constructors

In chapter 2 we introduced two simple dual-context calculi $\mathbb{III}(\Sigma)$ and $\mathbb{IIL}(\Sigma)$, which had only the basic types and no type constructors. In this chapter we are going to extend the $\mathbb{III}(\Sigma)$ and the $\mathbb{IIL}(\Sigma)$ calculi with products, sums and unit type. For each of these extensions will will define a sound interpretation using extended notions of II- and IL-multicategories, obtaining multicategorical versions of cartesian products, coproducts and terminal objects.

4.1 Dual-context binding operators

In the natural deduction formulation of intuitionistic logic, each propositional connective is defined by introduction and elimination rules. Consider for example the rules for disjunction $\vee$:

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee\text{I-L} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash B \vee A}\ \vee\text{I-R} \qquad \frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C \qquad \Gamma \vdash A \vee B}{\Gamma \vdash C}\ \vee\text{E}$$

The $\vee$I-L and $\vee$I-R rules are quite simple, the context of assumptions is the same in the premise and the conclusion of each rule. In the $\vee$E rule the formula A is discharged from the first premise and the formula B is discharged from the second premise.

In general many different natural deduction rules can be described by the following data: the number n of premises of the rule, the sets of formulae Δ_i which are discharged in from each premise, the conclusions A_i of each premise and the final conclusion B. Such data is called the **arity** of the rule [Acz80] and can be written as follows:

$$\frac{(\Delta_1)\, A_1 \qquad \ldots \qquad (\Delta_n)\, A_n}{B}$$

where Δ_i are the sets of discharged assumptions.

In type theory, each natural deduction gives rise to a *binding operator* typed by a rule of the form:

$$\frac{\Gamma,(\Delta_1)\vdash t_1 : A_1 \qquad \ldots \qquad \Gamma,(\Delta_n)\vdash t_n : A_n}{\Gamma\vdash \mathsf{op}((\Delta_1)t_1,\ldots,(\Delta_n)t_n) : B} \qquad \textbf{(OP)}$$

where $\Gamma,\Delta_1,\ldots,\Delta_n$ are typing contexts and the operator construct $\mathsf{op}((\Delta_1)t_1,\ldots,(\Delta_n)t_n)$ binds all variables from the context Δ_i in the term t_i for all i from 1 to n.

Simultaneous substitution for operator terms can be defined as follows:

$$\mathsf{op}((\Delta_1)t_1,\ldots,(\Delta_n)t_n)[\alpha] = \mathsf{op}((\Delta_1)t_1[\alpha],\ldots,(\Delta_n)t_n[\alpha])$$

where we assume that some measures are taken in order to avoid incidental capturing of free variables.

We can show that in any type system given by some set of typing rules of the form **OP** the following substitution rule is admissible:

$$\frac{\begin{array}{c} x_1 : A_1,\ldots,x_n : A_n \vdash t : B \\ \Pi\vdash s_1 : A_1 \qquad \ldots \qquad \Pi\vdash s_n : A_n \end{array}}{\Pi\vdash t[s_1/x_1,\ldots,s_n/x_n] : B}$$

We shall now consider a generalization of such binding operators in dual-context setting. We start with **Π-operators**. First, we need to change all single contexts to the dual contexts and allow both normal and safe assumption to be discharged:

$$\frac{\Gamma,(\Pi_1)\,;\Delta,(\Theta_1)\vdash A_1 \qquad \ldots \qquad \Gamma,(\Pi_n)\,;\Delta,(\Theta_n)\vdash A_n}{\Gamma\,;\Delta\vdash B}$$

Next, we consider operators with two kinds of premises, called **normal** and **safe**. Normal premises will share only normal d-contexts and safe premises will share arbitrary d-contexts. It turns out that many interesting operators require this feature. So arity of an Π-operator looks as follows:

$$\frac{\begin{array}{c} \Gamma,(\Pi_1)\,;(\Theta_1)\vdash A_1 \qquad \ldots \qquad \Gamma,(\Pi_n)\,;(\Theta_n)\vdash A_n\,; \\ \Gamma,(\Phi_1)\,;\Delta,(\Psi_1)\vdash B_1 \qquad \ldots \qquad \Gamma,(\Phi_m)\,;\Delta,(\Psi_m)\vdash B_m \end{array}}{\Gamma\,;\Delta\vdash C}$$

In the case of an Π-operator with only normal premises its arity is

$$\frac{\Gamma,(\Pi_1)\,;(\Theta_1)\vdash A_1 \qquad \ldots \qquad \Gamma,(\Pi_n)\,;(\Theta_n)\vdash A_n\,;}{\Gamma\,;\Delta\vdash C}$$

where Δ is arbitrary, not necessarily empty.

Consider a set of II-operators $\mathbb{O}$ with associated arities as above. We shall now define an extension of the $\mathbb{III}(\Sigma)$ calculus with such binding operators, called $\mathbb{III}(\Sigma)$+$\mathbb{O}$. For each II-operator in $\mathbb{O}$ we introduce a new kind of pre-term:

$$\mathsf{op}((\bar{x}_1 : \bar{\sigma}_1 ; \bar{y}_1 : \bar{\tau}_1)t_1, \ldots, (\bar{x}_n : \bar{\sigma}_n ; \bar{y}_n : \bar{\tau}_n)t_n ; \\ (\bar{x}'_1 : \bar{\sigma}'_1 ; \bar{y}'_1 : \bar{\tau}'_1)t'_1, \ldots, (\bar{x}'_m : \bar{\sigma}'_m ; \bar{y}'_m : \bar{\tau}'_m)t'_m)$$

where $(\bar{x}_i : \bar{\sigma}_i ; \bar{y}_i : \bar{\tau}_i)t$ stands for $(x_1 : \sigma_1, \ldots, x_{n_i} : \sigma_{n_i} ; y_1 : \tau_1, \ldots, y_{m_i} : \tau_{m_i})t$ and represents a binding of the variables $x_1, \ldots, x_n, y_1, \ldots, y_m$ in the term t. The set of free variables of a term is defined as before with the following additional case for each operator:

$$\mathsf{FV}(\mathsf{op}((\bar{x}_1 : \bar{\sigma}_1 ; \bar{y}_1 : \bar{\tau}_1)t_1, \ldots, (\bar{x}_n : \bar{\sigma}_n ; \bar{y}_n : \bar{\tau}_n)t_n ; (\bar{x}'_1 : \bar{\sigma}'_1 ; \bar{y}'_1 : \bar{\tau}'_1)t'_1, \ldots \\ (\bar{x}'_m : \bar{\sigma}'_m ; \bar{y}'_m : \bar{\tau}'_m)t'_m)) = \bigcup_{i=1}^{n}(\mathsf{FV}(t_i) - \{\bar{x}_i, \bar{y}_i\}) \cup \bigcup_{i=1}^{m}(\mathsf{FV}(t'_i) - \{\bar{x}'_i, \bar{y}'_i\})$$

In the presence of binding we require all substitutions to be capture-avoiding. The simultaneous substitution is defined as before with the following additional case for each operator:

$$\mathsf{op}((\bar{x}_1 : \bar{\sigma}_1 ; \bar{y}_1 : \bar{\tau}_1)t_1, \ldots, (\bar{x}_n : \bar{\sigma}_n ; \bar{y}_n : \bar{\tau}_n)t_n ; (\bar{x}'_1 : \bar{\sigma}'_1 ; \bar{y}'_1 : \bar{\tau}'_1)t'_1, \ldots \\ (\bar{x}'_m : \bar{\sigma}'_m ; \bar{y}'_m : \bar{\tau}'_m)t'_m)[\alpha] = \mathsf{op}((\bar{x}_1 : \bar{\sigma}_1 ; \bar{y}_1 : \bar{\tau}_1)t_1[\alpha], \ldots, (\bar{x}_n : \bar{\sigma}_n ; \bar{y}_n : \bar{\tau}_n)t_n[\alpha] ; \\ (\bar{x}'_1 : \bar{\sigma}'_1 ; \bar{y}'_1 : \bar{\tau}'_1)t'_1[\alpha], \ldots (\bar{x}'_m : \bar{\sigma}'_m ; \bar{y}'_m : \bar{\tau}'_m)t'_m[\alpha])$$

For each operator we add the following typing rule to $\mathbb{III}(\Sigma)$:

$$\frac{\begin{array}{ccc} \Gamma, \Pi_1 ; \Theta_1 \vdash t_1 : \sigma_1 & \ldots & \Gamma, \Pi_n ; \Theta_n \vdash t_n : \sigma_n ; \\ \Gamma, \Phi_1 ; \Delta, \Psi_1 \vdash s_1 : \tau_1 & \ldots & \Gamma, \Phi_m ; \Delta, \Psi_m \vdash s_m : \tau_m \end{array}}{\Gamma ; \Delta \vdash \mathsf{op}((\Pi_1 ; \Theta_1)t_1, \ldots, (\Pi_n ; \Theta_n)t_n ; (\Phi_1 ; \Psi_1)s_1, \ldots, (\Phi_m ; \Psi_m)s_m) : \rho} \quad \textbf{II-Op}$$

For a normal operator such rule will have the following form:

$$\frac{\Gamma, \Pi_1 ; \Theta_1 \vdash t_1 : \sigma_1 \quad \ldots \quad \Gamma, \Pi_n ; \Theta_n \vdash t_n : \sigma_n ;}{\Gamma ; \Delta \vdash \mathsf{op}((\Pi_1 ; \Theta_1)t_1, \ldots, (\Pi_n ; \Theta_n)t_n ;) : \rho}$$

where Δ is arbitrary.

The type system $\mathbb{III}(\Sigma)$+$\mathbb{O}$ retains all the basic structural properties of $\mathbb{III}(\Sigma)$. In particular, the safe substitution rule **II-Subst** is admissible. Before showing this we need the prove following lemma:

Lemma 4.1.1. *If a term* $\Gamma ; \Delta \vdash t : \rho$ *can be derived in* $\mathbb{III}(\Sigma)$+$\mathbb{O}$ *then* $\Gamma, \Gamma' ; \Delta, \Delta' \vdash t : \rho$ *is derivable as well.*

Proof. Induction on the derivation of t. It is important that in the typing rules for II-operators with only normal premises, Δ in the conclusion of each rule is allowed to be non-empty. □

Theorem 4.1.2. ***II-Subst*** *is admissible in* $\mathbb{III}(\Sigma)+\mathbb{O}$.

Proof. Recall the **II-Subst** rule for $\Gamma;\Delta\vdash t:\rho$

$$\frac{\begin{array}{ccc}\Pi;\vdash u_1:\sigma_1 & \ldots & \Pi;\vdash u_n:\sigma_n\\ \Pi;\Theta\vdash v_1:\tau_1 & \ldots & \Pi;\Theta\vdash v_m:\tau_m\end{array}}{\Pi;\Theta\vdash t[u_1/x_1,\ldots,u_n/x_n,v_1/y_1,\ldots,v_m/y_m]:\rho}$$

We consider the only the new case when t is typed using one of the added rules:

$$t=\mathsf{op}((\Pi_1;\Theta_1)t_1,\ldots,(\Pi_1;\Theta_1)t_n;(\Phi_1;\Psi_1)s_1,\ldots,(\Phi_1;\Psi_1)s_m)$$

Abbreviate $\alpha=u_1/x_1,\ldots,u_n/x_n$ and $\beta=v_1/y_1,\ldots,v_m/y_m$. By the definition of substitution

$$\mathsf{op}(t_1,\ldots,t_n;s_1,\ldots,s_m)[\alpha,\beta]=\mathsf{op}(t_1[\alpha,\beta],\ldots,t_n[\alpha,\beta];s_1[\alpha,\beta],\ldots,s_m[\alpha,\beta])$$

Consider arbitrary $\Gamma,\Phi_i;\Delta,\Psi_i\vdash s_i:\tau_i$. We assume that

$$\begin{array}{ll}\Gamma=x_1:\sigma_1,\ldots,x_n:\sigma_n & \Delta=y_1:\tau_1,\ldots,y_m:\tau_m\\ \Phi_i=x'_1:\rho_1,\ldots,x'_k:\rho_k & \Psi_i=y'_1:\theta_1,\ldots,y'_l:\theta_l\end{array}$$

By the properties of substitution

$$s_i[\alpha,\beta]=s_i[u_1/x_1,\ldots,u_n/x_n,v_1/y_1,\ldots,v_m/y_m,x'_1/x'_1,\ldots,x'_k/x'_k,y'_1/y'_1,\ldots,y'_l/y'_l]$$

Given $\Pi;\vdash u_1:\sigma_1,\ldots,\Pi;\vdash u_n:\sigma_n$ and $\Pi;\Theta\vdash v_1:\tau_1,\ldots,\Pi;\Theta\vdash v_m:\tau_m$ by lemma 4.1.1 we can apply weakening and get the following terms

$$\begin{array}{ccc}\Pi,\Phi_i;\vdash u_1:\sigma_1 & \ldots & \Pi,\Phi_i;\vdash u_n:\sigma_n\\ \Pi,\Phi_i;\Theta,\Psi_i\vdash v_1:\tau_1 & \ldots & \Pi,\Phi_i;\Theta,\Psi_i\vdash v_m:\tau_m\end{array}$$

Applying the **II-Subst** rule we get $\Pi,\Phi_i;\Theta,\Psi_i\vdash s_i[\alpha,\beta]:\tau_i$. Similarly we derive $\Pi,\Pi_i;\vdash t_i[\alpha,\beta]:\sigma_i$ for all $1\leq i\leq n$. Using the operator rule we get

$$\Pi;\Theta\vdash\mathsf{op}(t_1[\alpha,\beta],\ldots,t_n[\alpha,\beta];s_1[\alpha,\beta],\ldots,s_m[\alpha,\beta]):\rho$$

□

4.2 Products in $\mathbb{III}(\Sigma)$

In this section we shall consider dual-context products in the $\mathbb{III}(\Sigma)$ and the $\mathbb{IIL}(\Sigma)$ calculi give multicategorical interpretation for it.

In the single-context case, the typing rules for products are well-known. We adopt the elegant formulation in [Pit95]):

$$\frac{\Gamma \vdash n : \sigma \qquad \Gamma \vdash m : \tau}{\Gamma \vdash \mathsf{pair}(n,m) : \sigma \times \tau} \qquad \frac{\Gamma \vdash p : \sigma \times \tau \qquad \Gamma, x : \sigma, y : \tau \vdash s : \rho}{\Gamma \vdash \mathsf{let}(x,y) = p \text{ in } s : \rho}$$

with the following equality judgements:

$$\frac{\Gamma \vdash n : \sigma \qquad \Gamma \vdash m : \tau \qquad \Gamma, x : \sigma, y : \tau \vdash s : \rho}{\Gamma \vdash \mathsf{let}\ (x,y) = \mathsf{pair}(n,m) \text{ in } s = s[n/x, m/y] : \rho}\ \beta$$

$$\frac{\Gamma \vdash p : \sigma \times \tau \qquad \Gamma, z : \sigma \times \tau \vdash t : \rho}{\Gamma \vdash \mathsf{let}\ (x,y) = p \text{ in } t[\mathsf{pair}(x,y)/z] = t[p/z] : \rho}\ \eta$$

One can show that in order to soundly interpret such rules in a category with finite products one can use the existing product structure.

We start with extending the $\mathbb{III}(\Sigma)$ calculus with **normal products** $\sigma \times \tau$, which will be restricted to normal terms only. Terms of type $\sigma \times \tau$ are introduced by the following rule:

$$\frac{\Gamma ; \vdash t_1 : \sigma \qquad \Gamma ; \vdash t_2 : \tau ;}{\Gamma ; \Delta \vdash \mathsf{pair}(t_1, t_2 ;) : \sigma \times \tau}$$

Note that both t_1 and t_2 are normal terms. This restriction will play crucial role when we shall define basic equations for the normal product types.

The elimination rule for the $\sigma \times \tau$ types can be given as follows:

$$\frac{\Gamma ; \vdash p : \sigma \times \tau \quad ; \quad \Gamma, x : \sigma, y : \tau ; \Delta \vdash s : \rho}{\Gamma ; \Delta \vdash \mathsf{let}\ (x,y) = p \text{ in } s : \rho}$$

The important feature of this rule is that both variables x and y which are being bound in the term s are normal.

In order to give semantics of the normal product rules in an II-multicategory $\mathbf{M}$ we need a binary operation on objects $\times : |\mathbf{M}| \times |\mathbf{M}| \to |\mathbf{M}|$ which will be used for the interpretation of the $\times$ operation on basic types:

$$[\![\sigma \times \tau]\!] = [\![\sigma]\!] \times [\![\tau]\!]$$

The semantics of the introduction rule is given by the following operation on multiarrows:

$$pair_\Delta : \mathbf{M}([\Gamma ;], a) \times \mathbf{M}([\Gamma ;], b) \to \mathbf{M}([\Gamma ; \Delta], a \times b)$$

which commutes with II-composition in the following way:

$$pair_\Delta(f,g) \circ_{[\Pi;\Theta]} \langle \bar{s}\,;\bar{t}\rangle = pair_\Theta(f \circ_{[\Pi;]} \langle \bar{s}\,;\rangle, g \circ_{[\Pi;]} \langle \bar{s}\,;\rangle)$$

Note that since the variables in Δ were introduced by an implicit weakening the multi-arrows $\bar{t}$ do not appear on the right-hand side. Also the composition on the right-hand side is restricted to the normal d-context $[\Pi;]$ rather than $[\Pi;\Theta]$. This property ensures that the application of the $pair_\Delta$ to the multiarrows $[\Gamma;] \xrightarrow{f} a$ and $[\Gamma;] \xrightarrow{g} b$ is completely determined by the composition with a multi-arrow $p_{a,b} = pair_{[]}(\pi_x^\Pi, \pi_y^\Pi)$, where $[\Pi;] = [x:a, y:b;]$:

$$pair_\Delta(f,g) = pair_\Delta(\pi_x^\Pi \circ_{[\Gamma;]} \langle f,g\,;\rangle, \pi_y^\Pi \circ_{[\Gamma;]} \langle f,g\,;\rangle) = p_{a,b} \circ_{[\Gamma;\Delta]} \langle f,g\,;\rangle$$

So the semantics of the introduction rule can be given as follows:

$$[\![\Gamma\,;\Delta \vdash \mathsf{pair}(m,n) : \sigma \times \tau]\!] \stackrel{\text{def}}{=} p_{[\![\sigma]\!],[\![\tau]\!]} \circ_{[\![\Gamma;\Delta]\!]} \langle [\![m]\!], [\![n]\!]\,;\rangle$$

For the interpretation of the elimination rule we need the following operation on multiarrows:

$$split : \mathbf{M}([\Gamma\,;], a \times b) \times \mathbf{M}([\Gamma, x:a, y:b\,;\Delta], c) \to \mathbf{M}([\Gamma\,;\Delta], c)$$

which should commute with II-composition in the following way:

$$[\Pi\,;\Theta] \quad split(p,s)(\bar{f}\,;\bar{g}) = split(p',s')$$

where p' is the multiarrow $p(\bar{f};)$ and s' is the multiarrow $s(\bar{f},x,y)$.

Since *split* commutes with II-composition it can be expressed as follows:

$$[\Gamma\,;\Delta] \quad split(p,s) = sp(s)(\bar{x}, p\,;\bar{y})$$

using the operation

$$sp : \mathbf{M}([\Gamma, x:a, y:b\,;\Delta], c) \to \mathbf{M}([\Gamma, z:a \times b\,;\Delta], c)$$

which should also commute with II-composition:

$$[\Pi, z:a \times b\,;\Theta] \quad sp(s)(\bar{g}, z\,;\bar{h}) = sp(s')$$

where s' is $(\bar{g}, x, y\,;\bar{h})$.

We can define the interpretation of the elimination rule in the following way:

$$[\![\Gamma\,;\Delta \vdash \mathsf{let}\ (x,y) = p\ \mathbf{in}\ s]\!] \stackrel{\text{def}}{=} sp(s)(\bar{x}, [\![p]\!]\,;\bar{y})$$

We shall now consider basic equalities for the normal product types. First we have the following beta equality:

$$\frac{\Gamma;\vdash n:\sigma \qquad \Gamma;\vdash m:\tau \qquad \Gamma,x:\sigma,y:\tau;\Delta\vdash s:\rho}{\Gamma;\Delta\vdash \mathsf{let}\ (x,y)=\mathsf{pair}(n,m)\ \mathsf{in}\ s=s[n/x,m/y]:\rho}$$

This equality is important in understanding the intuition behind our definition of normal product types. Since we have chosen both x and y to be normal in the elimination rule for $\sigma\times\tau$ we have to restrict to normal n and m to satisfy the restriction of II-composition.

This beta equality rule translates into the following property of sp and $p_{a,b}$:

$$[\Gamma;\Delta]\quad sp(s)(\bar{x},p_{a,b}(f,g);\bar{y})=s(\bar{x},f,g;\bar{y})$$

for every multiarrow $s:[\Gamma,x:a,y:b;\Delta]\to c$, $f:[\Gamma;]\to a$ and $g:[\Gamma;]\to b$.

Next, we ask for the following eta equality to hold:

$$\frac{\Gamma;\vdash r:\sigma\times\tau \qquad \Gamma,z:\sigma\times\tau;\Delta\vdash t:\rho}{\Gamma;\Delta\vdash \mathsf{let}\ (x,y)=r\ \mathsf{in}\ t[\mathsf{pair}(x,y)/z]))=t[r/z]:\rho}$$

This equality translates into the following property ofsp and $p_{a,b}$:

$$[\Gamma;\Delta]\quad sp(t(\bar{x},p_{a,b}(x,y;);\bar{y}))(\bar{x},r;\bar{y})=t(\bar{x},r;\bar{y})$$

for every multiarrow $t:[\Gamma,z:a\times b;\Delta]\to c$ and $r:[\Gamma;]\to a\times b$.

Given an II-multicategory $\mathbf{M}$ with sp and $p_{a,b}$, which satisfy β and η equalities there exists the following bijection between the sets of multiarrows:

$$\mathbf{M}([\Gamma,x:a,y:b;\Delta],c)\to\mathbf{M}([\Gamma,z:a\times b;\Delta],c)$$

given by sp and with inverse given by the composition with $p_{a,b}$:

$$[\Gamma,x:a,y:b;]\quad sp(s)(\bar{x},p_{a,b}(x,y;);)=s(\bar{x},x,y;)\qquad\text{using the }\beta\text{ equality}$$
$$[\Gamma,z:a\times b;]\quad sp(h(\bar{x},p_{a,b}(x;y;)))(\bar{x},z;)=h(\bar{x},z;)\qquad\text{using the }\eta\text{ equality}$$

We can construct the following multiarrows

$$[z:a\times b;]\quad fst(z;)=sp(x)$$
$$[z:a\times b;]\quad snd(z;)=sp(y)$$

Using these multiarrows we can construct an isomorphism

$$\mathbf{M}([\Gamma;],a)\times\mathbf{M}([\Gamma;],b)\to\mathbf{M}([\Gamma;],a\times b)$$

given by the composition with $p_{a,b}$ and inverse by compositions with fst and snd. In one direction by the β equality for sp and $p_{a,b}$ we have

$$sp(x)(p_{a,b}(f,g;)) = f \qquad sp(y)(p_{a,b}(f,g;)) = g$$

In the opposite direction we have

$$\begin{aligned}[\Gamma;] \quad p_{a,b}(fst(p;),snd(p;);) &= s(p_{a,b}(fst(p_{a,b}(x,y;);),snd(p_{a,b};);))(p;)\\ &= s(p_{a,b}(x,y;))(p;)\\ &= p\end{aligned}$$

Using this isomorphism we can give a different interpretation of the normal product in an II-multicategory **M** using the following definition:

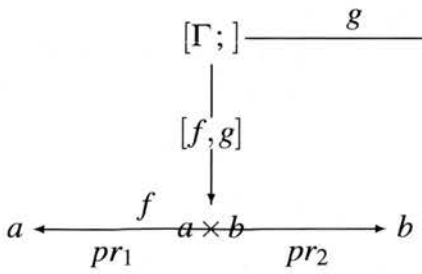
Definition 4.2.1 (Normal product)**.**
Let **M** be an II-multicategory. Let $a,b \in |\mathbf{M}|$. A **normal product** of a and b is an object $a \times b$ together with the two multiarrows

$$pr_1 : [x : a \times b;] \to a \qquad pr_2 : [x : a \times b;] \to b$$

such that for any pair of multiarrows

$$f : [\Gamma;] \to a \qquad g : [\Gamma;] \to b$$

there exists a unique multiarrow $[f,g] : [\Gamma;] \to a \times b$ making the following diagram commute:



$$\begin{aligned}[\Gamma;] \quad pr_1([f,g];) &= f\\ [\Gamma;] \quad pr_2([f,g];) &= g\end{aligned}$$

In II-multicategory with normal products we can define

$$\begin{aligned}[x : a, y : b;] \quad p_{a,b}(x,y;) &= [\pi_x,\pi_y]\\ [\Gamma,z : a \times b;\Delta] \quad sp(s)(\bar{x},z;\bar{y}) &= s(\bar{x},pr_1(z;),pr_2(z;))\end{aligned}$$

and define the semantics of the product rules as follows:

$$[\![\Gamma;\Delta\vdash \mathsf{pair}(m,n)]\!] \stackrel{\text{def}}{=} [[\![m]\!],[\![n]\!]]$$
$$[\![\Gamma;\Delta\vdash \mathsf{let}\ (x,y)=p\ \mathsf{in}\ s]\!] \stackrel{\text{def}}{=} [\![s]\!](\bar{x},pr_1([\![p]\!];),pr_2([\![p]\!];);)$$

This interpretations is sound with respect to β equality

$$[\Gamma;\Delta]\quad s(\bar{x},pr_1([n,m];),pr_2([n,m];);)=s(\bar{x},n,m;)$$

and η equality

$$[\Gamma;\Delta]\quad t(\bar{x},[pr_1(r;),pr_2(r;)];)=t(\bar{x},r;)$$

since $[pr_1(r;),pr_2(r;)]=r$ by the universal property of normal product.

We shall now consider a different version of product types called **safe products**. The introduction rule for the safe products looks as follows:

$$\frac{\Gamma;\Delta\vdash t_1:\sigma \qquad \Gamma;\Delta\vdash t_2:\tau}{\Gamma;\Delta\vdash \mathsf{pair}(t_1,t_2):\sigma\times\tau}$$

Note that in this rule t_1 and t_2 do not have to be normal and the introduction rule for normal product types is the special case of this rule.

The elimination rule for the safe $\sigma\times\tau$ has the following form:

$$\frac{\Gamma;\Delta\vdash p:\sigma\times\tau \qquad \Gamma;\Delta,x:\sigma,y:\tau\vdash s:\rho}{\Gamma;\Delta\vdash \mathsf{let}\ (x,y)=p\ \mathsf{in}\ s:\rho}$$

The important feature of this rule is that both x and y are safe variables.

We ask for the following beta equality for safe products:

$$\frac{\Gamma;\Delta\vdash n:\sigma \qquad \Gamma;\Delta\vdash m:\tau \qquad \Gamma;\Delta,x:\sigma,y:\tau\vdash s:\rho}{\Gamma;\Delta\vdash \mathsf{let}\ (x,y)=\mathsf{pair}(n,m)\ \mathsf{in}\ s=s[n/x,m/y]:\rho}$$

and the following eta equality:

$$\frac{\Gamma;\Delta\vdash r:\sigma\times\tau \qquad \Gamma;\Delta,z:\sigma\times\tau\vdash t:\rho}{\Gamma;\Delta\vdash \mathsf{let}\ (x,y)=r\ \mathsf{in}\ t[\mathsf{pair}(x,y)/z]))=t[r/z]:\rho}$$

Safe product types can be interpreted in an II-multicategory **M** with the following additional structure:

Definition 4.2.2 (Safe product in II-multicategory)**.**
Let **M** be a II-multicategory. Let $a,b\in|\mathbf{M}|$. A **safe product** of a and b is an object $a\times b$ together with the two multiarrows

$$pr_1:[;x:a\times b]\to a \qquad pr_2:[;x:a\times b]\to b$$

such that for any pair of multiarrows

$$f : [\Gamma ; \Delta] \to a \qquad g : [\Gamma ; \Delta] \to b$$

there exists a unique multiarrow $[f,g] : [\Gamma ; \Delta] \to a \times b$ such that

$$[\Gamma ; \Delta] \quad pr_1(;[f,g](\bar{x};\bar{y})) = f(\bar{x};\bar{y})$$
$$[\Gamma ; \Delta] \quad pr_2(;[f,g](\bar{x};\bar{y})) = g(\bar{x};\bar{y})$$

We can define an interpretation of safe product rules in an II-multicategory $\mathbf{M}$ with a safe product as follows:

$$[\![\sigma \times \tau]\!] \stackrel{\text{def}}{=} [\![\sigma]\!] \times [\![\tau]\!]$$
$$[\![\Gamma ; \Delta \vdash \mathsf{pair}(m,n)]\!] \stackrel{\text{def}}{=} [[\![m]\!], [\![n]\!]]$$
$$[\![\Gamma ; \Delta \vdash \mathsf{split}(p,s)]\!] \stackrel{\text{def}}{=} [\![s]\!](\bar{x};\bar{y}, pr_1(;[\![p]\!]), pr_2(;[\![p]\!]))$$

This interpretation is sound with respect to β equality

$$[\Gamma ; \Delta] \quad s(\bar{x};\bar{y}, pr_1(;[n,m]), pr_2(;[n,m])) = s(\bar{x};\bar{y},n,m)$$

and η equality

$$[\Gamma ; \Delta] \quad t(\bar{x};\bar{y},[pr_1(;r), pr_2(;r)]) = t(\bar{x};\bar{y},r)$$

since $[\Gamma ; \Delta] \quad [pr_1(;r), pr_2(;r)] = r$ by the universal property of safe product.

The property of having safe products is a stronger requirement compared to having normal products as the following proposition shows:

Proposition 4.2.3. *Let $\mathbf{M}$ be an II-multicategory with safe products. Then $\mathbf{M}$ has normal products.*

Proof. We shall use superscripts to distinguish between the normal and the safe products. The normal product will be defined as follows:

- $a \times^n b = a \times^s b$
- Projections are defined as follows:

$$[x : a \times b;] \quad pr_1^n(x;) = pr_1^s(;x)$$
$$[x : a \times b;] \quad pr_2^n(x;) = pr_2^s(;x)$$

- Given $f : [\Gamma;] \to a$ and $g : [\Gamma;] \to b$ take $[f,g]^n = [f,g]^s$

Then we have the following:

$$[\Gamma;] \quad pr_1^n([f,g]^n(\bar{x};);) = pr_1^s(;[f,g]^s(\bar{x};)) = f(\bar{x};)$$

□

So far we considered binary product types. Now we look at the 'nullary' product, called unit type and denoted as **1**. Terms of type **1** are introduced by the following rule:

$$\frac{}{\Gamma;\Delta \vdash \star : \mathbf{1}}$$

We consider two different elimination rules for unit type. The elimination rule for the **normal** unit type looks as follows:

$$\frac{\Gamma;\vdash s:\mathbf{1} \qquad \Gamma;\Delta\vdash t:\sigma}{\Gamma;\Delta\vdash \mathsf{let}\ \star\ \mathsf{be}\ s\ \mathsf{in}\ t:\sigma}$$

The elimination rule for the **safe** unit type will be

$$\frac{\Gamma;\Delta\vdash s:\mathbf{1} \qquad \Gamma;\Delta\vdash t:\sigma}{\Gamma;\Delta\vdash \mathsf{let}\ \star\ \mathsf{be}\ s\ \mathsf{in}\ t:\sigma}$$

The only difference between these two rules is that in the case of the normal unit type the term s must be normal. This restriction allows the following eta equality for the normal unit type:

$$\frac{\Gamma;\vdash s:\mathbf{1} \qquad \Gamma,x:\mathbf{1};\Delta\vdash t:\sigma}{\Gamma;\Delta\vdash \mathsf{let}\ \star\ \mathsf{be}\ s\ \mathsf{in}\ t[\star/x]=t[s/x]}$$

Note that the safe substitution $t[s/x]$ is valid since s is a normal term. In the case of the safe unit type the eta equality looks as follows:

$$\frac{\Gamma;\Delta\vdash s:\mathbf{1} \qquad \Gamma;\Delta,x:\mathbf{1}\vdash t:\sigma}{\Gamma;\Delta\vdash \mathsf{let}\ \star\ \mathsf{be}\ s\ \mathsf{in}\ t[\star/x]=t[s/x]}$$

The following beta equality is derivable for both safe and normal unit types:

$$\frac{\Gamma;\Delta\vdash t:\sigma}{\Gamma;\Delta\vdash \mathsf{let}\ \star\ \mathsf{be}\ \star\ \mathsf{in}\ t=t:\sigma}$$

We can construct the following bijection between sets of terms:

$$\Gamma,x:\mathbf{1};\Delta\vdash s:\sigma \mapsto \Gamma;\Delta\vdash s[\star/x]:\sigma$$

$$\Gamma;\Delta\vdash t:\sigma \mapsto \Gamma,x:\mathbf{1};\Delta\vdash \mathsf{let}\ \star\ \mathsf{be}\ x\ \mathsf{in}\ t:\sigma$$

Using the eta and beta equalities defined above, we can show these mappings indeed form a bijection.

In single-context case categorical interpretation for unit type is given using the notion of a terminal object. In dual-context case we need the following structure for the interpretation of the unit type:

- An object 1 for the interpretation of $\mathbf{1}$
- A multiarrow $\iota : [\Gamma\,;\Delta] \to 1$ for each d-context $[\Gamma\,;\Delta]$ so that the introduction rule is interpreted as $[\![\Gamma\,;\Delta \vdash \star : \mathbf{1}]\!] \stackrel{\text{def}}{=} \iota$.
- Interpretation of the elimination rule for normal unit type requires a multiarrow $p_a : [x : 1\,; y : a] \to a$ using which we define

$$[\![\Gamma\,;\Delta \vdash \mathsf{let}\ \star\ \mathsf{be}\ s\ \mathsf{in}\ t : \sigma]\!] \stackrel{\text{def}}{=} p_{[\![\sigma]\!]}([\![s]\!]\,;[\![t]\!])$$

The eta equality then translates into the following property of p_a:

$$p_a(s(\bar{x};)\,;t(\bar{x},\iota\,;\bar{y})) = t(\bar{x},s\,;\bar{y})$$

for every $s : [\Gamma;] \to 1$ and $t : [\Gamma,x : 1\,;\Delta] \to a$. If we take $[z : 1;] \xrightarrow{\pi_z} 1$ for s and $[z : 1,x : 1\,; y : a] \xrightarrow{\sigma_y} a$ for t we get $[z : 1\,; y : a]\ p_a(z;y) = y$. So we have that $[x : 1\,; y : a]\ p_a = \sigma_y$ and $[\![\Gamma\,;\Delta \vdash \mathsf{let}\ \star\ \mathsf{be}\ s\ \mathsf{in}\ t : \sigma]\!] = [\![t]\!]$.

Definition 4.2.4.
A **normal terminal object** in an II-multicategory $\mathbf{M}$ is an object 1 such that for any normal d-context $[\Gamma;]$ there exists a unique multiarrow $[\Gamma;] \to 1$. The object 1 is a **safe terminal object** if for any d-context $[\Gamma\,;\Delta]$ there exists a unique multiarrow $[\Gamma\,;\Delta] \to 1$.

4.3 Sum types

In this section we are going to consider several extensions of the $\mathbb{III}(\Sigma)$ calculus with sum type constructor. In the single-context case, sum constructor is defined as follows [Pit95]:

$$\frac{\Gamma \vdash t : \sigma}{\Gamma \vdash \mathsf{inl}_\tau(t) : \sigma + \tau} \qquad \frac{\Gamma \vdash t : \tau}{\Gamma \vdash \mathsf{inr}_\sigma(t) : \sigma + \tau}$$

$$\frac{\Gamma \vdash c : \sigma + \tau \qquad \Gamma, x : \sigma \vdash t_1 : \rho \qquad \Gamma, y : \tau \vdash t_2 : \rho}{\Gamma \vdash \mathsf{case}(c, (x : \sigma)\, t_1, (y : \tau)\, t_2) : \rho}$$

$$\frac{\Gamma \vdash s : \sigma \qquad \Gamma, x : \sigma \vdash t_1 : \rho \qquad \Gamma, y : \tau \vdash t_2 : \rho}{\Gamma \vdash \mathsf{case}(\mathsf{inl}_\tau(s), (x : \sigma)\, t_1, (y : \tau)\, t_2) = t_1[s/x]}$$

$$\frac{\Gamma \vdash s : \tau \qquad \Gamma, x : \sigma \vdash t_1 : \rho \qquad \Gamma, y : \tau \vdash t_2 : \rho}{\Gamma \vdash \mathsf{case}(\mathsf{inr}_\sigma(s), (x : \sigma)\, t_1, (y : \tau)\, t_2) = t_2[s/y]}$$

$$\frac{\Gamma \vdash s : \sigma+\tau \qquad \Gamma, z:\sigma+\tau \vdash t : \rho}{\Gamma \vdash \mathsf{case}(s,(x:\sigma)\, t[\mathsf{inl}_\tau(x)/z],(y:\tau)\, t[\mathsf{inr}_\sigma(y)/z]) = t[s/z]}$$

There are several ways of adding sums types to the $\mathbb{III}(\Sigma)$ calculus. Firstly, we restrict to normal terms only and define **normal** sums. The introduction rules for the normal $\sigma+\tau$ look as follows:

$$\frac{\Gamma;\vdash t:\sigma}{\Gamma;\Delta\vdash \mathsf{inl}_\tau(t):\sigma+\tau} \qquad \frac{\Gamma;\vdash t:\tau}{\Gamma;\Delta\vdash \mathsf{inr}_\sigma(t):\sigma+\tau}$$

Consider the following elimination rule for the terms of type $\sigma+\tau$:

$$\frac{\Gamma;\vdash s:\sigma+\tau \quad ; \quad \Gamma,x:\sigma;\Delta\vdash t_1:\rho \qquad \Gamma,y:\tau;\Delta\vdash t_2:\rho}{\Gamma;\Delta\vdash \mathsf{case}(s,(x:\sigma;)\, t_1,(y:\tau;)\, t_2):\rho}$$

These rules can be interpreted in an II-multicategory $\mathbf{M}$ as follows. First, we need is a binary operation on objects $+ : |\mathbf{M}| \times |\mathbf{M}| \to |\mathbf{M}|$ for constructing sums:

$$[\![\sigma+\tau]\!] \stackrel{\text{def}}{=} [\![\sigma]\!] + [\![\tau]\!]$$

For the interpretation of the introduction rules we need two natural operations on multiarrows:

$$inl_{a,b} : \mathbf{M}([\Gamma;],a) \to \mathbf{M}([\Gamma\,;\Delta],a+b) \qquad inr_{a,b} : \mathbf{M}([\Gamma;],b) \to \mathbf{M}([\Gamma\,;\Delta],a+b)$$

Such operations are determined by composition with the following multiarrows

$$inl_{a,b} : [x:a\,;] \to a+b \qquad inr_{a,b} : [x:b\,;] \to a+b$$

The interpretation of $\mathsf{inl}_\sigma(t)$ and $\mathsf{inr}_\tau(t)$ terms is given as follows:

$$[\![\mathsf{inl}_\sigma(t)]\!] = inl_{[\![\sigma]\!],[\![\tau]\!]}([\![t]\!];)$$
$$[\![\mathsf{inr}_\tau(t)]\!] = inr_{[\![\sigma]\!],[\![\tau]\!]}([\![t]\!];)$$

For the interpretation of the elimination rule we need an operation

$$\{-|-\} : \mathbf{M}([\Gamma,x:a\,;\Delta],c) \times \mathbf{M}([\Gamma,y:b\,;\Delta],c) \to \mathbf{M}([\Gamma,z:a+b\,;\Delta],c)$$

which should satisfy the following naturality property:

$$[\Pi,z:a+b\,;\Theta]\ \{f|g\}(\bar{s},z;\bar{t}) = \{f'|g'\}$$

where $[\Pi,x:a\,;\Theta]\ f' = f(\bar{s},x;\bar{t})$ and $[\Pi,y:b\,;\Theta]\ g' = g(\bar{s},y;\bar{t})$ for $f \in \mathbf{M}([\Gamma,x:a\,;\Delta],c)$ and $g \in \mathbf{M}([\Gamma,y:b\,;\Delta],c)$. Using this operation we define the interpretation of the elimination rule as follows:

$$[\![\Gamma\,;\Delta\vdash \mathsf{case}(s,(x:\sigma;)\, t_1,(y:\tau;)\, t_2)]\!] \stackrel{\text{def}}{=} \{[\![t_1]\!]|[\![t_2]\!]\}(\bar{x},[\![s]\!]\,;\bar{y})$$

For $\sigma+\tau$ type we consider the following two beta-equality rules:

$$\frac{\Gamma;\vdash s:\sigma \qquad \Gamma,x:\sigma;\Delta\vdash t_1:\rho \qquad \Gamma,y:\tau;\Delta\vdash t_2:\rho}{\Gamma;\Delta\vdash \mathsf{case}(\mathsf{inl}_\tau(s),(x:\sigma;)\,t_1,(y:\tau;)\,t_2)=t_1[s/x]}$$

$$\frac{\Gamma;\vdash s:\tau \qquad \Gamma,x:\sigma;\Delta\vdash t_1:\rho \qquad \Gamma,y:\tau;\Delta\vdash t_2:\rho}{\Gamma;\Delta\vdash \mathsf{case}(\mathsf{inr}_\sigma(s),(x:\sigma;)\,t_1,(y:\tau;)\,t_2)=t_2[s/y]}$$

Since the variables x and y in the terms t_1 and t_2 are normal, the term s must be normal as well so that the substitution is safe and as a consequence well-typed.

The beta rules translate into the following properties of $inl_{a,b}, inr_{a,b}$ and $\{-|-\}$:

$$[\Gamma,x:a;\Delta]\ \{f|g\}(\bar{x},inl_{a,b}(x;);\bar{y})=f$$
$$[\Gamma,y:b;\Delta]\ \{f|g\}(\bar{x},inr_{a,b}(y;);\bar{y})=g$$

where $f\in\mathbf{M}([\Gamma,x:a;\Delta],c), g\in\mathbf{M}([\Gamma,x:b;\Delta],c)$. Together these two properties mean that $\{-|-\}$ has an inverse given by composition with *inl* and *inr*. This $\{f|g\}$ is unique if we require the following eta-equality to hold:

$$\frac{\Gamma;\vdash s:\sigma+\tau \qquad \Gamma,z:\sigma+\tau;\Delta\vdash t:\rho}{\Gamma;\Delta\vdash \mathsf{case}(s,(x:\sigma;)\,t[\mathsf{inl}_\tau(x)/z],(y:\tau;)\,t[\mathsf{inr}_\sigma(y)/z])=t[s/z]}$$

This equality can be written as $[\Gamma,z:a+b;\Delta]\ \{t_1|t_2\}=t$ where $t\in\mathbf{M}([\Gamma,z:a+b;\Delta],c)$

$$[\Gamma,x:a;\Delta]\ t_1=t(\bar{x},inl_{a,b}(x;);\bar{y}) \qquad [\Gamma,y:b;\Delta]\ t_2=t(\bar{x},inl_{a,b}(y;);\bar{y})$$

Suppose that $[f|g]\in\mathbf{M}([\Gamma,z:a+b;\Delta],c)$ satisfies beta-equalities. Then using eta-equality we can show that $[f|g]=\{f|g\}$:

$$\{f|g\}=\{[f|g](\bar{x},inl_{a,b}(x;);\bar{y})|[f|g](\bar{x},inr_{a,b}(y;);\bar{y})\}=[f|g]$$

The uniqueness property also ensures that $\{-|-\}$ automatically satisfies the naturality requirement we imposed on it.

So the semantics of normal disjoint union types can be given using the following structure:

Definition 4.3.1 (Normal coproduct in II-multicategory)**.**
Let $\mathbf{M}$ be an II-multicategory. Let $a,b\in|\mathbf{M}|$. A **normal coproduct** of a and b is an object $a+b$ together with two multiarrows

$$inl_{a,b}:[x:a;]\to a+b \qquad inr_{a,b}:[x:b;]\to a+b$$

such that for any pair of multiarrows

$$f : [\Gamma, x : a; \Delta] \to c \qquad g : [\Gamma, y : b; \Delta] \to c$$

there exists a unique multiarrow $\{f|g\} : [\Gamma, x : a+b; \Delta] \to c$ such that

$$[\Gamma, x : a; \Delta]\ \{f|g\}(\bar{x}, inl_{a,b}(x;)\,; \bar{y}) = f(\bar{x}, x\,; \bar{y})$$
$$[\Gamma, y : b; \Delta]\ \{f|g\}(\bar{x}, inr_{a,b}(y;)\,; \bar{y}) = g(\bar{x}, y\,; \bar{y})$$

Given an II-multicategory with normal coproducts we extend the notion of dual-context structure for a signature Σ interpreting disjoint union type $[\![\sigma + \tau]\!]$ as $[\![\sigma]\!] + [\![\tau]\!]$. Then we extend the definition of $[\![\Gamma\,;\Delta \vdash t : \rho]\!]$ to handle the new kinds of terms:

Definition 4.3.2.

$$[\![\mathsf{inl}_\sigma(t)]\!] \stackrel{\text{def}}{=} inl_{[\![\sigma]\!],[\![\tau]\!]}([\![t]\!];)$$
$$[\![\mathsf{inr}_\tau(t)]\!] \stackrel{\text{def}}{=} inr_{[\![\sigma]\!],[\![\tau]\!]}([\![t]\!];)$$
$$[\![\Gamma\,;\Delta \vdash \mathsf{case}(s, (x : \sigma;)\, t_1, (y : \tau;)\, t_2)]\!] \stackrel{\text{def}}{=} \{[\![t_1]\!]|[\![t_2]\!]\}(\bar{x}, [\![s]\!]\,; \bar{y})$$

In the normal sums we restricted scope to normal terms. Another variant of sums called **safe** can defined by the following typing rules:

$$\frac{\Gamma\,;\Delta \vdash t : \sigma}{\Gamma\,;\Delta \vdash \mathsf{inl}_\tau(t) : \sigma+\tau} \qquad \frac{\Gamma\,;\Delta \vdash t : \tau}{\Gamma\,;\Delta \vdash \mathsf{inr}_\sigma(t) : \sigma+\tau}$$

$$\frac{\Gamma\,;\Delta \vdash s : \sigma+\tau \qquad \Gamma\,;\Delta, x : \sigma \vdash t_1 : \rho \qquad \Gamma\,;\Delta, y : \tau \vdash t_2 : \rho}{\Gamma\,;\Delta \vdash \mathsf{case}(s, (;x : \sigma)\, t_1, (;y : \tau)\, t_2) : \rho}$$

Comparing these rules with the corresponding rules for the normal sums we note that the introduction rules are no longer restricted to normal term and in the elimination rule the term s is arbitrary and the variables x and y are safe. These changes are propagated to the basic equalities, which look as follows:

$$\frac{\Gamma\,;\Delta \vdash s : \sigma \qquad \Gamma\,;\Delta, x : \sigma \vdash t_1 : \rho \qquad \Gamma\,;\Delta, y : \tau \vdash t_2 : \rho}{\Gamma\,;\Delta \vdash \mathsf{case}(\mathsf{inl}_\tau(s), (;x : \sigma)\, t_1, (;y : \tau)\, t_2) = t_1[s/x]}$$

$$\frac{\Gamma\,;\Delta \vdash s : \tau \qquad \Gamma\,;\Delta, x : \sigma \vdash t_1 : \rho \qquad \Gamma\,;\Delta, y : \tau \vdash t_2 : \rho}{\Gamma\,;\Delta \vdash \mathsf{case}(\mathsf{inr}_\sigma(s), (;x : \sigma)\, t_1, (;y : \tau)\, t_2) = t_2[s/y]}$$

$$\frac{\Gamma\,;\Delta \vdash s : \sigma+\tau \qquad \Gamma\,;\Delta, z : \sigma+\tau \vdash t : \rho}{\Gamma\,;\Delta \vdash \mathsf{case}(s, (;x : \sigma)\, t[\mathsf{inl}_\tau(x)/z], (;y : \tau)\, t[\mathsf{inr}_\sigma(y)/z]) = t[s/z]}$$

The systematic development of the interpretation for the safe sums is very similar to normal case so we omit it. In the end we obtain the following structure needed for the sound interpretation of safe sums:

Definition 4.3.3 (Safe coproduct in II-multicategory)**.**
Let $\mathbf{M}$ be an II-multicategory. Let $a, b \in |\mathbf{M}|$. A **safe coproduct** of a and b is an object $a+b$ together with two multiarrows

$$inl_{a,b} : [;x:a] \to a+b \qquad inr_{a,b} : [;x:b] \to a+b$$

such that for any pair of multiarrows

$$f : [\Gamma;\Delta,x:a] \to c \qquad g : [\Gamma;\Delta,y:b] \to c$$

there exists a unique multiarrow $\{f|g\} : [\Gamma;\Delta,z:a+b] \to c$ such that:

$$[\Gamma;\Delta,x:a]\ \{f|g\}(\bar{x};\bar{y},inl_{a,b}(;x)) = f$$
$$[\Gamma;\Delta,y:b]\ \{f|g\}(\bar{x};\bar{y},inr_{a,b}(;y)) = g$$

Given an II-multicategory $\mathbf{M}$ with safe coproducts we interpret safe sums in the following way:

Definition 4.3.4 (Interpretation of safe sums)**.**

$$[\![\mathsf{inl}_\sigma(t)]\!] \stackrel{\text{def}}{=} inl_{[\![\sigma]\!],[\![\tau]\!]}(;[\![t]\!])$$
$$[\![\mathsf{inr}_\tau(t)]\!] \stackrel{\text{def}}{=} inr_{[\![\sigma]\!],[\![\tau]\!]}(;[\![t]\!])$$
$$[\![\Gamma\,;\Delta \vdash \mathsf{case}(s,(;x:\sigma)\,t_1,(;y:\tau)\,t_2)]\!] \stackrel{\text{def}}{=} \{[\![t_1]\!]|[\![t_2]\!]\}(\bar{x};\bar{y},[\![s]\!])$$

This interpretation is sound with respect to beta and eta equalities for safe sums.

It is possible to define a third variant of sums which will be called **uniform**, which are essentially a combination of normal and safe sums. The typing rules for uniform sums are:

$$\frac{\Gamma;\Delta \vdash t:\sigma}{\Gamma;\Delta \vdash \mathsf{inl}_\tau(t):\sigma+\tau} \qquad \frac{\Gamma;\Delta \vdash t:\tau}{\Gamma;\Delta \vdash \mathsf{inr}_\sigma(t):\sigma+\tau}$$

$$\frac{\Gamma;\vdash s:\sigma+\tau \quad ; \quad \Gamma,x:\sigma;\Delta \vdash t_1:\rho \qquad \Gamma,y:\tau;\Delta \vdash t_2:\rho}{\Gamma;\Delta \vdash \mathsf{case}(s,(x:\sigma;)\,t_1,(y:\tau;)\,t_2):\rho}$$

$$\frac{\Gamma;\Delta \vdash s:\sigma+\tau \qquad \Gamma;\Delta,x:\sigma \vdash t_1:\rho \qquad \Gamma;\Delta,y:\tau \vdash t_2:\rho}{\Gamma;\Delta \vdash \mathsf{case}(s,(;x:\sigma)\,t_1,(;y:\tau)\,t_2):\rho}$$

with the following basic equalities:

$$\frac{\Gamma;\vdash s:\sigma \qquad \Gamma,x:\sigma;\Delta \vdash t_1:\rho \qquad \Gamma,y:\tau;\Delta \vdash t_2:\rho}{\Gamma;\Delta \vdash \mathsf{case}(\mathsf{inl}_\tau(s),(x:\sigma;)\,t_1,(y:\tau;)\,t_2) = t_1[s/x]}$$

$$\frac{\Gamma;\vdash s:\tau \qquad \Gamma,x:\sigma;\Delta\vdash t_1:\rho \qquad \Gamma,y:\tau;\Delta\vdash t_2:\rho}{\Gamma;\Delta\vdash \mathsf{case}(\mathsf{inr}_\sigma(s),(x:\sigma;)\,t_1,(y:\tau;)\,t_2)=t_2[s/y]}$$

$$\frac{\Gamma;\Delta\vdash s:\sigma \qquad \Gamma;\Delta,x:\sigma\vdash t_1:\rho \qquad \Gamma;\Delta,y:\tau\vdash t_2:\rho}{\Gamma;\Delta\vdash \mathsf{case}(\mathsf{inl}_\tau(s),(;x:\sigma)\,t_1,(;y:\tau)\,t_2)=t_1[s/x]}$$

$$\frac{\Gamma;\Delta\vdash s:\tau \qquad \Gamma;\Delta,x:\sigma\vdash t_1:\rho \qquad \Gamma;\Delta,y:\tau\vdash t_2:\rho}{\Gamma;\Delta\vdash \mathsf{case}(\mathsf{inr}_\sigma(s),(;x:\sigma)\,t_1,(;y:\tau)\,t_2)=t_2[s/y]}$$

$$\frac{\Gamma;\vdash s:\sigma+\tau \qquad \Gamma,z:\sigma+\tau;\Delta\vdash t:\rho}{\Gamma;\Delta\vdash \mathsf{case}(s,(x:\sigma;)\,t[\mathsf{inl}_\tau(x)/z],(y:\tau;)\,t[\mathsf{inr}_\sigma(y)/z])=t[s/z]}$$

$$\frac{\Gamma;\Delta\vdash s:\sigma+\tau \qquad \Gamma;\Delta,z:\sigma+\tau\vdash t:\rho}{\Gamma;\Delta\vdash \mathsf{case}(s,(;x:\sigma)\,t[\mathsf{inl}_\tau(x)/z],(;y:\tau)\,t[\mathsf{inr}_\sigma(y)/z])=t[s/z]}$$

We need the following structure on an Π-multicategory for the sound interpretation of uniform sums:

Definition 4.3.5 (Uniform coproduct)**.**
Let $\mathbf{M}$ be an Π-multicategory. Let $a,b\in|\mathbf{M}|$. A **uniform coproduct** of a and b is an object $a+b$ together with two multiarrows

$$inl_{a,b}:[;x:a]\to a+b \qquad inr_{a,b}:[;x:b]\to a+b$$

such that for any pair of multiarrows

$$f:[\Gamma;\Delta,x:a]\to c \qquad g:[\Gamma;\Delta,y:b]\to c$$

there exists a unique multiarrow $\{f|g\}:[\Gamma;\Delta,z:a+b]\to c$ such that

$$[\Gamma;\Delta,u:a]\ f|g(\bar{x};\bar{y},inl_{a,b}(;u))=f(\bar{x};\bar{y},u)$$
$$[\Gamma;\Delta,v:b]\ f|g(\bar{x};\bar{y},inr_{a,b}(;v))=g(\bar{x};\bar{y},v)$$

and for any pair of multiarrows

$$s:[\Gamma,x:a;\Delta]\to c \qquad t:[\Gamma,y:b;\Delta]\to c$$

there exists a unique multiarrow $s|t:[\Gamma,z:a+b;\Delta]\to b$ such that

$$[\Gamma,u:a;\Delta]\ s|t(\bar{x},inl_{a,b}(;u);\bar{y})=s(\bar{x};\bar{y},u)$$
$$[\Gamma,v:b;\Delta]\ s|t(\bar{x},inr_{a,b}(;v);\bar{y})=t(\bar{x};\bar{y},v)$$

This induces the following bijections between the set of multiarrow:

$$\mathbf{M}([\Gamma;\Delta,y:a],c)\times\mathbf{M}([\Gamma;\Delta,y:b],c)\cong\mathbf{M}([\Gamma;\Delta,y:a+b],c)$$
$$\mathbf{M}([\Gamma,x:a;\Delta],c)\times\mathbf{M}([\Gamma,x:b;\Delta],c)\cong\mathbf{M}([\Gamma,x:a+b;\Delta],c)$$

Using the uniform coproduct structure we can define the interpretation as follows:

Definition 4.3.6.
Given an II-multicategory with uniform coproducts the interpretation of the $\sigma+\tau$ terms is defined as follows:

$$[\![\mathsf{inl}_\tau(t)]\!] \stackrel{\text{def}}{=} inl_{[\![\sigma]\!],[\![\tau]\!]}(;[\![t]\!])$$
$$[\![\mathsf{inr}_\sigma(t)]\!] \stackrel{\text{def}}{=} inr_{[\![\sigma]\!],[\![\tau]\!]}(;[\![t]\!])$$
$$[\![\mathsf{case}(s,(x:\sigma;)\,t_1,(y:\tau;)\,t_2)]\!] \stackrel{\text{def}}{=} \{[\![t_1]\!]|[\![t_2]\!]\}(\bar{x},[\![s]\!];\bar{y})$$
$$[\![\mathsf{case}(s,(;x:\sigma)\,t_1,(;y:\tau)\,t_2)]\!] \stackrel{\text{def}}{=} \{[\![t_1]\!]|[\![t_2]\!]\}(\bar{x};\bar{y},[\![s]\!])$$

In this chapter we have extended the $\mathbb{III}(\Sigma)$ and the $\mathbb{IIL}(\Sigma)$ calculi with products, coproducts and unit type. Unlike the traditional categorical case the presence of dual-contexts allowed us to consider different versions of these extensions. For each extensions we provided a sound interpretation using extended notions of II- and IL-multicategories. In II- case we have have also shown that these extensions can be regarded as a special case of the general notion of binding operator.

Chapter 5

Safe dual-context lists

In this chapter we shall extend the $\mathbb{III}(\Sigma)$ calculus with safe dual-context list types which contain modified safe recursion scheme. We will give a sound interpretation for this type constructor using a notion of safe list object. Safe natural number object as an important special case of safe list objects will be considered. We will the extend safe dual-context list types with additional rule, which would allow definitions by flat recursion. We show that in any model of $\mathbb{III}(\Sigma)$ with extended safe list types every function from the system $\mathfrak{B}$ is representable. As a result of this effort, we get some insights about the fundamental roles of various components of system $\mathfrak{B}$, in particular explaining the choice of some of the initial functions.

5.1 Single-context case

In this section we recall typing rules, basic equalities and interpretation for the ordinary single-context list type $\sigma\ list$ [Pit95]. Such list type is parameterized with the type σ, which is a type of elements of lists. The introduction rules for the $\sigma\ list$ type can be given as follows:

$$\frac{}{\Gamma \vdash \mathsf{nil}_\sigma : \sigma\ list}\ \textbf{I-Nil} \qquad \frac{\Gamma \vdash h : \sigma \quad \Gamma \vdash t : \sigma\ list}{\Gamma \vdash \mathsf{cons}(h,t) : \sigma\ list}\ \textbf{I-Cons}$$

The term nil_σ represents the empty list and the term $\mathsf{cons}(h,t)$ represents a list obtained from a given list, represented by the term t, by adding an element represent by the term h.

The elimination rule for $\sigma\ list$ is given as follows:

$$\frac{\Gamma, x : \sigma, y : \sigma\ list, z : \tau \vdash s : \tau \qquad \Gamma \vdash l : \sigma\ list \qquad \Gamma \vdash n : \tau}{\Gamma \vdash \mathsf{listrec}(l,n,(x,y,z)s) : \tau}\ \textbf{I-Rec}$$

Recall that we use notation $(x,y,z)s$ to denote a binding of the variables x,y,z in the term s. The term $\mathsf{listrec}(l,n,(x,y,z)s) : \tau$ represents an application of a function f to a list, represented by the term l. The function f is defined by the following recursion scheme:

$$f(nil,\bar{x}) = n(\bar{x})$$
$$f(cons(h,t),\bar{x}) = s(\bar{x},h,t,f(t,\bar{x}))$$

This interpretation of $\mathsf{listrec}(l,n,(x,y,z)s)$ is reflected in the following basic equalities for $\sigma\ list$, which essentially code the above recursion scheme:

$$\frac{\Gamma \vdash n : \tau \qquad \Gamma, u : \sigma, v : \sigma\ list, w : \tau \vdash s : \tau}{\Gamma \vdash \mathsf{listrec}(\mathsf{nil}_\sigma, n, (u,v,w)s) = n : \tau}$$

$$\frac{\Gamma \vdash n : \tau \qquad \Gamma, u : \sigma, v : \sigma\ list, w : \tau \vdash s : \tau \qquad \Gamma \vdash h : \sigma \qquad \Gamma \vdash t : \sigma\ list}{\Gamma \vdash \mathsf{listrec}(\mathsf{cons}(h,t), n, (u,v,w)s) = s[h/u, t/v, \mathsf{listrec}(t,n,(u,v,w)s/w)) : \tau}$$

The following basic equality ensures the uniqueness of such a function, defined by the above recursion scheme:

$$\frac{\begin{array}{c}\Gamma \vdash l : \sigma\ list \qquad \Gamma \vdash n : \tau \qquad \Gamma, u : \sigma, v : \sigma\ list, w : \tau \vdash s : \tau \\ \Gamma, v : \sigma\ list, w : \tau \vdash g : \tau \qquad \Gamma, w : \tau \vdash g[\mathsf{nil}_\sigma / v] = w \\ \Gamma, u : \sigma, v : \sigma\ list, w : \tau \vdash s[g/w] = g[\mathsf{cons}(u,v)/v] : \tau\end{array}}{\Gamma \vdash \mathsf{listrec}(l,n,(u,v,w)s) = g[l/u, n/w] : \tau}$$

In a cartesian category $\mathbf{C}$ given an interpretation of σ as an object a, $\sigma\ list$ terms can be soundly interpreted using a list object, which consists of an object L and two morphisms $nil : 1 \to L$ and $cons : a \times L \to L$ which satisfy the following universal property: for each pair of morphisms $h : I \times a \times b \to b$ and $g : I \to b$ there exists a unique morphism $f : I \times L \to b$ making the following diagrams commute:

$$\begin{array}{ccc} I \times \mathbf{1} & \xrightarrow{id_I \times nil} & I \times L \\ {\scriptstyle \pi_1}\downarrow & & \downarrow{\scriptstyle f} \\ I & \xrightarrow[g]{} & b \end{array}$$

$$\begin{array}{ccc} I \times a \times L & \xrightarrow{id_I \times cons} & I \times L \\ {\scriptstyle \langle \pi_1, \pi_2, f \circ \langle \pi_1, \pi_3 \rangle \rangle}\downarrow & & \downarrow{\scriptstyle f} \\ I \times a \times b & \xrightarrow[h]{} & b \end{array}$$

5.2 Dual-context lists in $\mathbb{III}(\Sigma)$

In this section we are going to extend the $\mathbb{III}(\Sigma)$ calculus with a dual-context list type constructor and give a sound interpretation for this extension in an II-multicategory.

There are many different ways of modifying single-context list type for the dual-context setting. Recall that the introduction rules for list type define terms, which represent empty list and a list obtained by adding a new element to a given list. The elimination rule together with the basic equalities describes recursion on lists. We will consider a version of the dual-context list type which is based on safe recursion scheme of system $\mathfrak{B}$. We shall call such list type a **safe dual-context list**.

Strictly speaking safe recursion in system $\mathfrak{B}$ was defined for binary presentation of natural numbers, which is cumbersome to work with. Leivant[Lei94] suggested to use lists of 0's and 1's instead, which seem to be a more natural approach and requires only the minor modifications of system $\mathfrak{B}$.

The safe dual-context list type will be parameterized by list elements type σ. Later we shall consider a special version of safe list type for σ being $\mathbf{1}+\mathbf{1}$, where $\mathbf{1}$ is a unit type and $+$ is a uniform sum type.

Before we give typing rules for the safe list type, recall some important features of system $\mathfrak{B}$ which was defined on page 6:

- Constant zero $zero : [;] \to N$.
- Two successor functions with safe inputs $s_0 : [;x:N] \to N$ and $s_1 : [;x:N] \to N$. In the safe list type setting we will have one successor with two safe arguments.
- In the safe recursion scheme a recursion argument is always normal, a recursive value is substituted into a safe position in step functions, which also have an additional normal input for the recursion argument:

$$f(nil,\bar{x};\bar{y}) = g(\bar{x};\bar{y})$$
$$f(0.t,\bar{x};\bar{y}) = h_0(t,\bar{x};\bar{y},f(t,\bar{x};\bar{y}))$$
$$f(1.t,\bar{x};\bar{y}) = h_1(t,\bar{x};\bar{y},f(t,\bar{x};\bar{y}))$$

For the safe list type we will use the following modified safe recursion scheme:

$$f(nil,\bar{x};\bar{y}) = g(\bar{x};\bar{y})$$
$$f(const(a,t),\bar{x};\bar{y}) = h(a,t,\bar{x};\bar{y},f(t,\bar{x};\bar{y}))$$

Now we are ready to define typing rules for the safe dual-context list types. Given an element type σ the safe list type will be denoted as $\sigma\ list$. The introduction rules for this type are as follows:

$$\frac{}{\Gamma;\Delta \vdash \mathsf{nil}_\sigma : \sigma\ list}\ \textbf{II-Nil} \qquad \frac{\Gamma;\Delta \vdash h:\sigma \qquad \Gamma;\Delta \vdash t:\sigma\ list}{\Gamma;\Delta \vdash \mathsf{cons}_\sigma(h,t):\sigma\ list}\ \textbf{II-Cons}$$

The term nil_σ represents the empty list and the term $\mathsf{cons}_\sigma(h,t)$ represents a list obtained from a given list, represented by the term t, by adding an element represent by the term h. Note that neither h nor t are required to be normal.

The elimination rule for $\sigma\ list$ is given as follows:

$$\frac{\Gamma,x:\sigma,y:\sigma\ list\,;\Delta,z:\tau \vdash s:\tau \qquad \Gamma;\vdash l:\sigma\ list \qquad \Gamma;\Delta \vdash n:\tau}{\Gamma;\Delta \vdash \mathsf{listrec}(l,n,(x:\sigma,y:\sigma\ list;z:\tau)s):\tau}\ \textbf{II-Rec}$$

The term $\mathsf{listrec}(l,n,(x,y;z)s):\tau$ represents an application of the function f, defined by safe recursions above, to a list, represented by the term l. The term s represent the step function h The normal variables x and y in s are used for passing the recursion argument and the safe variable z for passing the recursive value. The term n represents the function g.

Proposition 5.2.1. *In the $\mathbb{III}(\Sigma)$ calculus extended with the **II-Nil**, **II-Cons** and **II-Rec** rules the **II-Subst** rule is admissible.*

Proof. The typing rules **II-Nil**, **II-Cons** and **II-Rec** are instances of general II-binding operators, so by the theorem 4.1.2 the **II-Subst** rule is admissible in $\mathbb{III}(\Sigma)$ extended with these rules. □

The basic equality rules for the $\sigma\ list$ type are given as follows:

$$\frac{\Gamma;\Delta \vdash n:\tau \qquad \Gamma,x:\sigma,y:\sigma\ list\,;\Delta,z:\tau \vdash s:\tau}{\Gamma;\Delta \vdash \mathsf{listrec}(\mathsf{nil}_\sigma,n,(x,y;z)s) = n:\tau}$$

$$\frac{\begin{array}{c}\Gamma,x:\sigma,y:\sigma\ list\,;\Delta,z:\tau \vdash s:\tau \\ \Gamma;\Delta \vdash n:\tau \qquad \Gamma;\vdash h:\sigma \qquad \Gamma;\vdash t:\sigma\ list\end{array}}{\Gamma;\Delta \vdash \mathsf{listrec}(\mathsf{cons}_\sigma(h,t),n,(x,y;z)s) = s[h/x,t/y,\mathsf{listrec}(t,n,(x,y;z)s)/z):\tau}$$

The first equality corresponds to the base case in the above recursion scheme, while the second equality corresponds to the inductive case. Note that the substitutions h/x and t/y are valid since both h and t are required to be normal.

The above two equalities define 'weak' dual-context safe list, which postulate the existence of the function, defined by the above safe recursion on lists. Adding the following equality ensures the uniqueness of this function:

$$\frac{\begin{array}{c}\Gamma;\vdash l:\sigma\ list \qquad \Gamma;\Delta\vdash n:\tau \qquad \Gamma,x:\sigma,y:\sigma\ list\,;\Delta,z:\tau\vdash s:\tau \\ \Gamma,u:\sigma\ list\,;\Delta,v:\tau\vdash g:\tau \qquad \Gamma;\Delta,v:\tau\vdash g[\mathsf{nil}_\sigma/u]=z \\ \Gamma,x:\sigma,y:\sigma\ list\,;\Delta,z:\tau\vdash s[g[y/u,z/v]/z]=g[\mathsf{cons}(x,y)/u,z/v]:\tau\end{array}}{\Gamma;\Delta\vdash \mathsf{listrec}(l,n,(x,y;z)s)=g[l/u,n/v]:\tau}$$

Using the method followed in the previous chapter we can show that the interpretation of $\sigma\ list$ terms in an II-multicategory requires the following structure:

Definition 5.2.2 (Safe list object)**.**
Let $\mathbf{M}$ be an II-multicategory. A **safe list object** for an object a is an object $L(a)$ together with multiarrows $[;]\xrightarrow{nil_a} L$ and $[;x:c,y:L(a)]\xrightarrow{cons_a} L(a)$ such that for any pair of multiarrows $g\in\mathbf{M}([\Gamma\,;\Delta],b)$ and $h\in\mathbf{M}([\Gamma,x:c,y:L(a)\,;\Delta,z:b],b)$ there exists a unique multiarrow $sr_a(g,h)\in\mathbf{M}([\Gamma,x:L(a)\,;\Delta],b)$ such that

$$[\Gamma\,;\Delta]\quad sr_a(g,h)(\bar{x},nil_A;\bar{y})=g$$

$$[\Gamma,x:c,y:L(a)\,;\Delta]\quad sr_a(g,h)(\bar{x},cons_a(;x,y);\bar{y})=h(\bar{x},x,y;\bar{y},sr_a(g,h)(\bar{x},y;\bar{y}))$$

The following lemma shows $sr(g,h)$ is natural in $\Gamma\,;\Delta]$:

Proposition 5.2.3. *Let $L(c)$ be a safe list object for an object c in an II-multicategory $\mathbf{M}$. For any pair of multiarrows $g\in\mathbf{M}([\Gamma\,;\Delta],b)$ and $h\in\mathbf{M}([\Gamma,x:c,y:L(c)\,;\Delta,z:b],b)$ and any family $s_i:[\Pi;]\to a_i$ and $t_j:[\Pi;\Theta]\to b_j$ the following holds:*

$$[\Pi,x:L(c)\,;\Theta]\quad sr_a(g,h)(s_1(\bar{u};),\ldots,s_n(\bar{u};),x;t_1(\bar{u};\bar{v}),\ldots,t_m(\bar{u};\bar{v}))=sr_a(g',h')$$

where g' is defined as $[\Pi,\Theta]\ g(s_1(\bar{u};),\ldots,s_n(\bar{u};);t_1(\bar{u};\bar{v}),\ldots,t_m(\bar{u};\bar{v}))$ and h' is defined as $[\Pi,x:c,y:L;\Theta,z:b]\ h(s_1(\bar{u};),\ldots,s_n(\bar{u};),x,y;t_1(\bar{u};\bar{v}),\ldots,t_m(\bar{u};\bar{v}),z)$.

Proof. Using the associativity of a II-composition and the properties of $sr(g,h)$ we can show that the following holds:

$$[\Pi;\Theta]\quad sr_a(g,h)(\bar{s},nil_a;\bar{t})=g(\bar{s};\bar{t})=g'$$

$$[\Pi,x:c,y:L;\Theta]\quad \begin{array}{l}sr_a(g,h)(\bar{s},cons_a(;x,y);\bar{t})=h(\bar{s},x,y;\bar{t},sr_a(g,h)(\bar{s};\bar{t},y))=\\ h'(\bar{u},x,y;\bar{v},sr(g',h'))\end{array}$$

From this using the uniqueness property of $sr_a(g',h')$ we conclude that

$$[\Pi,x:L(c)\,;\Theta]\quad sr_a(g,h)(\bar{s},x;\bar{t})=sr_a(g',h')$$

□

Definition 5.2.4 (Interpretation of safe list terms)**.**
Given an II-multicategory **M** with safe list object for the object $[\![\sigma]\!]$ we define the interpretation of σ *list* terms as follows:

$$[\![\sigma\ list]\!] \stackrel{\text{def}}{=} L([\![\sigma]\!])$$
$$[\![\Gamma\,;\Delta \vdash \mathsf{nil}_\sigma]\!] \stackrel{\text{def}}{=} nil_{[\![\sigma]\!]}$$
$$[\![\Gamma\,;\Delta \vdash \mathsf{cons}_\sigma(h,t)]\!] \stackrel{\text{def}}{=} cons_{[\![\sigma]\!]}(;[\![h]\!],[\![t]\!])$$
$$[\![\Gamma\,;\Delta \vdash \mathsf{listrec}(l,n,s)]\!] \stackrel{\text{def}}{=} sr_{[\![\sigma]\!]}([\![n]\!],[\![s]\!])(\bar{x},[\![l]\!];\bar{y})$$

This interpretation is sound with respect to the basic equalities of σ *list* types:

$$[\![\mathsf{listrec}(\mathsf{nil}_{[\![\sigma]\!]},n,s)]\!] = sr_{[\![\sigma]\!]}([\![n]\!],[\![s]\!])(\bar{x},[\![nil_{[\![\sigma]\!]}]\!]\,;\bar{y}) = [\![n]\!]$$
$$[\![\Gamma\,;\Delta \vdash \mathsf{listrec}(\mathsf{cons}(h,t),n,s)]\!] = sr_{[\![\sigma]\!]}([\![n]\!],[\![s]\!])(\bar{x},cons_{[\![\sigma]\!]}(;[\![h]\!],[\![t]\!])\,;\bar{y},[\![n]\!]) =$$
$$[\![s]\!](\bar{x},[\![h]\!],[\![t]\!]\,;\bar{y},sr_{[\![\sigma]\!]}([\![n]\!],[\![s]\!])(\bar{x},[\![t]\!]\,;\bar{y},[\![n]\!])$$

For the remaining equality we have that

$$[\![g]\!](\bar{x},nil_{[\![\sigma]\!]}\,;\bar{y},[\![n]\!]) = [\![n]\!]$$
$$[\![g]\!](\bar{x},cons_{[\![\sigma]\!]}(;x,y)\,;\bar{y},[\![n]\!]) = [\![h]\!](\bar{x},x,y;\bar{y},g(\bar{x},y;\bar{y},[\![n]\!]))$$

which means that the interpretation of $g[n/w]$ satisfies two properties of safe lists therefore it must be equal to $sr_{[\![\sigma]\!]}([\![n]\!],[\![s]\!])$. Next lemma shows that any two safe list objects in an II-multicategory are isomorphic:

Proposition 5.2.5. *Let* **M** *be an II-multicategory. Assume that* $nil_a : [;] \to L(a), cons_a : [;x : L(a)] \to L(a)$ *and* $nil'_a : [;] \to L'(a), cons'_a : [;x : L'(a)] \to L'(a)$ *are safe list objects for an object a. Then* $L(a)$ *and* $L'(a)$ *are isomorphic via normal isomorphism, which means that there exists two multiarrows*

$$i : [x : L(a);] \to L'(a) \qquad\qquad i' : [x : L'(a);] \to L(a)$$

such that $[x : L'(a);]\ i(i'(x;);) = x$ *and* $[x : L(a);]\ i'(i(x;);) = x$.

Proof. Consider $[;] \xrightarrow{nil'_a} L'(a)$ and $[x : a, y : L(a); z : L'(a)] \xrightarrow{cons'_a(;x,z)} L'(a)$. Since $L(a)$ is a safe list object then for such pair of multiarrows there exists $i : [x : L(a);] \to L'(a)$ satisfying the following properties:

$$[;]\ i(nil_a;) = nil'_a$$
$$[x : a, y : L(a);]\ i(cons_a(;x,y);) = cons'_a(;x,i(y;))$$

Similarly considering nil_a and $cons_a$ we get a multiarrow $i' : [x : L'(a);] \to L(a)$ satisfying the following properties:

$$[;]\ i'(nil'_a;) = nil_a$$
$$[x : a, y : L'(a);]\ i'(cons'_a(;x,y);) = cons_a(;x,i'(y;))$$

Take $[;] \xrightarrow{nil_a} L(a)$ and $[x : a, y : L(a); z : L(a)] \xrightarrow{cons_a(;x,z)} L(a)$. For such pair of multiarrows there exists a unique multiarrow $[x : L(a);] \xrightarrow{f} L(a)$ with the following properties:

$$[;]\ f(nil_a;) = nil_a$$
$$[x : a, y : L(a);]\ f(cons_a(;x,y);) = cons_a(;x,f(y;))$$

Obviously $[x : L(a);]\ f = x$ since π_x satisfies the above conditions. But on the other hand if we consider the composition $[x : L(a);]\ i'(i(x;);)$ then we can show that it satisfies these conditions as well:

$$[;]\ i'(i(nil_a;);) = i'(nil'_a;) = nil_a$$
$$[x : a, y : L(a);]\ i'(i(cons_a(;x,y););) = i'(cons'_a(;x,i(y;));) = cons_a(;x,i'(i(y;);))$$

Since f is unique we conclude that $[x : L(a);]\ i'(i(x;);) = x$. Similarly we can show that $[x : L'(a);]\ i(i'(x;);) = x$. □

The following lemma shows that using a safe list object we can construct a normal multiarrow which represents predecessor function.

Proposition 5.2.6. *Let* **M** *be an II-multicategory and $L(a)$ with multiarrows nil_a and $cons_a$ be a safe list object for an object a. Then there exists a unique multiarrow*

$$\iota : [x : L(a);] \to L(a)\ ]$$

such that the following holds:

$$[;]\ \iota(nil_a;) = nil_a$$
$$[x : a, y : L(a);]\ \iota(cons_a(;x,y);) = y$$

Proof. Using the safe list property for a pair of multiarrows nil_a and π_y we get ι satisfying the desired properties. □

We have defined the extension of the $\mathbb{III}(\Sigma)$ calculus with the safe dual-context list type. In case of $\mathbb{III}(\Sigma)$ extended with safe unity type and uniform sums we can consider

a special variant of safe list type with element of type $\mathbf{1}+\mathbf{1}$. We shall call such safe list type as **safe natural number type** and denote it as N. The following typing rules and equalities for N can be derived systematically from the rules of safe list type, unit type and uniform sums:

$$\frac{}{\Gamma;\Delta\vdash \mathsf{nil}:N} \qquad \frac{\Gamma;\Delta\vdash t:N}{\Gamma;\Delta\vdash \mathsf{s}_0(t):N} \qquad \frac{\Gamma;\Delta\vdash t:N}{\Gamma;\Delta\vdash \mathsf{s}_1(t):N}$$

$$\frac{\Gamma;\vdash l:N \qquad \Gamma;\Delta\vdash n:\tau \qquad \Gamma,x:N;\Delta,y:\tau\vdash h_0:\tau \qquad \Gamma,x:N;\Delta,y:\tau\vdash h_1:\tau}{\Gamma;\Delta\vdash \mathsf{bl}(l,n,(x;y)h_0,(x;y)h_1):\tau}$$

$$\frac{\Gamma;\Delta\vdash n:\tau \quad \Gamma,x:N;\Delta,y:\tau\vdash h_0:\tau \quad \Gamma,x:N;\Delta,y:\tau\vdash h_1:\tau}{\Gamma;\Delta\vdash \mathsf{bl}(\mathsf{nil},n,(x;y)h_0,(x;y)h_1)=n:\tau}$$

$$\frac{\Gamma;\Delta\vdash n:\tau \qquad \Gamma;\vdash t:N \qquad \Gamma,x:N;\Delta,y:\tau\vdash h_0:\tau \qquad \Gamma,x:N;\Delta,y:\tau\vdash h_1:\tau}{\Gamma;\Delta\vdash \mathsf{bl}(\mathsf{s}_0(t),n,(x;y)h_0,(x;y)h_1)=h_0[t/x,\mathsf{bl}(t,n,(x;y)h_0,(x;y)h_1)/y]:\tau}$$

$$\frac{\Gamma;\Delta\vdash n:\tau \qquad \Gamma;\vdash t:N \qquad \Gamma,x:N;\Delta,y:\tau\vdash h_0:\tau \qquad \Gamma,x:N;\Delta,y:\tau\vdash h_1:\tau}{\Gamma;\Delta\vdash \mathsf{bl}(\mathsf{s}_1(t),n,(x;y)h_0,(x;y)h_1)=h_1[t/x,\mathsf{bl}(t,n,(x;y)h_0,(x;y)h_1)/y]:\tau}$$

$$\frac{\begin{array}{c}\Gamma;\vdash l:N \qquad \Gamma;\Delta\vdash n:\tau \\ \Gamma,x:\mathbf{2}\,list;\Delta,y:\tau\vdash h_0:\tau \quad \Gamma,x:N;\Delta,y:\tau\vdash 1_0:\tau \\ \Gamma,x:\mathbf{2}\,list;\Delta,y:\tau\vdash g:\tau \qquad \Gamma;\Delta,z:\tau\vdash g(\bar{x},\mathsf{nil};\bar{y},z)=z \\ \Gamma,x:\mathbf{2}\,list;\Delta,y:\tau\vdash h_0(\bar{x},x;\bar{y},g(\bar{x},x;\bar{y},y))=g(\bar{x},\mathsf{s}_0(x);\bar{y},w):\tau \\ \Gamma,x:\mathbf{2}\,list;\Delta,y:\tau\vdash h_1(\bar{x},x;\bar{y},g(\bar{x},x;\bar{y},y))=g(\bar{x},\mathsf{s}_1(x);\bar{y},w):\tau\end{array}}{\Gamma;\Delta\vdash \mathsf{bl}(l,n,(x;y)h_0,(x;y)h_1)=g[l/x,n/y]:\tau}$$

The above typing rules can be added to the $\mathbb{III}(\Sigma)$ calculus directly, without having to do sum and unit type extensions. Semantics of this extensions can be defined using the following notion:

Definition 5.2.7 (Safe natural number object)**.**
Let $\mathbf{M}$ be an II-multicategory. A **safe natural number object** (SNNO for short) is an object I together with multiarrows $[;]\xrightarrow{z} I$, $[;x:I]\xrightarrow{s_0} I$ and $[;x:I]\xrightarrow{s_1} I$ such that for any triple of multiarrows

$$g\in\mathbf{M}([\Gamma;\Delta],b) \qquad h_0\in\mathbf{M}([\Gamma,x:I;\Delta,y:b],b) \qquad h_1\in\mathbf{M}([\Gamma,x:I;\Delta,y:b],b)$$

there exists a unique multiarrow $sr(g,h_0,h_1) \in \mathbf{M}([\Gamma,x:I;\Delta],b)$ such that the following holds:

$$[\Gamma;\Delta]\ sr(g,h_0,h_1)(\bar{x},z;\bar{y}) = g$$
$$[\Gamma,x:I;\Delta]\ sr(g,h_0,h_1)(\bar{x},s_0(;x);\bar{y}) = h_0(\bar{x},x;\bar{y},sr(g,h_0,h_1)(\bar{x},x;\bar{y}))$$
$$[\Gamma,x:I;\Delta]\ sr(g,h_0,h_1)(\bar{x},s_1(;x);\bar{y}) = h_1(\bar{x},x;\bar{y},sr(g,h_0,h_1)(\bar{x},x;\bar{y}))$$

Theorem 5.2.8. *The II-multicategory* ***B*** *has a SNNO.*

Proof. For the SNNO take $\mathbb{N} \in |\mathbf{B}|$ with the following functions:

$$z = 0 \qquad s_0(;x) = 2x+1 \qquad s_1(;x) = 2x+2$$

These functions are clearly polytime computable and can bounded as follows:

$$|z| \leq 1 \qquad |s_i(;x)| \leq 2+|x|$$

Given g, h_0 and h_1 we define $sr(g,h_0,h_1)$ using the safe recursion scheme:

$$f(\bar{x},z;\bar{y}) = g(\bar{x};\bar{y})$$
$$f(\bar{x},s_0(;n);\bar{y}) = h_0(\bar{x},n;\bar{y},f(\bar{x},n;\bar{y}))$$
$$f(\bar{x},s_1(;n);\bar{y}) = h_1(\bar{x},n;\bar{y},f(\bar{x},n;\bar{y}))$$

We shall now demonstrate that f defined in such a way can be bounded (this proof is taken from Lemma 4.1 in [BC92]). Assume that g is bounded by the a polynomial p_g. Then the following holds:

$$|f(\bar{x},z;\bar{y})| = |g(\bar{x};\bar{y}| \leq p_g(|\bar{x}|) + \max_j |y_j|$$

Assuming bounds p_{h_0} and p_{h_1} for the function h_0 and h_1 respectively we can show that

$$|f(\bar{x},s_i(;n);\bar{y})| \leq p_h(|\bar{x}|+|n|) + \max\{\max_j |y_j|, |f(\bar{x},n;\bar{y})\}|$$

where $p_h = p_{h_0} + p_{h_1}$. By induction we shall show that $p_f(x) = x \cdot p_h(x) + p_g(x)$ is indeed a bounding polynomial for f. For the initial case we have

$$|f(\bar{x},z;\bar{y})| \leq p_g(|\bar{x}|) + \max_j |y_j| \leq p_f(|\bar{x}|+|z|) + \max_j |y_j|$$

For the inductive case we assume that $|f(\bar{x},n;\bar{y})| \leq p_f(|\bar{x}|+|n|)+\max_j |y_j|$. Then for $|f(\bar{x},s_i(;n);\bar{y})|$ we have:

$$\begin{aligned}
|f(\bar{x},s_i(;n);\bar{y})| &\leq p_{h_i}(|\bar{x}|+|n|)+\max\{\max_j |y_j|, |f(\bar{x},n;\bar{y})|\} \\
&\leq p_{h_i}(|\bar{x}|+|n|)+p_f(|\bar{x}|+|n|)+\max_j |y_j| \\
&\leq p_{h_i}(|\bar{x}|+|n|)+(|\bar{x}|+|n|)\cdot p_h(|\bar{x}|+|n|)+p_g(|\bar{x}|+|n|)+\max_j |y_j| \\
&\leq (|\bar{x}|+|n|+1)\cdot p_h(|\bar{x}|+|n|)+p_g(|\bar{x}|+|n|)+\max_j |y_j| \\
&\leq (|\bar{x}|+|s_i(;n)|)\cdot p_h(|\bar{x}|+|s_i(;n)|)+p_g(|\bar{x}|+|n|)+\max_j |y_j| \\
&\leq p_f(|\bar{x}|+|s_i(;n)|)+\max_j |y_j|
\end{aligned}$$

So p_f is indeed a bounding polynomial for the function f. It can be shown that safe recursion can be executed in polynomial time if the length of result of the recursion is bounded by a polynomial and the step and the base functions are polytime [BC92]. So we have shown that $f \in \mathbf{B}([x_1:\mathbb{N},\ldots,x_n:\mathbb{N},x:\mathbb{N};y_1:\mathbb{N},\ldots,y_m:\mathbb{N}],\mathbb{N})$ which proves that $\mathbb{N}$ is a SNNO.

□

5.3 Extending safe dual-context lists

In the previous sections we extended $\mathbb{III}(\Sigma)$ with the safe dual-context list type $\sigma\ list$ and gave multicategorical semantics for such extensions. The typing rules for the safe list type were constructed following the basic principles of Bellantoni and Cook's system $\mathfrak{B}$. In particular the elimination rule for $\sigma\ list$ together with the basic the equalities associated with it were essentially describing list version of safe recursion scheme.

Consider the following typing derivation the $\mathbb{III}(\Sigma)$ calculus extended with the dual-context list types:

$$\frac{x:\sigma\ list\,;\vdash \mathsf{nil}_\sigma:\sigma\ list \qquad x:\sigma\ list\,;\vdash x:\sigma\ list \qquad x:\sigma\ list,u:\sigma,v:\sigma\ list\,;w:\sigma\ list\vdash v:\sigma\ list}{x:\sigma\ list\,;\vdash \mathsf{listrec}(x,\mathsf{nil}_\sigma,(u,v;w)v):\sigma\ list}$$

It is easy to see that the term $\mathsf{listrec}(x,\mathsf{nil}_\sigma,(u,v;w)v)$ represents a predecessor function $pred(x:\sigma\ list;):\sigma\ list$ given by the following safe recursion:

$$\begin{aligned}
&[;]\ pred(nil;) = nil \\
&[x:\sigma,y:\sigma\ list;]\ pred(cons(;x,y);) = y
\end{aligned}$$

Since in the safe recursion scheme the argument of the recursion is always normal we can not define a safe version of the predecessor function. Clearly we can not allow the argument of recursion to be safe since in such a scheme the nesting of recursions would be possible and as a result we would be able to define functions which are not polytime computable. On the other hand a safe predecessor is indeed one of the basic ingredient of system $\mathfrak{B}$ (see page 6).

In order to solve this problem we propose to extend the safe dual-context list type with the following rule:

$$\frac{\Gamma;\Delta\vdash l:\sigma\ \mathit{list} \qquad \Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,x:\sigma,y:\sigma\ \mathit{list}\vdash s:\tau}{\Gamma;\Delta\vdash \mathsf{cases}(l,n,(;x,y)s):\tau}$$

which should satisfy the following basic equalities:

$$\frac{\Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,x:\sigma,y:\sigma\ \mathit{list}\vdash s:\tau}{\Gamma;\Delta\vdash \mathsf{cases}(\mathsf{nil}_\sigma,n,(;x,y)s)=n:\tau}$$

$$\frac{\Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,x:\sigma,y:\sigma\ \mathit{list}\vdash s:\tau \qquad \Gamma;\Delta\vdash h:\sigma \qquad \Gamma;\Delta\vdash t:\sigma\ \mathit{list}}{\Gamma;\Delta\vdash \mathsf{cases}(\mathsf{cons}(h,t),n,(;x,y)s)=s[h/x,t/y]:\tau}$$

It is easy to see that this rule allows us to define functions by *flat recursion* [Lei94]:

$$[\Gamma;\Delta]\ f(\bar{x};nil,\bar{y})=g(\bar{x};\bar{y})$$

$$[\Gamma;\Delta,u:\sigma,v:\sigma\ \mathit{list}]\ f(\bar{x};cons(;u,v),\bar{y})=h(\bar{x};\bar{y},u,v)$$

Adding the following equality ensures that the function defined flat recursion is unique:

$$\frac{\begin{array}{c}\Gamma;\Delta\vdash l:\sigma\ \mathit{list} \qquad \Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,u:\sigma,v:\sigma\ \mathit{list}\vdash s:\tau\\ \Gamma;\Delta,x:\sigma\ \mathit{list}\vdash g:\tau \qquad \Gamma;\Delta\vdash g[\mathsf{nil}_\sigma/x]=n\\ \Gamma;\Delta,u:\sigma,v:\sigma\ \mathit{list}\vdash g[\mathsf{cons}(u,v)/x]=s:\tau\end{array}}{\Gamma;\Delta\vdash \mathsf{cases}(l,n,(;u,v)s)=g[l/x]:\tau}$$

Interpretation of the cases rule requires an additional property of safe list objects in a multicategory:

Definition 5.3.1.

An II-multicategory $\mathbf{M}$ with a **safe list object** L satisfies **flat recursion property** if for every pair of multiarrows $g\in\mathbf{M}([\Gamma;\Delta],b)$ and $h\in\mathbf{M}([\Gamma;\Delta x:a,y:L],b)$ there exists a unique multiarrow

$$cases(g,h)\in\mathbf{M}([\Gamma;\Delta,x:L,b)$$

satisfying the following properties:

$$[\Gamma;\Delta]\ cases(g,h)(\bar{x};\bar{y},nil)=g$$

$$[\Gamma;\Delta,u:a,v:L(a)]\ cases(g,h)(\bar{x};\Delta,cons(;u,v))=h(\bar{x};\bar{y},u,v)$$

Given an II-multicategory $\mathbf{M}$ with a safe list object L which satisfies flat recursion property we can interpret cases terms as follows:

$$[\![\mathsf{cases}(l,n,(;u,v)s)]\!] \stackrel{\text{def}}{=} cases([\![n]\!],[\![s]\!])(\bar{x};\bar{y},[\![n]\!])$$

This interpretation is obviously sound with respect to the basic equalities for the cases operator.

For the safe natural number type the same extension can be defined:

$$\frac{\Gamma;\Delta\vdash l:N \qquad \Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,x:N\vdash h_0:\tau \qquad \Gamma;\Delta,x:N\vdash h_1:\tau}{\Gamma;\Delta\vdash \mathsf{cases}(l,n,(;x)h_0,(;x)h_1):\tau}$$

$$\frac{\Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,x:N\vdash h_0:\tau \qquad \Gamma;\Delta,x:N\vdash h_1:\tau}{\Gamma;\Delta\vdash \mathsf{cases}(\mathsf{zero},n,(;x)h_0,(;x)h_1)=n:\tau}$$

$$\frac{\Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,x:N\vdash h_0:\tau \qquad \Gamma;\Delta,x:N\vdash h_1:\tau \qquad \Gamma;\Delta\vdash t:N}{\Gamma;\Delta\vdash \mathsf{cases}(\mathsf{s}_0(t),n,(;x)h_0,(;x)h_1)=h_0[t/x]:\tau}$$

$$\frac{\Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,x:N\vdash h_0:\tau \qquad \Gamma;\Delta,x:N\vdash h_1:\tau \qquad \Gamma;\Delta\vdash t:N}{\Gamma;\Delta\vdash \mathsf{cases}(\mathsf{s}_1(t),n,(;x)h_0,(;x)h_1)=h_1[t/x]:\tau}$$

$$\frac{\begin{array}{c}\Gamma;\Delta\vdash l:N \qquad \Gamma;\Delta\vdash n:\tau \qquad \Gamma;\Delta,x:N\vdash h_0:\tau \qquad \Gamma;\Delta,x:N\vdash h_0:\tau \\ \Gamma;\Delta,x:N\vdash g:\tau \qquad \Gamma;\Delta\vdash g[\mathsf{zero}/x]=n \\ \Gamma;\Delta,x:N\vdash g[\mathsf{s}_0(x)/x]=h_0:\tau \qquad \Gamma;\Delta,x:N\vdash g[\mathsf{s}_1(x)/x]=h_1:\tau\end{array}}{\Gamma;\Delta\vdash \mathsf{cases}(l,n,(;x)h_0,(;x)h_1)=g[l/x]:\tau}$$

We shall refer to $\mathbb{III}(\Sigma)$ extended with safe natural number type and the above rule as $\mathbb{IISB}(\Sigma)$. The interpretation of $\mathbb{IISB}(\Sigma)$requires SNNO with additional property:

Definition 5.3.2.
A SNNO I in an II-multicategory $\mathbf{M}$ satisfies **flat recursion property** if for any triple of multiarrows

$$g\in\mathbf{M}([\Gamma;\Delta],b) \qquad h_0\in\mathbf{M}([\Gamma;\Delta,x:N],b) \qquad h_1\in\mathbf{M}([\Gamma;\Delta,x:N],b)$$

there exists a unique multiarrow $c(g,h_0,h_1)\in\mathbf{M}([\Gamma;\Delta,x:N],b)]$ which satisfies the following conditions:

$$[\Gamma;\Delta]\ c(g,h_0,h_1)(\bar{x};\bar{y},z)=g$$
$$[\Gamma;\Delta,x:N]\ c(g,h_0,h_1)(\bar{x};\bar{y},s_i(;x))=h_i(\bar{x};\bar{y},x)$$

Given an II-multicategory **M** with a SNNO I which satisfies the flat recursion property we can interpret cases terms as follows:

$$[\![\mathsf{cases}(l,n,(;x)h_0,(;x)H_1)]\!] \stackrel{\text{def}}{=} c([\![n]\!],[\![h_0]\!],[\![h_1]\!])(\bar{x};\bar{y},[\![n]\!])$$

This interpretation is obviously sound with respect to the basic equalities for the cases operator.

Proposition 5.3.3. *In the multicategory* **B** *the SNNO* $\mathbb{N}$ *satisfies the flat recursion property.*

Proof. Given polytime computable functions

$$g : [x_1 : \mathbb{N},\ldots,x_n : \mathbb{N}; y_1 : \mathbb{N},\ldots,y_m : \mathbb{N}] \to \mathbb{N})$$
$$h_0 : [x_1 : \mathbb{N},\ldots,x_n : \mathbb{N}; y_1 : \mathbb{N},\ldots,y_m : \mathbb{N},x : \mathbb{N}] \to \mathbb{N})$$
$$h_1 : [x_1 : \mathbb{N},\ldots,x_n : \mathbb{N}; y_1 : \mathbb{N},\ldots,y_m : \mathbb{N},x : \mathbb{N}] \to \mathbb{N})$$

with bounding polynomials p_g, p_{h_0} and p_{h_1} we define $c(g,h_0,h_1)$ as an algorithm which executes g, h_0 or h_1 depending on the value of recursion argument. Such $c(g,h_0,h_1)$ is polytime computable function with a bounding polynomial $p_g + p_{h_0} + p_{h_1}$:

$$|c(g,h_0,h_1)(\bar{x};z,\bar{y})| = |g(\bar{x};\bar{y})| \leq p_g(|\bar{x}|) + \max_j |y_j| \leq p_{h_0}(|\bar{x}|) + p_{h_1}(|\bar{x}|) + p_g(|\bar{x}|) + \max\{\max_j |y_j|, 1\}$$

$$|c(g,h_0,h_1)(\bar{x};s_i(;n),\bar{y})| = |h_i(\bar{x};\bar{y},n)| \leq p_{h_i}(|\bar{x}|) + \max\{\max_j |y_j|, |n|\} \leq p_{h_0}(|\bar{x}|) + p_{h_1}(|\bar{x}|) + p_g(|\bar{x}|) + \max\{\max_j |y_j|, |n| + 2\}$$

□

In the $\mathbb{IISB}(\Sigma)$ calculus we can define the safe predecessor and conditional functions, which are included in the system $\mathfrak{B}$ as basic:

- The safe predecessor function $[;x : N]\ \mathsf{cases}(x,\mathsf{zero},(;x)x,(;x)x) : N$
- The conditional function $[;x : N, y : \tau, z : \tau]\ \mathsf{cases}(x,y,(;x)y,(;x)z) : \tau$

5.4 Representing system $\mathfrak{B}$

In this section we are going to show that the $\mathbb{III}\mathbb{SB}(\Sigma)$ calculus defined in the previous section contains all the components of the system $\mathfrak{B}$. Doing a syntactic translation from $\mathfrak{B}$ to $\mathbb{III}\mathbb{SB}(\Sigma)$ is a routine exercise since $\mathfrak{B}$ was the main motivation for $\mathbb{III}\mathbb{SB}(\Sigma)$.

Instead we shall concentrate on the semantic side and study what functions can be represented in an II-multicategory, which is equipped with SNNO, satisfying the flat recursion property (SNNOF for short). First we need to represent natural numbers as multiarrows in such a multicategory:

Definition 5.4.1.
In an II-multicategory with SNNOF for any natural number n we define the **numeral** $\hat{n}$ to be the following multiarrow:

$$\hat{n} = s_{n_0}(;(s_{n_1}(;\ldots(;s_{n_k}(;z)\ldots)$$

where $n_k, n_{k-1}, \ldots, n_0$ is a binary representation of n.

Now we define the notion of representable function in an II-multicategory with SNNOF:

Definition 5.4.2.
A function on natural numbers $f : \mathbb{N}^n \times \mathbb{N}^m \to \mathbb{N}$ is **representable** in an II-multicategory $\mathbf{M}$ with SNNOF I if there exists a multiarrow

$$[x_1 : I, \ldots, x_n : I; y_1 : I, \ldots, y_m : I] \xrightarrow{\hat{f}} I$$

such that for any sequence of natural numbers $(a_1, \ldots, a_n, a_{n+1}, \ldots, a_{n+m})$ we have

$$\hat{f} \circ_{[;]} \langle \hat{a}_1, \ldots, \hat{a}_n; \hat{a}_{n+1}, \ldots, \hat{a}_{n+m} \rangle = \hat{k}$$

where $k = f(a_1, \ldots, a_n, a_{n+1}, \ldots, a_{n+m})$.

Theorem 5.4.3. *Every function $f(x_1, \ldots, x_n; y_1, \ldots, y_m)$ definable in the system $\mathfrak{B}$ is representable in an II-multicategory $\mathbf{M}$ with a SNNOF I.*

Proof. We shall do the proof by induction on the definition of f in $\mathfrak{B}$. :

- If f is the zero function then take $\hat{f}$ to be $s_0(;z)$.
- If f is the projection function $\pi_j^{n,m}$ then take $\hat{f}$ to be the following multiarrow:

$$[x_1 : I, \ldots, x_n : I\,;\, x_{n+1} : I, \ldots, x_{n+m} : I]\, x_j$$

- If f is the successor function s_i then take $\hat{f}$ to be s_i.
- If f is the predecessor function p then it will be represented as the multiarrow $[;x:I]\,\mathsf{cases}(x,\mathsf{z},(;x)x,(;x)x)$.
- If f is the conditional function C then it will be represented as the multiarrow $[;x:I,y:I,z:I]\,\mathsf{cases}(x,y,(;x)y,(;x)z)$.
- If f is defined using the safe composition scheme

$$h(s_1(\bar{x};),\ldots,s_n(\bar{x};);t_1(\bar{x};\bar{y}),\ldots,t_m(\bar{x};\bar{y}))$$

then by induction we have $\hat{h}$, $\hat{s}_1,\ldots,\hat{s}_n$ and $\hat{t}_1,\ldots,\hat{t}_m$. We define $\hat{f}$ as the composition $\hat{h}(\hat{s}_1,\ldots,\hat{s}_n;\hat{t}_1,\ldots,\hat{t}_m)$.

- If f is defined using the safe recursion scheme:

$$\begin{aligned} f(0,\bar{x};\bar{y}) &= g(\bar{x};\bar{y}) \\ f(z.0,\bar{x};\bar{y}) &= h_0(z,\bar{x};\bar{y},f(z,\bar{x};\bar{y})) \\ f(z.1,\bar{x};\bar{y}) &= h_1(z,\bar{x};\bar{y},f(z,\bar{x};\bar{y})) \end{aligned}$$

then define $\hat{f}$ as $\mathsf{srec}(x,\hat{g},(x;y)\hat{h}_0,(x;y)\hat{h}_1)$

□

Theorem 5.4.4. *The following are equivalent:*

1. $f(x_1,\ldots,x_n)$ *is a polytime computable function on natural numbers.*
2. $f(x_1,\ldots,x_n;)$ *is representable in every II-multicategory with a SNNOF.*

Proof. We know that if $f(x_1,\ldots,x_n)$ is a polytime computable function on natural numbers then $f(x_1,\ldots,x_n;)$ belongs to $\mathfrak{B}$. Using the previous theorem we can conclude that $f(x_1,\ldots,x_n;)$ is representable in every II-multicategory with a SNNOF.

If $f(x_1,\ldots,x_n;)$ is representable in every II-multicategory with a SNNOF then it is representable in the multicategory **B**. □

We note that the use of II-multicategories (as opposed to ordinary categories with structure) is essential to our proof of this theorem because it relies on the multicategory **B**, which would not be easy to dress up as a category with structure since it is not clear how to define a tensor product $\otimes$ and linear exponential ! for encoding of safe/normal distinction.

The $\mathbb{IISB}(\Sigma)$ calculus defined in this chapter can be regarded as a typed version of system $\mathfrak{B}$. It contains safe composition and safe recursions schemes and all the basic functions of system $\mathfrak{B}$ can be represented in it. The interpretation of $\mathbb{IISB}(\Sigma)$ uses safe natural number object which satisfies flat recursion property, which can be regarded as a 'feasible' analogue of familiar categorical notion of natural number object.

Constructing such typed version of the system $\mathfrak{B}$ along with appropriate multicategorical interpretation sheds some light on the roles played by the different constituents of system $\mathfrak{B}$. We can summarize it as follows:

- The projection functions and the safe composition scheme together define the structure of II-multicategory, which is a dual-context analogue of the notion of category.
- The zero function, the binary successors and safe recursion constitute the definition of safe natural number object, which can be thought of as feasible analogue of natural number object.
- The predecessor and the conditional functions of system $\mathfrak{B}$ rather than being basic ingredients come from the special property of SNNO, which allows definitions by flat recursion.

Chapter 6

Strong endofunctors and initial algebras

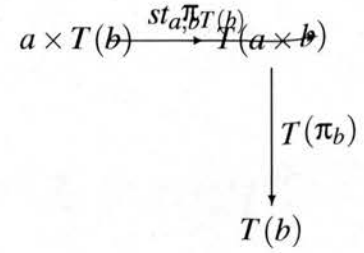
The single-context natural numbers type is an example of a so called inductive datatype. In categorical setting inductive datatypes are studied using the notion of parameterized initial algebra for a *strong* endofunctor [CS92, CS95, Jac95]. In this chapter we shall define a notion of strong endofunctors for II-multicategories and a notion of safe initial algebras for such endofunctors. We shall prove that some of the familiar properties of initial algebras hold for the safe initial algebras. We also show that in an II-multicategory with uniform coproducts and safe terminal objects the safe natural number object can be recast as a safe initial algebra for a suitable strong endofunctor.

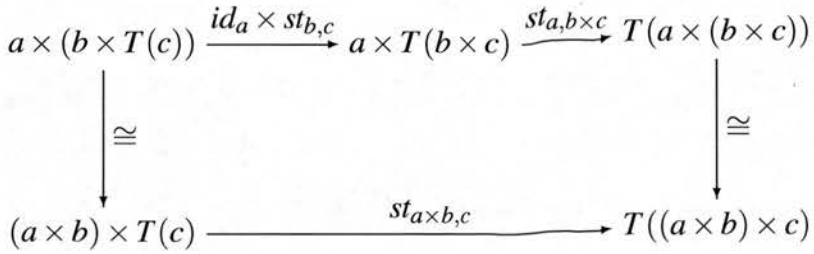
6.1 Strong endofunctors

We start by recalling the definition of strong endofunctor on a cartesian category [Jac01]. Then we work out a similar definition for a single-context cartesian multicategory and then extend it to a dual-context II-multicategory.

Definition 6.1.1.
An endofunctor $T : \mathbf{C} \to \mathbf{C}$ on a cartesian category $\mathbf{C}$ is called **strong** if it comes equipped with a natural transformation $st_{a,b} : a \times Tb \to T(a \times b)$ called a **strength**

which makes the following diagrams commute:





In a categorical setting such strong endofunctors are special kinds of endofunctors, which in turn are special kinds of functors. In a multicategorical setting such chain of specializations becomes problematic. The main reason for this is the asymmetry in multiarrows, which have sequences of objects as domains and objects as co-domains. The questions which have to be resolved is what is the action of an endofunctor on dual-contexts.

A first attempt to define endofunctors in a multicategorical setting would be to consider a map of objects $T : |\mathbf{M}| \to |\mathbf{M}|$ and assume that an application of an endofunctor T to a multiarrow $a_1 \ldots a_n \to b$ gives a multiarrow $T(a_1) \ldots T(a_n) \to T(b)$. Such a route was followed by Hermida in case of monoidal multicategories and we develop a similar definition for cartesian multicategories in Appendix B. Moreover we do not have to restrict to endofunctors since general functors can be defined in the same way.

The problem with this approach is that we can't obtain strong endofunctors in this way. One can systematically relate cartesian multicategories and cartesian categories and show that such functors on multicategories correspond to product-preserving functors on cartesian categories. Since strong endofunctors do not preserve products in general such a definition would be too narrow for our purposes.

If we look at the above definition of a strong endofunctor we see that strength transformation allows us to construct a morphism $a \times T(b) \to T(c)$ out of a morphism $T(a \times b) \to T(c)$. This gives an idea of a different approach to the definition of a strong endofunctor in a multicategorical settings. Consider a map of objects $T : |\mathbf{M}| \to |\mathbf{M}|$. An application of T to a multiarrow $a_1 \ldots a_n \to b$ will happen at a *position* i giving a multiarrow $a_1 \ldots a_{i-1}.T(a_i).a_{i+1} \ldots a_n \to T(b)$. Obviously such an approach works

only for endofunctors and is not obvious how to define general functors in the same way.

We propose the following definition of a strong endofunctors based on the above idea. In Appendix C we give a more precise correspondence between single-context strong endofunctors as defined below and strong endofunctors on a cartesian category. We shall be working with cartesian multicategories, which are defined in Appendix A.

Definition 6.1.2.
A **strong endofunctor** T on a cartesian multicategory $\mathbf{C}$ is given by the following family of maps:

- A map of objects $T : |\mathbf{C}| \to |\mathbf{C}|$
- For each object b, each sequence of objects $a_1 \ldots a_n$ and each object a_i within this sequence a map of multiarrows

$$T_{a_i} : \mathbf{C}(a_1 \ldots a_n, b) \to \mathbf{C}(a_1 \ldots a_{i-1}\ T(a_i)\ a_{i+1} \ldots a_n, T(b))$$

These maps should satisfy the following properties:

- **Identity law** $T_{a_i}(\pi_{a_i}) = \pi_{T(a_i)}$
- **Composition law**

$$T_{a_j^i}(f(g_1, \ldots, g_n)) = T_{b_i}(f)(g_1, \ldots, g_{i-1}, T_{a_j^i}(g_i), g_{i+1}, \ldots, g_n)$$

  where $f : \bar{b} \to c$ and $g_i : \bar{a}_i \to a_i$.

Extending this definition to Π-multicategories is quite straightforward. We again need a map of objects $F : |\mathbf{M}| \to |\mathbf{M}|$. Then for each object b, for each dual-context $[\Gamma;\Delta]$ and for each variable x within this context (either normal or safe) we need a map of multiarrows

$$\mathbf{M}([\Gamma;\Delta], b) \to \mathbf{M}([\Gamma';\Delta'], F(b))$$

where the context $[\Gamma';\Delta']$ is the same as $[\Gamma;\Delta]$ except that the variable x has type $F(a)$. We shall use $F_x(f)$ to denote application of F to a multiarrow f at a position x.

Remark 6.1.3. Another natural definition of a dual-context strong endofunctor would be one in which the positions the functor can be applied to are restricted to either normal or safe. We do not consider this further because such restrictions are not suitable for our purposes.

Which properties should these maps satisfy? Firstly, we require that they preserve normal and safe projections:

$$F_{x_i}(\pi_{x_i}) = \pi_{x_i} \qquad \text{where } \pi_{x_i} : [\Gamma;] \to a_i \tag{ID-S}$$

$$F_{y_j}(\sigma_{y_j}) = \sigma_{y_j} \qquad \text{where } \sigma_{y_j} : [\Gamma;\Delta] \to b_j \tag{ID-N}$$

Secondly, our strong endofunctors should preserve linear composition. Given a multiarrow $f : [\Gamma;\Delta] \to b$ and a family of multiarrows, which do not share any variables:

$$g_i : [\Gamma_i;] \to a_i \qquad h_j : [\Pi_j;\Theta_j] \to b_j \qquad \Gamma_i, \Pi_j, \Theta_j \text{ mutually disjoint}$$

we can compose them together and get a multiarrow

$$[\Pi;\Theta]\, f(g_1(\bar{x}_1;),\ldots,g_n(\bar{x}_n;);h_1(\bar{u}_1;\bar{v}_1),\ldots,h_m(\bar{u}_m;\bar{v}_m))$$
$$[\Pi;\Theta] = [\Gamma_1,\ldots,\Gamma_n,\Pi_1,\ldots,\Pi_m;\Theta_1,\ldots,\Theta_m]$$

Suppose F is applied to such composition at a position x which belongs to the context Γ_i. Since g_i is the only multiarrow which really depends on the input x, such application should commute with the composition in the following way:

$$T_x(f(g_1(\bar{x}_1;),\ldots,g_n(\bar{x}_n;);h_1(\bar{u}_1;\bar{v}_1),\ldots,h_m(\bar{u}_m;\bar{v}_m))) =$$
$$T_{x_i}(f)(g_1(\bar{x}_1;),\ldots,T_x(g_i)(\bar{x}_i;),\ldots,g_n(\bar{x}_n;);h_1(\bar{u}_1;\bar{v}_1),\ldots,h_m(\bar{u}_m;\bar{v}_m)) \tag{COMP-N}$$

Similarly, if the position x comes from the context $[\Pi_i;\Theta_j]$ the following should hold:

$$T_x(f(g_1(\bar{x}_1;),\ldots,g_n(\bar{x}_n;);h_1(\bar{u}_1;\bar{v}_1),\ldots,h_m(\bar{u}_m;\bar{v}_m))) =$$
$$T_{x_i}(f)(g_1(\bar{x}_1;),\ldots,g_n(\bar{x}_n;);h_1(\bar{u}_1;\bar{v}_1),\ldots,T_x(h_i)(\bar{u}_i;\bar{v}_i),\ldots,h_m(\bar{u}_m;\bar{v}_m)) \tag{COMP-S}$$

Putting it all together, we propose the following definition:

Definition 6.1.4 (Strong endofunctor)**.**
A **strong endofunctor** $F : \mathbf{M} \to \mathbf{M}$ on an Π-multicategory $\mathbf{M}$ is given by a family of maps, consisting of:

- A map of objects $F : |\mathbf{M}| \to |\mathbf{M}|$
- For each object b, for each dual-context $[\Gamma;\Delta]$ and for each variable $x : a$ within this context (either normal or safe) a map of multiarrows

$$\mathbf{M}([\Gamma;\Delta],b) \to \mathbf{M}([\Gamma';\Delta'],F(b))$$

  where context $[\Gamma';\Delta']$ is the same as $[\Gamma;\Delta]$ except variable x has type $F(a)$.

These maps should preserve normal and safe projections (ID-N and ID-S above) and linear composition (COMP-N and COMP-S above).

As in the case of single-context multicategories this definition only makes sense for endofunctors and there is no obvious way of generalizing it to functors between different multicategories.

There is an obvious identity endofunctor and strong endofunctors can be composed.

Definition 6.1.5.
Let $T : \mathbf{M} \to \mathbf{M}$ and $S : \mathbf{M} \to \mathbf{M}$ be strong endofunctors on $\mathbf{M}$. The strong endofunctor $S \cdot T : \mathbf{M} \to \mathbf{M}$ is defined as follows:

- $S \cdot T(a) = S(T(a))$ for each $a \in |\mathbf{M}|$.
- For each multiarrow $f \in \mathbf{M}([\Gamma;\Delta], a)$ $(S \cdot T)_x(f) = S_x(T_x(f))$

Lemma 6.1.6. *$S \cdot T$ is a strong endofunctor.*

Proof. A routine check that $S \cdot T$ satisfies properties of a strong endofunctor. □

We now give some examples of strong endofunctors:

Example 6.1.7.
Suppose $\mathbf{M}$ is an Π-multicatgory with a safe product $\times$. Let $\mathbf{c} \in |\mathbf{M}|$. We define strong endofunctor $P^{\mathbf{c}}$ as follows:

- $P^{\mathbf{c}}(a) = \mathbf{c} \times a$
- Given a multiarrow $f : [\Gamma, z : a; \Delta] \to b$ we define a multiarrow $P^{\mathbf{c}}_z(f)$ as follows:

$$[\Gamma, z : c \times a; \Delta]\ P^{\mathbf{c}}_z(f)(\bar{x}, z; \bar{y}) = [pr_1(;z), f(\bar{x}, pr_2(;z); \bar{y})]$$

- Given a multiarrow $f : [\Gamma; \Delta, z : a] \to b$ we define a multiarrow $P^{\mathbf{c}}_z(f)$ as follows:

$$[\Gamma, ; \Delta, z : c \times a]\ P^{\mathbf{c}}_z(f)(\bar{x}; \bar{y}, z) = [pr_1(;z), f(\bar{x}, pr_2(;z); \bar{y})]$$

Lemma 6.1.8. *$P^{\mathbf{c}}$ is a strong endofunctor.*

Proof. We verify the properties of a strong endofunctor:

- For the safe identity preservation we have

$$[\Gamma; \Delta, z : c \times a]\ P^{\mathbf{c}}_z(\sigma_z) = [pr_1(;z), \sigma_z(\bar{x}; \bar{y}, pr_2(;z))] = [pr_1(;z), pr_2(;z)] = z$$

- For the normal identity preservation property we have:

$$[\Gamma, z : c \times a;]\, P^c_z(\pi_z) = [pr_1(;z), \pi_z(\bar{x}, pr_2(;z);)] = [pr_1(;z), pr_2(;z)] = z$$

- For the COMP-S property the left-hand side is equal to

$$[pr_1(;z), f(\bar{g};\bar{h}, s(\bar{u};\bar{v}, pr_2(;z)))]$$

  Using the definition of P^c we have $P^c_x(f) = [pr_1(;x), f(\bar{x};\bar{y}, pr_2(;x)]$. If we substitute the family $\bar{g};\bar{h}, [pr_1(;z), s(\bar{u};\bar{v}, pr_2(;z))]$ we can write the right-hand side of COMP-S as $[pr_1(;z), f(\bar{g};\bar{h}, s(\bar{u};\bar{v}, pr_2(;z)))]$, which is equal to the left-hand side of COMP-S. Similarly we can check that COMP-N holds.

□

Example 6.1.9.
Suppose **M** is an II-multicategory with a uniform coproduct. Let $\mathbf{c} \in |\mathbf{M}|$. The endofunctor S^c will be defined as follows:

- $S^c(a) = c + a$

- Given a multiarrow $f : [\Gamma, z : a; \Delta] \to b$ we define

$$[\Gamma, z : c + a; \Delta]\, S^c_z(f) = \mathsf{case}(z, (x : c;) inl_{c,b}(;x), (z : a;) inr_{c,b}(; f(\bar{x}, z; \bar{y})))$$

Example 6.1.10.
Suppose **M** is an II-multicategory with a safe coproduct. The endofunctor d is defined as follows:

- $d(a) = a + a$

- For any multiarrow $f : [\Gamma, z : a; \Delta] \to b$ we define

$$[\Gamma, z : a + a; \Delta]\, d_z(f) = inl_{b,b}(; \{f|f\}(\bar{x}, z; \bar{y}))$$

Checking that d and S^c are strong endofunctors is similar to the proof we gave for the endofunctor P^c and is quite technical, so we decided to omit it.

6.2 Safe initial algebras

In the previous section we have defined the notion of strong endofunctor on an II-multicategory. In this section we shall consider a notion of a *safe initial algebra* for such endofunctors. Such initial algebras can be used for the interpretation of the safe natural number type we defined i the previous chapter.

We start with the following categorical definition of parameterized initial algebra [CS92, CS95, Jac01]:

Definition 6.2.1.
Let $\mathbf{C}$ be a cartesian category and $T : \mathbf{C} \to \mathbf{C}$ be a strong endofunctor on $\mathbf{C}$. A **parameterized initial algebra** for T is a morphism $T(I) \xrightarrow{\alpha} I$ such that for any morphism $A \times T(B) \xrightarrow{h} B$ there exists a unique morphism $A \times I \xrightarrow{f} B$ making the following diagram commute:

$$\begin{array}{ccccc} A \times T(I) & \xrightarrow{\langle \pi_1, st_{A,I} \rangle} & A \times T(A \times I) & \xrightarrow{id \times T(f)} & A \times T(B) \\ \downarrow{\scriptstyle id \times \alpha} & & & & \downarrow{\scriptstyle h} \\ A \times I & & \xrightarrow[f]{} & & B \end{array}$$

Let us specialize this definition to one particular case and see how initial algebras can represent functions defined by recursion [Rom89]. Let $\mathbf{C}$ be a cartesian category with distributive coproducts. Consider a strong endofunctor N with action on objects as $N(x) = \mathbf{1} + x$ and on morphisms as $N(f) = \{inl(id_1), inr(f)\}$. Then a parameterized initial algebra $\alpha : \mathbf{1} + N \to N$ for this endofunctor is nothing more but a parameterized natural number object given by morphisms $zero : \mathbf{1} \to N$ and $s : N \to N$ [LS86].

A function on natural numbers $f : \mathbb{N}^m \to \mathbb{N}$ is **representable** in such $\mathbf{C}$ with a parameterized natural number object N if there exists a morphism $\hat{f} : N^m \to N$ such that for any sequence of natural numbers $a_1, \ldots, a_n$ we have $\hat{f} \circ \langle \hat{a}_1, \ldots, \hat{a}_n \rangle = \hat{k}$ where $k = f(a_1, \ldots, a_n)$ and $\hat{n}$ is defined as n applications of s to $zero$.

Then using properties of parameterized natural number object is possible to represent functions, which are defined by the following recursion scheme:

$$f(0, y) = g(y)$$
$$f(s(x), y) = h(y, f(x, y))$$

This scheme differs from the usual primitive recursion scheme, which has an additional

parameter in the step function:

$$f'(0,y) = g'(y)$$
$$f'(s(x),y) = h'(x,y,f'(x,y))$$

But we can code such recursion scheme using products as $f'(x,y) = \pi_1(f(x,y))$ where f is defined by recursion as follows:

$$f(0,y) = pair(g'(y), zero)$$
$$f(s(x),y) = pair(h'(\pi_2(f(x,y)),y,\pi_1(f(x,y))), s(\pi_2(f(x,y))))$$

The goal of this section is to define such notion of dual-context initial algebra that when applied to a particular dual-context strong endofunctor can represent the safe recursion scheme:

$$f(0,\bar{x};\bar{y}) = g(\bar{x};\bar{y})$$
$$f(z.0,\bar{x};\bar{y}) = h_0(z,\bar{x};\bar{y},f(z,\bar{x};\bar{y}))$$
$$f(z.1,\bar{x};\bar{y}) = h_1(z,\bar{x};\bar{y},f(z,\bar{x};\bar{y}))$$

Such initial algebra should be a multiarrow of the form $[;x : T(I)] \xrightarrow{\alpha} I$. Why do we insist that x be safe? Recall that in the system $\mathfrak{B}$ the successors s_0 and s_1 each have one safe argument and asking for a function with a safe argument is in general a stronger requirement since turning a safe argument into normal is allowed by the safe composition.

Next we should decide on the signature for the step function h. Obviously h should have both normal and safe parameters given by the context $[\Gamma;\Delta]$. In addition to those parameters h should have a position for plugging in a recursive value. The safe recursion scheme dictates that this position should be safe. Another parameter of the step function should be a normal variable for plugging in the recursion argument. So the step function should be $[\Gamma,x : T(\mathbf{I});\Delta,y : T(a)] \xrightarrow{h} a$.

For each such step multiarrow we require the existence of a multiarrow $[\Gamma,z : \mathbf{I};\Delta] \xrightarrow{f} a$ where z is the recursion argument, which according to the safe recursion scheme should be normal. This f should satisfy the following property:

$$[\Gamma,z : T(\mathbf{I});\Delta]\ f(\bar{x},\alpha(;z);\bar{y}) = h(\bar{x},z;\bar{y},T_z(f)(\bar{x},z;\bar{y}))$$

Note that the functor T is applied to the normal z. Summing up, we propose the following definition of initial algebra:

Definition 6.2.2 (Safe initial algebra).
Given an **II**-multicategory **M** and a strong endofunctor $T : \mathbf{M} \to \mathbf{M}$ a **safe initial algebra** for T is a multiarrow $[;x : T\mathbf{I}] \xrightarrow{\alpha} \mathbf{I}$ such that for any object a and any multiarrow $[\Gamma,x : T(\mathbf{I});\Delta,y : T(a)] \xrightarrow{h} a$ there exists a unique multiarrow $[\Gamma,z : \mathbf{I};\Delta] \xrightarrow{f} a$ such that

$$[\Gamma,z : T(\mathbf{I});\Delta]\ f(\bar{x},\alpha(;z);\bar{y}) = h(\bar{x},z;\bar{y},T_z(f)(\bar{x},z;\bar{y}))$$

Obtaining the safe recursion scheme out of this definition requires the strong endofunctor $B(x) = \mathbf{1}+x+x$. Let **M** be an II-multicategory with a uniform coproduct and safe terminal object **1**. Then, as we have seen in the previous section, B is indeed a strong endofunctor. Assume that $\alpha = \{\varepsilon|s_0|s_1\}$ is a safe initial algebra for B, where $[;] \xrightarrow{\varepsilon} I$, $[;x : I] \xrightarrow{s_0} I$ and $[;x : I] \xrightarrow{s_1} I$.

For any multiarrow $[\Gamma,x : \mathbf{1}+I+I;\Delta,y : \mathbf{1}+a+a] \xrightarrow{h} a$ there exists a unique multiarrow $[\Gamma,z : I;\Delta] \xrightarrow{f} a$ such that

$$[\Gamma,z : 1+I+I;\Delta]\ f(\bar{x},\alpha(;z);\bar{y}) = h(\bar{x},z;\bar{y},B_z(f)(\bar{x},z;\bar{y}))$$

Substituting $[inl_{1,I+I}(;x)/z]$ we get

$$[\Gamma;\Delta]\ f(\bar{x},\varepsilon;\bar{y}) = g(\bar{x};\bar{y})$$

Similarly for the substitutions $[inr_{1,I+I}(;inl_{I,I}(;x)/z]$ and $[inr_{1,I+I}(;inr_{I,I}(;x)/z]$ we have

$$[\Gamma,x : I;\Delta]\ f(\bar{x},s_0(;x);\bar{y}) = h_0(\bar{x},x;\bar{y},f(\bar{x},x;\bar{y}))$$
$$[\Gamma,x : I;\Delta]\ f(\bar{x},s_1(;x);\bar{y}) = h_1(\bar{x},x;\bar{y},f(\bar{x},x;\bar{y}))$$

We shall now prove some properties of safe initial algebras which are dual-context version of corresponding properties of categorical initial algebras. We start with checking that safe initial algebras are unique up to isomorphism:

Proposition 6.2.3. *Let **M** be an II-multicategory and $T : \mathbf{M} \to \mathbf{M}$ be a strong endofunctor on* **M**. *Assume that $[;x : T(I)] \xrightarrow{\alpha} I$ and $[;x : T(I')] \xrightarrow{\alpha'} I'$ are safe initial algebras for T. Then there exists $[x : I;] \xrightarrow{i} I'$ and $[x : I';] \xrightarrow{i'} I$ such that $[x : I;]\ i'(i(x;);) = x$ and $[x : I';]\ i(i'(x;);) = x$*

Proof. Consider the multiarrow $[x : T(I);y : T(I')]\ \alpha'(;y)$. By the initial algebra property of α there exists $[x : I;] \xrightarrow{i} I'$ such that $[z : T(I);]\ i(\alpha(;z);) = \alpha'(;T_x(i)(z;))$. Similarly for the multiarrow $[x : T(I');y : T(I)]\ \alpha(;y)$ there exists a multiarrow $[x : I';] \xrightarrow{i'} I$ such that

$$[z : T(I');]\ i'(\alpha'(;z);) = \alpha(;T_x(i')(z;))$$

Consider the multiarrow $[x:I;]\ i'(i(x;);)$ such that

$$[z:T(I);]\ i'(i(\alpha(;z););) = i'(\alpha'(;T_x(i)(z;));) = \alpha(;T_x(i')(T_x(i)(z;);))$$

By the composition property of strong functors this is equal to

$$[z:T(I);]\ \alpha(;T_x(i'(i(x;);))(z;))$$

Now consider the multiarrow $[x:T(I);y:T(I)]\ y$. By the initial algebra property there exists a unique $[x:I;] \xrightarrow{f} I$ such that $[z:T(I);]\ f(\alpha(;z);) = \alpha(;T_x(f)(z;))$. On one hand f must be equal to $[x:I;]\pi_x I$ since $[z:T(I);]\ pi_x(\alpha(;z);) = \alpha(;z) = \alpha(;T_x(\pi_x)(z;))$ by the projection property of safe endofunctors.

But the multiarrow $[x:I;]\ i'(i(x;))$ satisfies the same equation. So by uniqueness we have that $[x:I;]\ i'(i(x;);) = x$. Similarly we obtain that $[x:I';]\ i(i'(x;);) = x$. □

Note that since we are using the initiality property of α to construct i and i' these multiarrows are normal. In general we can't show the stronger property of having a safe isomorphism. This observation also applies to the next property which states that any safe initial algebra has a normal inverse and can be seen as a dual-context analogue of Lambek's Lemma for initial algebras in category theory:

Proposition 6.2.4. *Let $\mathbf{M}$ be an II-multicategory and $T : \mathbf{M} \to \mathbf{M}$ be a strong endofunctor on $\mathbf{M}$. Suppose that $[;x:T(I)] \to I$ is a safe initial algebra for T. Then there exists a multiarrow $: [x:I;] \xrightarrow{\alpha^{-1}} T(I)$ such that*

$$[x:\mathbf{I};]\ \alpha(;\alpha^{-1}(x;)) = x$$
$$[x:T(\mathbf{I});]\ \alpha^{-1}(\alpha(;x);) = x$$

Proof. We shall construct $\alpha^{-1} : [x:\mathbf{I};] \to T(\mathbf{I})$ as follows:

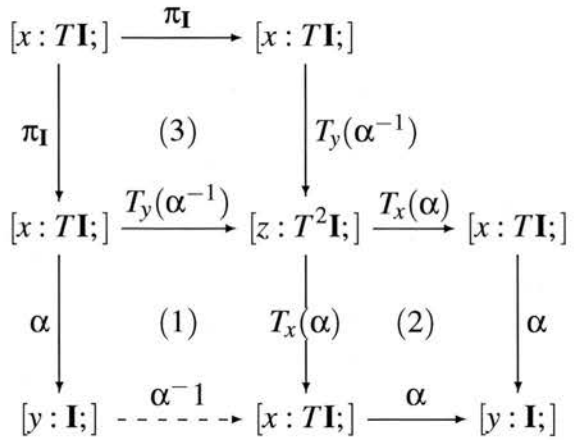


Diagram (1) defines α^{-1} using the initiality property of α for the step function $T_x(\alpha)$. Diagram (2) commutes which makes the combined horizontal diagram (1-2) commute. Since T preserves composition

$$T_x(\alpha) \circ \langle T_y(\alpha^{-1});\rangle = T_y(\alpha \circ \langle \alpha^{-1};\rangle) \qquad \text{T-Comp}$$

so the outer diagram is the initiality property of α for the step function α. By uniqueness we have

$$\alpha \circ \langle \alpha^{-1};\rangle = \pi_{\mathbf{I}} \qquad \text{L1}$$

Diagram (3) commutes so combined vertical diagram (3-1) commutes. Together with T-Comp and L1 this gives

$$\alpha^{-1} \circ \langle \alpha;\rangle = \pi_{T\mathbf{I}}$$

so α is an isomorphism. □

Using the normal inverse α^{-1}, constructed in the previous lemma, we can show that the application of the strong endofunctor T to a safe initial algebra for T gives back another safe initial algebra:

Proposition 6.2.5. *Let $\mathbf{M}$ be an II-multicategory and $T : \mathbf{M} \to \mathbf{M}$ be a strong endofunctor on $\mathbf{M}$. Assume that $[;x : T(I)] \xrightarrow{\alpha} I$ is a safe initial algebras for T. Then a multiarrow $T_x(\alpha) : [;x : T(T(I))] \to T(I)$ is also a safe initial algebra for T.*

Proof. We need to check that for any multiarrow $h : [\Gamma, x : T(T(I)); \Delta, y : T(a)]$ there exists a unique multiarrow $f : [\Gamma, z : T(I); \Delta] \to a$ such that the following holds:

$$[\Gamma, u : T(T(I)); \Delta] \quad f(\bar{x}, T_x(\alpha)(;u); \bar{y}) = h(\bar{x}, u; \bar{y}, T_z(f)(\bar{x}, z; \bar{y}))$$

By Lemma 6.2.4 we have a multiarrow $[x : I;] \xrightarrow{\alpha^{-1}} T(I)$ which is a normal inverse of α. Consider a multiarrow defined by the following composition:

$$[\Gamma, z : T(I); \Delta, y : T(a)] \quad h'(\bar{x}, z; \bar{y}, y) = h(\bar{x}, T_x(\alpha^{-1})(z;); \bar{y}, y))$$

Since α is a safe initial algebra for such h there exists a unique $f' : [\Gamma, z : I; \Delta] \to a$ such that the following holds:

$$[\Gamma, z : T(I); \Delta] \quad f'(\bar{x}, \alpha(;z); \bar{y}) = h'(\bar{x}, z; \bar{y}, T_z(f')(\bar{x}, z; \bar{y}))$$

We can now define f using the following composition:

$$[\Gamma, z : T(I); \Delta] \quad f(\bar{x}, z; \bar{y}) = f'(\bar{x}, \alpha(;z); \bar{y})$$

Then we have the following:

$$[\Gamma, u : T(T(I)); \Delta] \quad f(\bar{x}, T_x(\alpha)(;u); \bar{y}) = f'(\bar{x}, \alpha(;T_x(\alpha)(;u)); \bar{y})$$
$$= h'(\bar{x}, T_x(\alpha)(;u); \bar{y}, T_z(f')(\bar{x}, T_x(\alpha)(;u); \bar{y}))$$

□

In this section we have defined a notion of safe initial algebra for a strong endofunctor which is an analogue of the categorical notion of initial algebra and have shown that some of the usual properties of initial algebras hold for safe initial algebras as well.

6.3 Flat recursion property

In the previous section we have proved that any safe initial algebra $[;x : T(I)] \xrightarrow{\alpha} I$ has an inverse $[x : I;] \xrightarrow{\alpha^{-1}} T(I)$. Since α^{-1} was constructed using initiality property it is essential that the argument x is normal. It seems that having a safe inverse does not follow from the properties of initial algebra and it is therefore interesting to add the existence of such a safe inverse as an additional property to an initial algebra.

Definition 6.3.1.
Let $\mathbf{M}$ be an II-multicategory with a strong endofunctor $T : \mathbf{M} \to \mathbf{M}$. A multiarrow $[;x : T(I)] \xrightarrow{\alpha} I$ is called a **strongly safe initial algebra** for T if it is a safe initial algebra for T and there exists a multiarrow $[;x : \mathbf{I}] \xrightarrow{\alpha^{-1}} T\mathbf{I}$ such that

$$[;x : \mathbf{I}] \vdash \alpha(;\alpha^{-1}(;x)) = x$$
$$[;x : T\mathbf{I}] \vdash \alpha^{-1}(;\alpha(;x)) = x$$

The property of being a strongly safe initial algebra has an illuminating alternative characterization:

Definition 6.3.2.
Let $\mathbf{M}$ be an II-multicategory with a strong endofunctor $T : \mathbf{M} \to \mathbf{M}$ and a multiarrow α be a safe initial algebra for T. We say that α satisfies the **flat recursion property** if for any object a and any multiarrow $[\Gamma, x : T(I); \Delta] \xrightarrow{h} a$ there exists a unique multiarrow $[\Gamma; x : \mathbf{I}, \Delta] \xrightarrow{f} a$ such that the following holds:

$$[\Gamma, z : T(I); \Delta]\ f(\bar{x}; \alpha(;z), \bar{y}) = h(\bar{x}, z; \bar{y})$$

Proposition 6.3.3. *A safe initial algebra α is strongly safe iff α satisfies the flat recursion property.*

Proof. Assume that α has a safe inverse. Given a multiarrow $[\Gamma, x : T(I); \Delta] \xrightarrow{h} a$ consider the following multiarrow:

$$[\Gamma, z : I; \Delta]\ f(\bar{x}; z, \bar{y}) = h(\bar{x}, \alpha^{-1}(; z); \bar{y})$$

For this multiarrow the following holds:

$$[\Gamma, z : T(I); \Delta]\ f(\bar{x}; \alpha(; z), \bar{y}) = h(\bar{x}, \alpha^{-1}(; \alpha(; z)); \bar{y}) = h(\bar{x}, z; \bar{y})$$

So α satisfies the flat recursion property.

In order to show the opposite direction consider a multiarrow $[x : T(I);]\ x$. Since α satisfies the flat recursion property there exists a multiarrow $[; x : I] \xrightarrow{\alpha^{-1}} T(I)$ such that

$$[x : T(I);]\ \alpha^{-1}(; \alpha(; x)) = x$$

Consider a multiarrow $[x : T(I);]\ \alpha(; x)$. Since α satisfies the flat recursion property there exists a multiarrow $[; x : I] \xrightarrow{f} T(I)$ such that $[x : T\mathbf{I};]\ f(; \alpha(; x)) = \alpha(; x)$. On one hand we should have that $[; x : I]\ f(; x) = x$. On the other hand the following holds

$$[; x : I]\ f(; x) = \alpha(; \alpha^{-1}(; x))$$

since $[x : T(I);]\ \alpha(; \alpha^{-1}(; \alpha(; x))) = \alpha(; x)$. So by uniqueness of f we have that

$$[; x : I]\ \alpha(; \alpha^{-1}(; x)) = x$$

□

We shall now prove that our definition of a strongly safe initial algebra is equivalent to the definition of safe natural number object, satisfying the flat recursion property:

Theorem 6.3.4. *Let* $\mathbf{M}$ *be an II-multicategory with a uniform coproduct and a safe terminal object. Consider a strong endofunctor* $B(x) = \mathbf{1} + x + x$ *on* $\mathbf{M}$. *Then the following holds:*

- *There exists a safe initial algebra for* B *in* $\mathbf{M}$ *iff there exists a safe binary list object.*
- *There exists a strongly safe initial algebra for* B *in* $\mathbf{M}$ *iff there exists a safe binary list object, satisfying flat recursion property.*

Proof. We shall give only the sketch of the proof. Let I be an SNNOF. Then we consider a multiarrow $[;x:1+I+I]\ \alpha(;x) = \{z'|s_0|s_1\}$, where $z' : [;x:1] \to I$ is obtained from z. Any multiarrows $[\Gamma,x:1+I+I;\Delta,y:1+b+b] \to b$ is uniquely determined by the multiarrows $g : [\Gamma\,;\Delta] \to b$, $h_0 : [\Gamma,x:I\,;\Delta,y:b] \to b$ and $h_1 : [\Gamma,x:I\,;\Delta,y:b] \to b$. For these multiarrows using the universal property of SNNO we obtain the multiarrow $f : [\Gamma,x:I;\Delta] \to a$. Checking that this multiarrow satisfies the required universal property of safe initial algebra is routine. In the same way we can check that this construction satisfies the flat recursion property and in the opposite direction construct the SNNO I out of the safe initial algebra $[;x:1+I+I] \xrightarrow{\alpha} I$. □

In this chapter we defined strong endofunctors for II-multicategories and safe initial algebras for such endofunctors. We have shown how safe initial algebras can be used for interpretation of safe inductive datatypes in particular safe natural number objects introduced in the previous chapter. We have proved that some of the usual properties of initial algebras hold for the safe initial algebras. In particular we have proved the dual-context analogue of Lambek's lemma showing that safe initial algebras have normal inverse. Obtaining the safe inverse seemed not trivial so we added sych property to the definition of safe initial algebras and showed that the resulting notion can be used to give a sound interpretation of extended safe natural number type, introduced in the previous chapter.

Chapter 7

Conclusions and future work

7.1 Main results

In this thesis we worked with calculi in which normal and safe variables were considered. The separation of variables into two kinds led us to using dual-contexts for typing the free variables occurring in terms. In this setting we were able to define different rules for using normal and safe variables. In particular, we considered different structural rules for normal and safe variables. In order to maintain the separation between normal and safe variables the substitution had to be restricted.

We showed how different dual-context calculi with restricted substitution could be described in a uniform way using the minimal dual-context calculus and specifying structural rules as renaming operations. We paid special attention to the dual-context calculus $\mathbb{III}(\Sigma)$.

We observed that constructing a complete interpretation of a basic dual-context calculus with safe substitution using standard techniques was problematic because of absence of necessary type constructors in the calculus. We proposed an alternative approach to the interpretation of such calculus based on the notion of dual-context multicategory. We proved soundness and completeness for this interpretation. We then defined a uniform notion of R-multicategory, which covered different flavors of dual-context multicategories in a uniform way.

One important dual-context multicategory for interpretation of $\mathbb{III}(\Sigma)$ was the multicategory $\mathbf{B}$ built out of polytime computable functions on natural numbers with size bounded in a special way. This multicategory was used in establishing our main representability results.

We speculated about the relation between multicategorical and categorical models

of $\mathbb{II}(\Sigma)$. In order to establish it precisely we needed to extend $\mathbb{II}(\Sigma)$ with various type constructors. We considered such an extension for product types. We observed that several different flavors of product types exist and gave sound multicategorical interpretation for them. We also considered extensions of $\mathbb{II}(\Sigma)$ with sums and unit type.

The most important extension of $\mathbb{II}(\Sigma)$ was done when we considered safe dual-context natural numbers. The typing rules for this extension were formulated in such a way that the safe recursion scheme of Bellantoni and Cook's system $\mathfrak{B}$ could be defined. The safe dual-context natural numbers were interpreted using the notion of safe natural number object. We also defined safe natural number object with an additional property, which was capable of expressing the flat recursion scheme. In this way we obtained the calculus $\mathbb{IISB}(\Sigma)$ in which whole system $\mathfrak{B}$ could be defined.

The $\mathbb{IISB}(\Sigma)$ calculus allowed us to better understand the relationship between the different parts of system $\mathfrak{B}$. The fundamental part of system $\mathfrak{B}$ consists of the projection functions and the safe composition scheme, which are the basic ingredients needed in the setting with two kinds of variables. Adding successors and safe recursion into the system can be regarded as adding a safe dual-context natural number type. Finally, the addition of the safe predecessor and conditional can be seen as a consequence of adding the flat recursion.

We showed that every function from $\mathfrak{B}$ could be represented in an II-multicategory with a safe natural number object, which satisfied flat recursion property (SNNOF for short). Using the multicategory $\mathbf{B}$ we also showed that only the polytime functions could be represented in every II-multicategory with SNNOF.

We recast SNNO as safe initial algebra for a particular strong endofunctor on an II-multicategory. This required a definition of strong endofunctors and safe initial algebra. We showed that adding a generalized flat recursion property to a safe initial algebra was equivalent to requiring that it has a safe inverse.

7.2 Future work

One of the interesting possibilities for continuing the research presented in this thesis is studying extensions of basic dual-context calculi with other common type constructors. In this thesis we considered variations of products and sums in the II-case. Similar extensions can be considered without difficulty in the IL-case and another interesting possibility is to consider these extensions in an arbitrary basic dual-context calculus

given by the minimal dual-context calculus extended with a collection of renamings.

One important extension to consider would be the exponential types of linear logic, e.g. as defined in type system DILL. Having a multicategorical semantics of exponential types together with monoidal product types is essential in order to establish the relationship between dual-context multicategories and categories with additional structure (like linear/non-linear models of DILL) used for interpretation of dual-context calculi.

Function types is another extension, which is very interesting from the feasible computability point of view. In order to explain this in more details we need to consider briefly some of the development, which have been proposed to extend Bellantoni and Cook's system $\mathfrak{B}$.

Hofmann [Hof97] constructed a type system SLR based on the ideas from Bellantoni and Cook's approach. In SLR dual-contexts are not present explicitly, instead a unary type constructor $\Box$ is used for representing normal values. Among other things, SLR includes linear function types and recursor constant, which enable definitions by safe recursion. Hofmann showed that in the presence of function types one needs to impose a linearity discipline on using safe values otherwise it would be possible to define functions which are not polytime computable.

We have related extensions of $\mathbb{III}(\Sigma)$ with system $\mathfrak{B}$. It would be interesting to try to relate extensions of $\mathbb{IIL}(\Sigma)$ with monoidal products, function spaces and exponential types to the type system SLR. Doing this might potentially bring the same benefits as those provided by having a dual-context formulation of intuitionistic linear logic. It might be also interesting to consider another type system developed by Schwichtenberg and Bellantoni [SB02], which is related to SLR.

One can also extend $\mathbb{III}(\Sigma)$ with function types and investigate what happens when safe natural numbers and function spaces are mixed. In the light of Hofmann's observations about linearity restrictions on safe variables, we conjecture that it would be possible to define functions, which are not polytime computable.

Our reformulation of safe natural number objects as safe initial algebras for strong endofunctor can be used for considering safe versions of other inductive data-types. In particularly, a notion of 'safe tree' type can be studied in this setting. It would be interesting to see how such safe trees relate to the 'feasible trees' of Hofmann's SLR [Hof99].

A completely different direction is inspired by Leivant's characterization of polytime computable functions [Lei93, LM93] mentioned in Chapter 1. It suggest a generalization of dual-context calculi, in which contexts are divided into more than two

zones. Another possible motivation for considering such a generalization is Otto's characterization of Kalmar's elementary functions, which uses three kinds of variables. An interpretation of such a generalized calculus would require modifying the notion of dual-context multicategories. One way of doing such modification is to consider a notion of T-multicategories [Lei04], which can be roughly described as multicategories with structures defined by functor T as domains of multiarrows.

Appendix A

Single-context multicategories

A.1 Monoidal multicategories

In this section we recall a definition of *multicategories* taken from [Her00]. We shall call them *monoidal multicategories* to distinguish from *cartesian multicategories* introduced in the next section. [Lei04] is a more recent source for multicategories.

Definition A.1.1 (Monoidal multicategory)**.**
A **monoidal multicategory M** consists of

- a set $|\mathbf{M}|$, whose elements are called the **objects**
- for each sequence of objects $a_1 \dots a_n$ and each object b a set $\mathbf{M}(a_1 \dots a_n, b)$, whose elements are called **multiarrows** and denoted as $a_1 \dots a_n \xrightarrow{f} b$
- for each object bb, each sequence of objects $a_1 \dots a_n$ and each sequence of sequences of objects $\bar{a}_1 \dots \bar{a}_n$ a function be called **composition**:

$$\mathbf{M}(a_1 \dots a_n, b) \times \prod_{i=1}^{n} \mathbf{M}(\bar{a}_i, a_i) \to \mathbf{M}(\bar{a}_1 \dots \bar{a}_n, b)$$

  where $\bar{a}_1 \dots \bar{a}_n$ stands for concatenation of sequences $\bar{a}_1, \dots, \bar{a}_n$. The composition will be denoted as $f \circ \langle g_1, \dots, g_n \rangle$.

- for each object a a multiarrow $id_a \in \mathbf{M}(a, a)$ called **identity**

satisfying the following laws:

- **associativity law**

$$f \circ \langle g_1 \circ \langle \bar{h}_1 \rangle \dots g_n \circ \langle \bar{h}_n \rangle \rangle = (f \circ \langle \bar{g} \rangle) \circ \langle \bar{h}_1 \dots \bar{h}_n \rangle$$

  whenever f, g_i, h_j are multiarrows for which these composites make sense.

- **identity laws**

$$f \circ \langle id_{a_1} \dots id_{a_n} \rangle = f = id_a \circ \langle f \rangle$$

where $a_1 \dots a_n \xrightarrow{f} a$.

We are interested in the following special kind of monoidal multicategories:

Definition A.1.2.
A **symmetric** monoidal multicategory is a monoidal multicategory **M** equipped with symmetry maps

$$- \cdot \sigma : \mathbf{M}(a_1 \dots a_n, b) \to \mathbf{C}(a_{\sigma(1)} \dots a_{\sigma(n)}, b)$$

for each object b, each sequence of objects $a_1 \dots a_n$ and each permutation σ which satisfy the following properties:

- $f \cdot 1 = 1$ where 1 is the identity permutation
- $(f \cdot \sigma) \cdot \tau = f \cdot (\sigma \circ \tau)$ where $\sigma \circ \tau$ is a composition of permutations σ and τ.
- Compatibility with composition:

$$(f \cdot \sigma) \circ \langle g_{\sigma(1)} \cdot \pi_{\sigma(1)} \cdots g_{\sigma(n)} \cdot \pi_{\sigma(n)} \rangle = (f \circ \langle g_1 \dots g_n \rangle) \cdot (\sigma \circ (\pi_{\sigma(1)} \dots \pi_{\sigma(n)}))$$

Alternatively, symmetric monoidal multicategories can be defined using labeled families of objects $[x_1 : a_1, \dots, x_n : a_n]$ as domains of multiarrows instead of sequences of objects. We shall refer to such labeled families as **contexts** and use capital Greek letters to range over them. Concatenation of sequences will be replaced with union of contexts which have mutually disjoint sets of labels. Such unions will be denoted as $\Gamma_1, \dots, \Gamma_n$. Formally such multicategories are defined as follows [Lei04] (they are called fat since they contain essentially the same sets of multiarrows with different labels in contexts):

Definition A.1.3.
A **fat symmetric monoidal multicategory M** consists of:

- A set of objects $|\mathbf{M}|$
- For each context $[x_1 : a_1, \dots, x_n : a_n]$ and each object b a set of multiarrows $\mathbf{M}([x_1 : a_1, \dots, x_n : a_n], b)$

- For each object c, each context $[x_1 : b_1, \ldots, x_n : b_n]$, each family of mutually disjoint contexts $\{\Gamma\}_{x_i \in \{x_1,\ldots,x_n\}}$ a composition map:

$$\mathbf{M}([x_1 : b_1, \ldots, x_n : b_n], c) \times \prod_{x_i} \mathbf{M}([\Gamma_{x_i}], b_i) \to \mathbf{M}([\Gamma_{x_1}, \ldots, \Gamma_{x_n}], c)$$

- For each context $[x : a]$ an identity multiarrow $id_{[x:a]} \in \mathbf{M}([x : a], a)$

satisfying the following laws:

- **Associativity law**

$$f \circ \langle g_{x_1} \circ \langle \bar{h}_1 \rangle, \ldots, g_{x_n} \circ \langle \bar{h}_n \rangle \rangle = (f \circ \langle g_{x_1}, \ldots, g_{x_n} \rangle) \circ \langle \bar{h}_1, \ldots, \bar{h}_n \rangle$$

- **Identity laws**

$$f \circ id_{[x_1:a_1]}, \ldots, id_{[x_n:a_n]} \rangle = f = id_{[x:a]} \circ langle f \rangle$$

Leinster showed that the two definitions of symmetric multicategories are equivalent [Lei04].

We now recall how monoidal multicategories and categories can be related.

Definition A.1.4.
Given a monoidal multicategory $\mathbf{M}$ we define a **base category** of $\mathbf{M}$ denoted as $\hat{\mathbf{M}}$ as follows:

- For the set of objects $|\hat{\mathbf{M}}|$ take $|\mathbf{M}|$. For a set of morphisms $\hat{\mathbf{M}}(a,b)$ take a set of multiarrows $\mathbf{M}(a,b)$.

- Composition of $f \in \hat{\mathbf{M}}(b,a)$ and $g \in \hat{\mathbf{M}}(c,b)$ will be defined using the composition of $\mathbf{M}$ as $f \circ g = f \circ \langle g \rangle$. For identity morphisms in $\hat{\mathbf{M}}$ take identity multiarrow of $\mathbf{M}$

Proposition A.1.5. $\hat{\mathbf{M}}$ *is a category.*

Proof. Routine check. □

Definition A.1.6.
A monoidal multicategory $\mathbf{M}$ is called **representable** if for every sequence of objects $a_1 \ldots a_n$ there exists a **universal** multiarrow $m_{\bar{a}} : a_1 \ldots a_n \to \otimes \bar{a}$ such that:

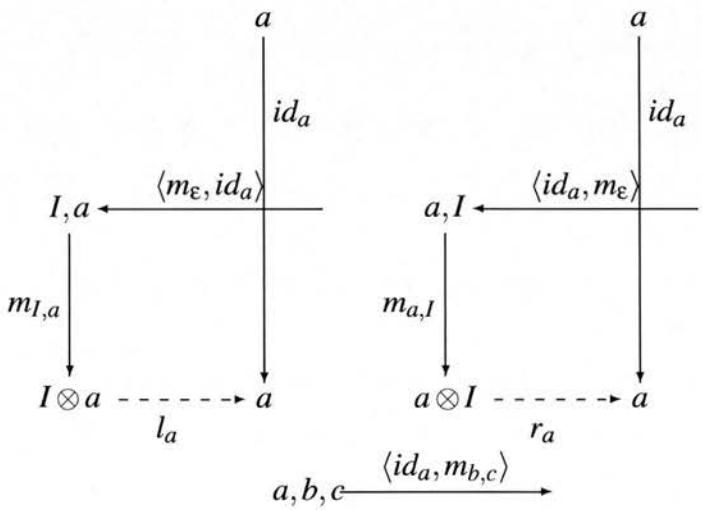
- For any multiarrow $f \in \mathbf{M}(a_1 \ldots a_n, b)$ there exists a unique multiarrow $\hat{f} \in \mathbf{M}(\otimes \bar{a}, b)$ such $\hat{f} \circ \langle m_{\bar{a}} \rangle = f$

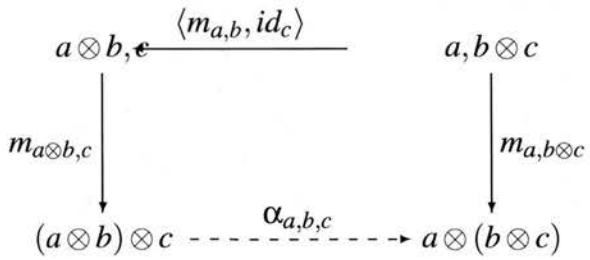
- Universal arrows are closed under composition

Proposition A.1.7. *If a monoidal multicategory* $\mathbf{M}$ *is representable then it's base category* $\hat{\mathbf{M}}$ *is monoidal.*

Proof. Define monoidal structure on $\hat{\mathbf{M}}$ as follows:

- For any objects a and b take $a \otimes b$ to be a codomain of some universal arrow m_{ab}, for I take a codomain of some universal arrow m_ε
- Structural isomorphisms are constructed using universal arrows as follows:





- We do skip checking the coherence diagrams, details can be found in [Her00]

□

In [Her00] the following notion of the functor between the monoidal multicategories is introduced:

Definition A.1.8.

Given a pair of monoidal multicategories $\mathbf{M_1}$ and $\mathbf{M_2}$ a **functor** $T : \mathbf{M}_1 \to \mathbf{M}_2$ is given by the following family of maps:

- A map of objects $T_0 : |\mathbf{M}_1| \to |\mathbf{M}_2|$
- For any sequence $a_1 \dots a_n$ a map of multiarrows

$$T_{a_1 \dots a_n} : \mathbf{M}_1(a_1 \dots a_n, a) \to \mathbf{M}_2(T_0(a_1) \dots T_0(a_n), T_0(a))$$

satisfying the following properties:

- $T_a(id_a) = id_{T_0 a}$
- $T_{\bar{b}_1 \dots \bar{b}_n}(f \circ \langle g_1 \dots g_n \rangle) = T_{a_1 \dots a_n}(f) \circ \langle T_{\bar{b}_1}(g_1) \dots T_{\bar{b}_n}(g_n) \rangle$

Hermida showed that such functors between monoidal multicategories correspond to monoidal functors between the underlying monoidal categories.

A.2 Cartesian multicategories

In the previous section we have introduced monoidal multicategories, which were connected to monoidal categories. In this section we shall consider special multicategories corresponding to cartesian categories.

Definition A.2.1 (Cartesian multicategory)**.**
A **cartesian multicategory C** consists of

- a set $|\mathbf{C}|$, whose elements are called the **objects**
- for each sequence of objects $a_1 \dots a_n$ and each object a a set $\mathbf{C}(a_1 \dots a_n, a)$, whose elements are called **multiarrows** and denoted as $a_1 \dots a_n \xrightarrow{f} a$
- for each object c, each sequence of objects $a_1 \dots a_n$ and each sequence of objects $b_1 \dots b_m$ a function which is called **composition**:

$$\mathbf{C}(b_1 \dots b_m, c) \times \prod_{i=1}^{m} \mathbf{C}(a_1 \dots a_n, b_i) \to \mathbf{C}(a_1 \dots a_n, c)$$

The composition will be denoted as $f \circ_{\bar{a}} \langle g_1 \dots g_n \rangle$ where $\bar{a}$ stands for $a_1 \dots a_n$. We shall use ε to denote the empty sequence.

- for each sequence $a_1 \dots a_n$ and each $1 \leq i \leq n$ a multiarrow $\pi_i \in \mathbf{C}(a_1 \dots a_n, a_i)$ which is called **projection**

satisfying the following laws:

- **associativity law**

$$f \circ_{\bar{a}} \langle g_1 \circ_{\bar{a}} \langle \bar{h} \rangle \dots g_n \circ_{\bar{a}} \langle \bar{h} \rangle \rangle = (f \circ_{\bar{b}} \langle \bar{g} \rangle) \circ_{\bar{a}} \langle \bar{h} \rangle$$

  where $h_i \in \mathbf{C}(\bar{a}, b_i)$, $g_j \in \mathbf{C}(b_1 \dots b_m, c_j)$ and $f \in \mathbf{M}(c_1 \dots c_n, d)$ and $\bar{b} = b_1 \dots b_m$.

- **identity laws**

$$f \circ_{\bar{a}} \langle \pi_1 \dots \pi_n \rangle = f \qquad \text{where} \quad a_1 \dots a_n \xrightarrow{f} a \qquad a_1 \dots a_n \xrightarrow{\pi_i} a_i$$

$$\pi_i \circ_{\bar{b}} \langle g_1 \dots g_n \rangle = g_i \qquad \text{where} \quad a_1 \dots a_n \xrightarrow{\pi_i} a_i \qquad b_1 \dots b_m \xrightarrow{g_i} a_i$$

Comparing cartesian and monoidal categories we see that the main differences is in composition scheme. In monoidal case we can compose $f : b_1 \dots b_n \to c$ with a family $g_i : \bar{a}_i \to b_i$ and get a multiarrow in domain $\bar{a}_1 \dots \bar{a}_n$. In cartesian case all g_i must have the same domain $\bar{a}$ which will become domain of a composite multiarrow as well. Another difference is that identity multiarrows in cartesian case are not restricted to singleton domains. Using these two features of cartesian multicategories it is possible to define several operations on multiarrows:

- *Interchange* which produces a multiarrows in $\mathbf{C}(\bar{a}_1.c.\bar{a}_2.b.\bar{a}_3, d)$ from a multiarrow in $\mathbf{C}(\bar{a}_1.b.\bar{a}_2.c.\bar{a}_3, d)$ can be defined as $f \circ_{\bar{a}_1.c.\bar{a}_2.b.\bar{a}_3} \langle \bar{\pi}_{a_1}\ \pi_b\ \bar{\pi}_{a_2}\ \pi_c\ \bar{\pi}_{a_3} \rangle$ where $\bar{\pi}_{\bar{a}}$ denotes a sequence of identity multiarrows for each of a sequence $\bar{a}$.

- *Weakening* which produces a multiarrow in $\mathbf{C}(\bar{a}.b, c)$ from a multiarrow in $\mathbf{C}(\bar{a}, c)$ can be defined as $f \circ_{\bar{a}.b} \langle \bar{\pi}_{\bar{a}} \rangle$

- *Contraction* which produces a multiarrows in $\mathbf{C}(\bar{a}.b, c)$ from a multiarrow in $\mathbf{C}(\bar{a}.b.b, c)$ can be defined as $f \circ_{\bar{a}.b} \langle \bar{\pi}_a\ \pi_b\ \pi_b \rangle$

Lambek [Lam89] gave a different definition of cartesian multicategories by adding interchange, weakening and contraction operations with appropriate laws to monoidal multicategories. It can be shown that such definition is equivalent to ours.

Definition A.2.2.
Given a cartesian category $\mathbf{C}$ an **underlying** cartesian multicategory $\mathbf{U}_C$ consists of:

- A set of objects $|\mathbf{U}_C| = |\mathbf{C}|$

- For each object b and each sequence of objects $a_1 \dots a_n$ a set of multiarrows

$$\mathbf{U}_C(a_1 \dots a_n, b) = \mathbf{C}(a_1 \times \dots \times a_n, b)$$

- For each sequence of objects $a_1 \dots a_n$ and each object a_i within such sequence a projection multiarrow $\pi_i : a_1 \dots a_n \to a_i$ given projection morphism π_i
- For each object c, each sequence of objects $a_1 \dots a_n$ and each sequence of objects $b_1 \dots b_m$ a composition function $f \circ_{\bar{a}} \langle g_1, \dots, g_n \rangle = f \circ [g_1, \dots, g_n]$

Proposition A.2.3. $\mathbf{U}_C$ *is a cartesian multicategory.*

Proof. The identity and associativity laws of cartesian multicategory follow from the properties of product on $\mathbf{C}$. □

Definition A.2.4.
Given a cartesian multicategory $\mathbf{C}$ we define a **base category** of $\mathbf{C}$ denoted as $\hat{\mathbf{C}}$ as follows:

- For the set of objects $|\hat{\mathbf{C}}|$ take $|\mathbf{C}|$. For a set of morphisms $\hat{\mathbf{C}}(a,b)$ take a set of unary multiarrows $\mathbf{C}(a,b)$.
- Composition of $f \in \hat{\mathbf{C}}(b,a)$ and $g \in \hat{\mathbf{C}}(c,b)$ will be defined using the composition of $\mathbf{C}$ as $f \circ g = f \circ_c \langle g \rangle$. For the identity morphism id_a take $\pi_1 \in \mathbf{C}(a,a)$.

Proposition A.2.5. $\hat{\mathbf{C}}$ *is a category.*

Definition A.2.6.
A cartesian multicategory $\mathbf{C}$ is called **representable** if for every sequence of objects $a_1 \dots a_n$ there exists a **universal** multiarrow $p_{\bar{a}} : a_1 \dots a_n \to \times \bar{a}$ such that for any multiarrow $f \in \mathbf{C}(a_1 \dots a_n, b)$ exists a unique multiarrow $\hat{f} \in \mathbf{C}(\times \bar{a}, b)$ such that $f = \hat{f} \circ_{\bar{a}} \langle p_{\bar{a}} \rangle$.

Proposition A.2.7. *If* $\mathbf{C}$ *is a representable cartesian multicategory then the corresponding base category* $\hat{\mathbf{C}}$ *has finite products.*

Proof. We define finite product structure on $\hat{\mathbf{C}}$ as follows:

- For a product $a \times b$ take codomain of the universal multiarrow p_{ab}. For projection maps take $\hat{\pi}_1 \in \hat{\mathbf{C}}(a \times b, a)$ and $\hat{\pi}_2 \in \hat{\mathbf{C}}(a \times b, b)$.

  Given morphisms $f \in \hat{\mathbf{C}}(d,a)$ and $g \in \hat{\mathbf{C}}(d,b)$ take $[f,g] = p_{ab} \circ_d \langle f,g \rangle$. For such choice we have

  $$\hat{\pi}_1 \circ [f,g] = (\hat{\pi}_1 \circ_{ab} \langle p_{ab} \rangle) \circ_d \langle f,g \rangle = \pi_1 \circ_d \langle f,g \rangle = f$$

  Similarly we have $\hat{\pi}_2 \circ [f,g] = g$.

Given any morphism $t \in \hat{\mathbf{C}}(d, a \times b)$ such that $\hat{\pi}_1 \circ t = f$ and $\hat{\pi}_2 \circ t = g$ we have the following:

$$[f,g] = p_{ab} \circ_d \langle f, f \rangle = p_{ab} \circ_d \langle \hat{\pi}_1 \circ t, \hat{\pi}_2 \circ t \rangle = (p_{ab} \circ_{a \times b} \langle \hat{\pi}_1, \hat{\pi}_2 \rangle) \circ_d \langle t \rangle$$

p_{ab} is universal so we must have that $p_{ab} \circ_{a \times b} \langle \hat{\pi}_1, \hat{\pi}_2 \rangle = id_{a \times b}$ since the following holds:

$$(p_{ab} \circ_{a \times b} \langle \hat{\pi}_1, \hat{\pi}_2 \rangle) \circ_{ab} \langle p_{ab} \rangle = p_{ab} \circ_{ab} \langle \pi_1, \pi_2 \rangle = p_{ab}$$

So we can conclude that $t = [f,g]$.

- For the terminal object 1 take a codomain of a universal multiarrow p_ε. We have that $p_\varepsilon \circ_1 \langle\rangle = id_1$ since p_ε is universal and $(p_\varepsilon \circ_1 \langle\rangle) \circ_\varepsilon \langle p_\varepsilon \rangle = p_\varepsilon \circ_\varepsilon \langle\rangle = p_\varepsilon$.

  For any object a we have a morphism $p_\varepsilon \circ_a \langle\rangle \in \hat{\mathbf{C}}(a, 1)$. Given any other morphism $f \in \hat{\mathbf{C}}(a, 1)$ we have the following:

$$f = id_1 \circ_a \langle f \rangle = (p_\varepsilon \circ_1 \langle\rangle) \circ_a \langle f \rangle = p_\varepsilon \circ_a \langle\rangle$$

  So $p_\varepsilon \circ_a \langle\rangle$ is a unique morphism in $\hat{\mathbf{C}}(a, 1)$.

□

Appendix B

Proofs for R-multicategories

Proposition B.0.8. **M** *is an Π-multicategory.*

Proof. • N-Id

$$\pi_{x_i} \circ_{[\Gamma;]} \langle g_{x_1}, \ldots, g_{x_n} ; \rangle = (\pi_x \cdot \nu) \circ \langle g_{x_1} \cdot \varepsilon_{x_1}, \ldots, g_{x_n} \cdot \varepsilon_{x_n} ; \rangle$$

Using Ren-Comp-2 we get

$$(\pi_x \circ \langle (g_{x_i} \cdot \varepsilon_{x_i}) \cdot \tau ; \rangle) \cdot \sigma$$

where

$$\varepsilon_{x_i} : [\Pi ; \Theta] \to [\Pi' ; \Theta'] \quad \tau : [\Pi' ; \Theta'] \to [\Pi'' ; \Theta''] \quad \sigma : [\Pi'' ; \Theta''] \to [\Pi ; \Theta]$$

are isomorphic renamings. Using Ren-Sup and R-N-Id we get $g_{x_i} \cdot (\sigma \circ (\tau \circ \varepsilon_{x_i}))$ which is equal to g_{x_i} by Ren-Id.

- S-Id is similar to the previous case.
- **Associative law**

$$f \circ_{[\Pi;\Theta]} \langle g_{x_1} \circ_{[\Pi;]} \langle \bar{t} ; \rangle \ldots g_{x_n} \circ_{[\Pi;]} \langle \bar{t} ; \rangle ; h_{y_1} \circ_{[\Pi;\Theta]} \langle \bar{t} ; \bar{s} \rangle, \ldots h_{y_m} \circ_{[\Pi;\Theta]} \langle \bar{t} ; \bar{s} \rangle \rangle = (f \circ_{[\Gamma;\Delta]} \langle \bar{g} ; \bar{h} \rangle) \circ_{[\Pi;\Theta]} \langle \bar{t} ; \bar{s} \rangle$$

$$f : [x_1 : a_1, \ldots, x_n : a_n ; y_1 : b_1, \ldots, y_m : b_m] \to c$$
$$g_{x_i} : [\Gamma ;] \to a_i \qquad h_{y_j} : [\Gamma ; \Delta] \to b_j$$
$$[\Gamma ; \Delta] = [u_1 : d_1, \ldots, u_k : d_k ; v_1 : e_1, \ldots, v_l : e_l]$$
$$t_{u_i} : [\Pi ;] \to d_i \qquad s_{v_j} : [\Pi ; \Theta] \to e_j$$

We assume the following renamings:

– Isomorphic renamings on $[\Gamma\,;\Delta]$

$$\varepsilon_{x_i} : [\Gamma\,;\,] \to [\Gamma_{x_i}\,;\,] \qquad \varepsilon_{y_j} : [\Gamma\,;\Delta] \to [\Gamma_{y_j}\,;\Delta_{y_j}]$$

– Contraction renaming on $[\Gamma\,;\Delta]$

$$\sigma : [\Gamma_{x_1},\ldots,\Gamma_{x_n},\Gamma_{y_1},\ldots,\Gamma_{y_m}\,;\Delta_{y_1},\ldots,\Delta_{y_m}] \to [\Gamma\,;\Delta]$$

– Isomorphic renamings on $[\Pi\,;\Theta]$

$$\begin{aligned}
&\tau_{u_i} : [\Pi\,;\,] \to [\Pi_{u_i}\,;\,] && \tau_{v_j} : [\Pi\,;\Theta] \to [\Pi_{v_j}\,;\Theta_{v_j}]\\
&\tau_{x_i} : [\Pi\,;\,] \to [\Pi_{x_i}\,;\,] && \tau_{y_j} : [\Pi\,;\Theta] \to [\Pi_{y_j}\,;\Theta_{y_j}]\\
&\tau^{x_i}_{u_j} : [\Pi_{u_j}\,;\,] \to [\Pi^{x_i}_{u_j}\,;\,] && \tau^{y_i}_{u_j} : [\Pi_{u_j}\,;\,] \to [\Pi^{y_i}_{u_j}\,;\,]\\
&\tau^{y_i}_{v_j} : [\Pi_{v_j}\,;\Theta_{v_j}\,;\,] \to [\Pi^{y_i}_{v_j}\,;\Theta^{y_i}_{v_j}]
\end{aligned}$$

– Contraction renaming on $[\Pi\,;\Theta]$

$$\begin{aligned}
&\mu_{u,v} : [\Pi_{u_1},\ldots,\Pi_{u_k},\Pi_{v_1},\ldots,\Pi_{v_l}\,;\Theta_{v_1},\ldots,\Theta_{v_l}] \to [\Pi\,;\Theta]\\
&\mu_{x_i} : [\Pi^{x_i}_{u_1},\ldots,\Pi^{x_i}_{u_k}\,;\,] \to [\Pi\,;\,]\\
&\mu_{y_j} : [\Pi^{y_j}_{u_1},\ldots,\Pi^{y_j}_{u_k},\Pi^{y_j}_{v_1},\ldots,\Pi^{y_j}_{v_l}\,;\Theta^{y_j}_{v_1},\ldots,\Theta^{y_j}_{v_l}] \to [\Pi\,;\Theta]\\
&\mu_{x,y} : [\Pi_{x_1},\ldots,\Pi_{x_n},\Pi_{y_1},\ldots,\Pi_{y_m}\,;\Theta_{y_1},\ldots,\Theta_{y_m}] \to [\Pi\,;\Theta]
\end{aligned}$$

– Contraction renaming

$$\mu : [\bar{\Pi}^{x_1},\ldots,\bar{\Pi}^{x_n},\bar{\Pi}^{y_1},\ldots,\bar{\Pi}^{y_m}\,;\bar{\Theta}^{y_1},\ldots,\bar{\Theta}^{y_m}] \to [\Pi_{u_1},\ldots,\Pi_{u_k},\Pi_{v_1},\ldots,\Pi_{v_l}\,;\Theta_{v_1},\ldots,\Theta_{v_l}]$$

$$\bar{\Pi}^{x_i} = \Pi^{x_i}_{u_1},\ldots,\Pi^{x_i}_{u_k} \qquad \bar{\Pi}^{y_j} = \Pi^{y_j}_{u_1},\ldots,\Pi^{y_j}_{u_k} \qquad \bar{\Theta}^{y_j} = \Theta^{y_j}_{u_1},\ldots,\Theta^{y_j}_{u_k}$$

$$\mu(\nu^h_l) = \{$$

We use the following family abbreviations:

$$\begin{aligned}
\vec{t}^{x_i} &= t_{u_1}\cdot\tau_{u_1}\cdot\tau^{x_i}_{u_1},\ldots,t_{u_k}\cdot\tau_{u_k}\cdot\tau^{x_i}_{u_k}\\
\vec{t}^{y_j} &= t_{u_1}\cdot\tau_{u_1}\cdot\tau^{y_j}_{u_1},\ldots,t_{u_k}\cdot\tau_{u_k}\cdot\tau^{y_j}_{u_k}\\
\vec{s}^{y_j} &= s_{v_1}\cdot\tau_{v_1}\cdot\tau^{y_j}_{v_1},\ldots,s_{v_l}\cdot\tau_{v_l}\cdot\tau^{y_j}_{v_l}
\end{aligned}$$

Righ-hand side can be rewritten as follows:

$$(f \circ \langle g_{x_1} \cdot \varepsilon_{x_1}, \ldots, g_{x_n} \cdot \varepsilon_{x_n} \,;\, h_{y_1} \cdot \varepsilon_{y_1}, \ldots, h_{y_m} \cdot \varepsilon_{y_m} \rangle \cdot \sigma) \circ \\ \langle t_{u_1} \cdot \tau_{u_1}, \ldots, t_{u_k} \cdot \tau_{u_k} \,;\, s_{v_1} \cdot \tau_{v_1}, \ldots, s_{v_l} \cdot \tau_{v_l} \rangle \cdot \mu_{u,v}$$

By applying Ren-Comp-2 we get

$$(f \circ \langle g_{x_1} \cdot \varepsilon_{x_1}, \ldots, g_{x_n} \cdot \varepsilon_{x_n} \,;\, h_{y_1} \cdot \varepsilon_{y_1}, \ldots, h_{y_m} \cdot \varepsilon_{y_m} \rangle \circ \\ \langle \vec{t}^{x_1}, \ldots, \vec{t}^{x_n}, \vec{t}^{y_1}, \ldots, \vec{t}^{y_m} \,;\, \vec{s}^{y_1}, \ldots, \vec{s}^{y_m} \rangle) \cdot \mu \cdot \mu_{u,v}$$

By associativity law of R-multicategories we get

$$f \circ \langle (g_{x_1} \cdot \varepsilon_{x_1}) \circ \langle \vec{t}^{x_1} \,;\, \rangle, \ldots, (g_{x_n} \cdot \varepsilon_{x_n}) \circ \langle \vec{t}^{x_n} \,;\, \rangle \,; \\ (h_{y_1} \cdot \varepsilon_{y_1}) \circ \langle \vec{t}^{y_1} \,;\, \vec{s}^{y_1} \rangle, \ldots, (h_{y_m} \cdot \varepsilon_{y_m}) \circ \langle \vec{t}^{y_m} \,;\, \vec{s}^{y_m} \rangle \rangle) \cdot \mu \cdot \mu_{u,v}$$

Since ε_{x_i} and εy_j are isomorphism renamings we can apply Lemma and get

$$f \circ \langle g_{x_1} \circ \langle \vec{t}^{x_1} \,;\, \rangle, \ldots, g_{x_n} \circ \langle \vec{t}^{x_n} \,;\, \rangle \,;\, h_{y_1} \circ \langle \vec{t}^{y_1} \,;\, \vec{s}^{y_1} \rangle, \ldots, h_{y_m} \circ \langle \vec{t}^{y_m} \,;\, \vec{s}^{y_m} \rangle \rangle \cdot \mu \cdot \mu_{u,v}$$

Left-hand side can be rewritten as follows:

$$f \circ \langle g_{x_1} \circ \langle \vec{t}^{x_1} \,;\, \rangle \cdot \mu_{x_1} \cdot \tau_{x_1}, \ldots, g_{x_n} \circ \langle \vec{t}^{x_n} \,;\, \rangle \cdot \mu_{x_n} \cdot \tau_{x_n} \,;\, h_{y_1} \circ \langle \vec{t}^{y_1} \,;\, \vec{s}^{y_1} \rangle \cdot \mu_{y_1} \cdot \tau_{y_1}, \ldots, \\ h_{y_m} \circ \langle \vec{t}^{y_m} \,;\, \vec{s}^{y_m} \rangle \cdot \mu_{y_m} \cdot \tau_{y_m} \rangle \cdot \mu_{x,y}$$

Using Ren-Comp-1 we get

$$f \circ \langle g_{x_1} \circ \langle \vec{t}^{x_1} \,;\, \rangle, \ldots, g_{x_n} \circ \langle \vec{t}^{x_n} \,;\, \rangle \,;\, h_{y_1} \circ \langle \vec{t}^{y_1} \,;\, \vec{s}^{y_1} \rangle, \ldots, h_{y_m} \circ \langle \vec{t}^{y_m} \,;\, \vec{s}^{y_m} \rangle \rangle \cdot \mu_{x,y} \circ \\ (\tau_{x_1} \circ \mu_{x_1} + \cdots + \tau_{x_n} \circ \mu_{x_n} + \tau_{y_1} \circ \mu_{y_1} + \cdots + \tau_{y_m} \circ \mu_{y_m})$$

□

Proposition B.0.9. $\hat{\mathbf{M}}$ *is an R-multicategory.*

- We have

$$f : [x_1 : a_1, \ldots, x_n : a_n \,;\, y_1 : b_1, \ldots, y_m : b_m] \to c$$

$$g_{x_i} : [\Gamma_{x_i} \,;\,] \to a_i \qquad \varepsilon_{x_i} : [\Gamma_{x_i} \,;\,] \to [\Pi_{x_i} \,;\,]$$

$$h_{y_j} : [\Gamma_{y_j} \,;\, \Delta_{y_j}] \to b_j \qquad \varepsilon_{y_j} : [\Gamma_{y_j} \,;\, \Delta_{y_j}] \to [\Pi_{y_j} \,;\, \Theta_{y_j}]$$

$$[\Gamma \,;\, \Delta] = [\Gamma_{x_1}, \ldots, \Gamma_{x_n}, \Gamma_{y_1}, \ldots, \Gamma_{y_m} \,;\, \Delta_{y_1}, \ldots, \Delta_{y_m}]$$

$$[\Pi \,;\, \Theta] = [\Pi_{x_1}, \ldots, \Pi_{x_n}, \Pi_{y_1}, \ldots, \Pi_{y_m} \,;\, \Theta_{y_1}, \ldots, \Theta_{y_m}]$$

$$[\Gamma_{x_i} \,;\,] = [u_1^{x_i} : c_1^{x_i}, \ldots, u_{k_{x_i}}^{x_i} : c_{k_{x_i}}^{x_i} \,;\,] \qquad [\Pi_{x_i} \,;\,] = [z_1^{x_i} : a_1^{x_i}, \ldots, z_{q_{x_i}}^{x_i} : a_{q_{x_i}}^{x_i} \,;\,]$$

$$[\Gamma_{y_i} \,;\, \Delta_{y_i}] = [u_1^{y_i} : d_1^{y_i}, \ldots, u_{k_{y_i}}^{y_i} : d_{k_{y_i}}^{y_i} \,;\, v_1^{y_i} : e_1^{y_i}, \ldots, v_{l_{y_i}}^{y_i} : e_{l_{y_i}}^{y_i}]$$

$$[\Pi_{y_i} \,;\, \Theta_{y_i}] = [z_1^{y_i} : a_1^{y_i}, \ldots, z_{q_{y_i}}^{y_i} : a_{q_{y_i}}^{y_i} \,;\, w_1^{y_i} : b_1^{y_i}, \ldots, w_{r_{y_i}}^{y_i} : b_{r_{y_i}}^{y_i}]$$

We need to show:

$$f \circ \langle g_{x_1} \cdot \varepsilon_{x_1}, \ldots, g_{x_n} \cdot \varepsilon_{x_n} \,;\, h_{y_1} \cdot \varepsilon_{y_1}, \ldots, h_{y_m} \cdot \varepsilon_{y_m} \rangle = (f \circ \langle g_{x_1}, \ldots, g_{x_n} \,;\, h_{y_1}, \ldots, h_{y_m} \rangle) \cdot (\varepsilon_{x_1} + \cdots + \varepsilon_{x_n} + \varepsilon_{y_1} + \cdots + \varepsilon_{y_m})$$

We introduce the following abbreviations:

$$\bar{\pi}_{\Pi_{x_i}} = \pi_{z_1^{x_i}} \ldots \pi_{z_{q_{x_i}}^{x_i}} \qquad \bar{\pi}_{\Pi_{y_i}} = \pi_{z_1^{y_i}} \ldots \pi_{z_{q_{y_i}}^{y_i}} \qquad \bar{\sigma}_{\Theta_{y_i}} = \sigma_{w_1^{y_i}}, \ldots, \sigma_{w_{r_{y_i}}^{y_i}}$$

$$\bar{\pi}_{\Gamma_{x_i}} = \pi_{u_1^{x_i}} \ldots \pi_{u_{k_{x_i}}^{x_i}} \qquad \bar{\pi}_{\Gamma_{y_i}} = \pi_{u_1^{y_i}} \ldots \pi_{u_{k_{y_i}}^{y_i}} \qquad \bar{\sigma}_{\Delta_{y_i}} = \sigma_{v_1^{y_i}}, \ldots, \sigma_{s_{l_{y_i}}^{v_i}}$$

$$\bar{\pi}_{\varepsilon_{x_i}} = \pi_{\varepsilon_{x_i}(u_1^{x_i})} \ldots \pi_{\varepsilon_{x_i}(u_{k_{x_i}}^{x_i})} \qquad \bar{\pi}_{\varepsilon_{y_i}} = \pi_{\varepsilon_{y_i}(u_1^{y_i})} \ldots \pi_{\varepsilon_{y_i}(u_{k_{y_i}}^{y_i})} \qquad \bar{\sigma}_{\varepsilon_{y_i}} = \sigma_{\varepsilon(v_1^{y_i})}, \ldots, \sigma_{\varepsilon(s_{l_{y_i}}^{v_i})}$$

Left-hand side can be rewritten as:

$$f \circ_{[\Pi;\Theta]} \langle g_{x_1} \circ_{[\Pi_{x_1};]} \langle \bar{\pi}_{\varepsilon_{x_1}} \,;\, \rangle \circ_{[\Pi;]} \langle \bar{\pi}_{\Pi_{x_1}} \,;\, \rangle, \ldots, g_{x_n} \circ_{[\Pi_{x_n};]} \langle \bar{\pi}_{\varepsilon_{x_n}} \,;\, \rangle \circ_{[\Pi;]} \langle \bar{\pi}_{\Pi_{x_n}} \,;\, \rangle \,;$$
$$h_{y_1} \circ_{[\Pi_{y_1};\Theta_{y_1}]} \langle \bar{\pi}_{\varepsilon_{y_1}} \,;\, \bar{\sigma}_{\varepsilon_{y_1}} \rangle \circ_{[\Pi;\Theta]} \langle \bar{\pi}_{\Pi_{y_1}} \,;\, \bar{\sigma}_{\Theta_{y_1}} \rangle, \ldots,$$
$$h_{y_m} \circ_{[\Pi_{y_m};\Theta_{y_m}]} \langle \bar{\pi}_{\varepsilon_{y_m}} \,;\, \bar{\sigma}_{\varepsilon_{y_m}} \rangle \circ_{[\Pi;\Theta]} \langle \bar{\pi}_{\Pi_{y_m}} \,;\, \bar{\sigma}_{\Theta_{y_m}} \rangle \rangle$$

By associativity and identity laws this is equal to:

$$f \circ_{[\Pi;\Theta]} \langle g_{x_1} \circ_{[\Pi;]} \langle \bar{\pi}_{\varepsilon_{x_1}} \,;\, \rangle, \ldots, g_{x_n} \circ_{[\Pi;]} \langle \bar{\pi}_{\varepsilon_{x_n}} \,;\, \rangle \,;$$
$$h_{y_1} \circ_{[\Pi;\Theta]} \langle \bar{\pi}_{\varepsilon_{y_1}} \,;\, \bar{\sigma}_{\varepsilon_{y_1}} \rangle, \ldots, h_{y_m} \circ_{[\Pi;\Theta]} \langle \bar{\pi}_{\varepsilon_{ym}} \,;\, \bar{\sigma}_{\varepsilon_{ym}} \rangle \rangle$$

Right-hand side can be rewritten as:

$$f \circ_{[\Gamma;\Delta]} \langle g_{x_1} \circ_{[\Gamma;]} \langle \bar{\pi}_{\Gamma_{x_1}} \,;\, \rangle, \ldots, g_{x_n} \circ_{[\Gamma;]} \langle \bar{\pi}_{\Gamma_{xn}} \,;\, \rangle \,;\, h_{y_1} \circ_{[\Gamma;\Delta]} \langle \bar{\pi}_{\Gamma_{y_1}} \,;\, \bar{\sigma}_{\Delta_{y_1}} \rangle, \ldots,$$
$$h_{y_m} \circ_{[\Gamma;\Delta]} \langle \bar{\pi}_{\Gamma_{ym}} \,;\, \bar{\sigma}_{\Delta_{ym}} \rangle \rangle \circ_{[\Pi;\Theta]} \langle \bar{\pi}_{\varepsilon_{x_1}}, \ldots, \bar{\pi}_{\varepsilon_{xn}}, \bar{\pi}_{\varepsilon_{y_1}}, \ldots, \bar{\pi}_{\varepsilon_{ym}} \,;\, \bar{\sigma}_{\varepsilon_{y_1}}, \ldots, \bar{\sigma}_{\varepsilon_{ym}} \rangle$$

By associativity and identity laws this is equal to:

$$f \circ_{[\Pi;\Theta]} \langle g_{x_1} \circ_{[\Pi;]} \langle \bar{\pi}_{\varepsilon_{x_1}} ; \rangle, \ldots, g_{x_n} \circ_{[\Pi;]} \langle \bar{\pi}_{\varepsilon_{x_n}} ; \rangle ; \\ h_{y_1} \circ_{[\Pi;\Theta]} \langle \bar{\pi}_{\varepsilon_{y_1}} ; \bar{\sigma}_{\varepsilon_{y_1}} \rangle, \ldots, h_{y_m} \circ_{[\Pi;\Theta]} \langle \bar{\pi}_{\varepsilon_{y_m}} ; \bar{\sigma}_{\varepsilon_{y_m}} \rangle \rangle$$

- Given

$$[\Gamma ; \Delta] = [x_1 : a_1, \ldots, x_n : a_n ; y_1 : b_1, \ldots, y_m : b_m]$$
$$[\Pi ; \Theta] = [u_1 : c_1, \ldots, u_k : c_k ; v_1 : d_1, \ldots, v_l : d_l]$$
$$\varepsilon : [\Gamma ; \Delta] \to [\Pi ; \Theta]$$
$$f : [\Gamma ; \Delta] \to e \qquad g_{u_i} : [\Pi_{u_i} ;] \to c_i \qquad h_{v_j} : [\Pi_{v_j} ; \Theta_{v_j}] \to d_j$$
$$g_{x_i} = g_{\varepsilon(x_i)} \cdot \sigma_{x_i} \qquad \sigma_{x_i} : [\Pi_{\varepsilon(x_i)} ;] \to [\Pi_{x_i} ;]$$
$$h_{y_j} = h_{\varepsilon(y_j)} \cdot \sigma_{y_j} \qquad \sigma_{y_j} : [\Pi_{\varepsilon(y_j)} ; \Theta_{\varepsilon(y_j)}] \to [\Pi_{y_j} ; \Theta_{y_j}]$$
$$[\Pi_u, \Pi_v ; \Theta_v] = [\Pi_{u_1}, \ldots, \Pi_{u_k}, \Pi_{v_1}, \ldots, \Pi_{v_l} ; \Theta_{v_1}, \ldots, \Theta_{v_l}]$$
$$[\Pi_x, \Pi_y, \Theta_y] = [\Pi_{x_1}, \ldots, \Pi_{x_n}, \Pi_{y_1}, \ldots, \Pi_{y_m} ; \Theta_{y_1}, \ldots, \Theta_{y_m}]$$
$$\sigma : [\Pi_x, \Pi_y, \Theta_y] \to [\Pi_u, \Pi_v ; \Theta_v]$$

We need to show the following:

$$(f \cdot \varepsilon) \circ \langle g_{u_1}, \ldots, g_{u_k} ; h_{v_1}, \ldots, h_{v_l} \rangle = (f \circ \langle g_{x_1}, \ldots, g_{x_n} ; h_{y_1}, \ldots, h_{y_m} \rangle) \cdot \sigma$$

Left-hand side is equal:

$$f \circ_{[\Pi;\Theta]} \langle \pi_{\varepsilon(x_1)}, \ldots, \pi_{\varepsilon(x_n)} ; \sigma_{\varepsilon(y_1)}, \ldots, \sigma_{\varepsilon(y_m)} \rangle \circ_{[\Pi_u, \Pi_v ; \Theta_v]} \langle g_{u_1} \circ_{[\Pi_u, \Pi_v ;]} \langle \bar{\pi}_{\Pi_{u_1}} ; \rangle, \ldots, \\ g_{u_k} \circ_{[\Pi_u, \Pi_v ;]} \langle \bar{\pi}_{\Pi_{u_k}} ; \rangle ; h_{v_1} \circ_{[\Pi_u, \Pi_v ; \Theta_v]} \langle \bar{\pi}_{\Pi_{v_1}} ; \bar{\sigma}_{\Theta_{v_1}} \rangle, \ldots, h_{v_l} \circ_{[\Pi_u, \Pi_v ; \Theta_v]} \langle \bar{\pi}_{\Pi_{v_l}} ; \bar{\sigma}_{\Theta_{v_l}} \rangle \rangle$$

By associativity and identity laws this is equal to:

$$f \circ_{[\Pi_u, \Pi_v ; \Theta_v]} \langle g_{\varepsilon(x_1)} \circ_{[\Pi_u, \Pi_v ;]} \langle \bar{\pi}_{\Pi_{\varepsilon(x_1)}} ; \rangle, \ldots, g_{\varepsilon(x_n)} \circ_{[\Pi_u, \Pi_v ;]} \langle \bar{\pi}_{\Pi_{\varepsilon(x_n)}} ; \rangle ; \\ h_{\varepsilon(y_1)} \circ_{[\Pi_u, \Pi_v ; \Theta_v]} \langle \bar{\pi}_{\Pi_{\varepsilon(y_1)}} ; \bar{\sigma}_{\Theta_{\varepsilon(y_1)}} \rangle, \ldots, h_{\varepsilon(y_m)} \circ_{[\Pi_u, \Pi_v ; \Theta_v]} \langle \bar{\pi}_{\Pi_{\varepsilon(y_m)}} ; \bar{\sigma}_{\Theta_{\varepsilon(y_m)}} \rangle \rangle$$

Right-hand side is equal to:

$$f \circ_{[\Pi_x, \Pi_y ; \Theta_y]} \langle g_{\varepsilon(x_1)} \circ_{[\Pi_x, \Pi_y ;]} \langle \bar{\pi}_{\Pi_{x_1}} ; \rangle, \ldots, g_{\varepsilon(x_n)} \circ_{[\Pi_x, \Pi_y ;]} \langle \bar{\pi}_{\Pi_{x_n}} ; \rangle ; \\ h_{\varepsilon(y_1)} \circ_{[\Pi_x, \Pi_y ; \Theta_y]} \langle \bar{\pi}_{\Pi_{y_1}} ; \bar{\sigma}_{\Theta_{y_1}} \rangle, \ldots, h_{\varepsilon(y_m)} \circ_{[\Pi_x . \Pi_y ; \Theta_y]} \langle \bar{\pi}_{\Pi_{y_m}} ; \bar{\sigma}_{\Theta_{y_m}} \rangle \rangle \\ \circ_{[\Pi_u, \Pi_v ; \Theta_v]} \langle \bar{\pi}_{\Pi_{\varepsilon(x_1)}}, \ldots, \bar{\pi}_{\Pi_{\varepsilon(x_n)}}, \bar{\pi}_{\Pi_{\varepsilon(y_1)}}, \ldots, \bar{\pi}_{\Pi_{\varepsilon(y_m)}} ; \bar{\sigma}_{\Theta_{\varepsilon(y_1)}}, \ldots, \bar{\sigma}_{\Theta_{\varepsilon(y_m)}} \rangle$$

By associativity and identity laws this is equal to:

$$f \circ_{[\Pi_u,\Pi_v;\Theta_v]} \langle g_{\varepsilon(x_1)} \circ_{[\Pi_u,\Pi_v;]} \langle \bar{\pi}_{\Pi_{\varepsilon(x_1)}} ; \rangle, \ldots, g_{\varepsilon(x_n)} \circ_{[\Pi_u,\Pi_v;]} \langle \bar{\pi}_{\Pi_{\varepsilon(x_n)}} ; \rangle ;$$
$$h_{\varepsilon(y_1)} \circ_{[\Pi_u,\Pi_v;\Theta_v]} \langle \bar{\pi}_{\Pi_{\varepsilon(y_1)}} ; \bar{\sigma}_{\Theta_{\varepsilon(y_1)}} \rangle, \ldots, h_{\varepsilon(y_m)} \circ_{[\Pi_u,\Pi_v;\Theta_v]} \langle \bar{\pi}_{\Pi_{\varepsilon(y_m)}} ; \bar{\sigma}_{\Theta_{\varepsilon(y_m)}} \rangle \rangle$$

□

Appendix C

Single-context strong endofunctors

In this section we study the notion of strong endofunctor on the single-context multicategory and show how it is related to the strong endofunctors on underlying monoidal category. To our knowledge strong endofunctors on single-context multicategories were not considered before in the literature on multicategories.

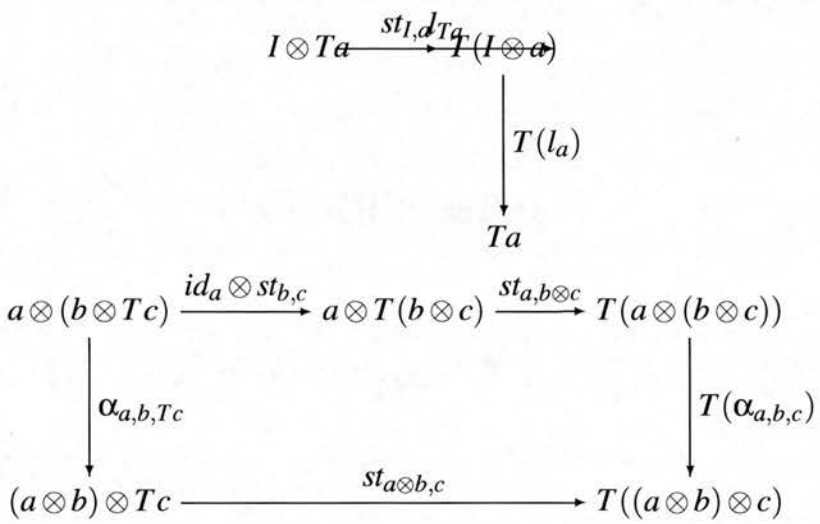
Definition C.0.10 (Strong endofunctor on cartesian category [Jac01]).
An endofunctor $T : \mathbf{C} \to \mathbf{C}$ on a cartesian category $\mathbf{C}$ is called **strong** if it comes equipped with a natural transformation $st_{a,b} : a \times Tb \to T(a \times b)$ called **strength** which makes the following diagrams commute:

$$\begin{array}{ccc} a \times T(b) & \xrightarrow{st_{a,b}} & T(a \times b) \\ & & \downarrow{\scriptstyle T(\pi_b)} \\ & & T(b) \end{array}$$

$$\begin{array}{ccccc} a \times (b \times T(c)) & \xrightarrow{id_a \times st_{b,c}} & a \times T(b \times c) & \xrightarrow{st_{a,b\times c}} & T(a \times (b \times c)) \\ \downarrow{\scriptstyle \cong} & & & & \downarrow{\scriptstyle \cong} \\ (a \times b) \times T(c) & & \xrightarrow{\qquad st_{a\times b,c} \qquad} & & T((a \times b) \times c) \end{array}$$

Definition C.0.11 (Strong endofunctor on monoidal category).
An endofunctor $T : \mathbf{S} \to \mathbf{S}$ on a monoidal category $\mathbf{S}$ is called **endofunctor with strength** if it comes with a natural transformation $st_{a,b} : a \otimes Tb \to T(a \otimes b)$ called

strength which makes the following diagrams commute:



C.1 Strong endofunctors on cartesian multicategories

Definition C.1.1.

A **strong endofunctor** T on a cartesian multicategory $\mathbf{C}$ is given by the following family of maps:

- A map of objects $T : |\mathbf{C}| \to |\mathbf{C}|$
- For each object b, each sequence of objects $a_1 \ldots a_n$ and each object a_i within this sequence a map of multiarrows

$$T_{a_i} : \mathbf{C}(a_1 \ldots a_n, b) \to \mathbf{C}(a_1 \ldots a_{i-1}\, T(a_i)\, a_{i+1} \ldots a_n, T(b)$$

These maps should satisfy the following properties:

- **Identity law** $T_{a_i}(\pi_{a_i}) = \pi_{T(a_i)}$
- **Unary composition law**

$$T_d(f \circ_{\bar{a}.d.\bar{b}} \langle \bar{\pi}_{\bar{a}}, g, \bar{\pi}_{\bar{b}} \rangle) = T_b(f) \circ_{\bar{a}.T(d).\bar{b}} \langle \bar{\pi}_{\bar{a}}, T_b(g), \bar{\pi}_{\bar{b}} \rangle$$

  where $f : \bar{a}.c.\bar{b} \to e$, $g : \bar{a}.d.\bar{b} \to c$ and $\bar{\pi}_{\bar{a}}$ denotes a sequence of projections from appropriate context to each $a_i \in \bar{a}$.

Definition C.1.2.

Given a strong endofunctor $T : \mathbf{C} \to \mathbf{C}$ on a representable cartesian multicategory we define an endofunctor $\hat{T}$ on a corresponding base category $\hat{\mathbf{C}}$ as follows:

$$\hat{T}(a) = T(a) \qquad \hat{T}(a \xrightarrow{f} b) = T_a(f)$$

Proposition C.1.3. *$\hat{T} : \hat{\mathbf{C}} \to \hat{\mathbf{C}}$ is a strong endofunctor.*

Proof. First check that $\hat{T}$ satisfies endofunctor properties:

- Given an identity morphism id_a we have $\hat{T}(id_a) = T_a(\pi_a) = \pi_{T(a)} = id_{T(a)}$ since T preserves identity multiarrows.
- Given a composition of $b \xrightarrow{f} c$ with $a \xrightarrow{g} b$ we have $\hat{T}(f \circ g) = T_a(f \circ_a \langle g \rangle) = T_b(f) \circ_{T(a)} \langle T_a(g) \rangle = \hat{T}(f) \circ \hat{T}(g)$ since T preserves unary composition.

Strength $st_{a,b} : a \times T(b) \to T(a \times b)$ is defined as the unique multiarrow such that $st_{a,b} \circ_{a.T(b)} \langle p_{a,T(b)} \rangle = T_b(p_{a,b})$.

- Naturality of $st_{a,b}$

$$\begin{array}{ccc} a \times T(b) & \xrightarrow{st_{a,b}} & T(a \times b) \\ \Big\downarrow{\scriptstyle f \times \hat{T}(g)} & & \Big\downarrow{\scriptstyle \hat{T}(f \times g)} \\ a' \times T(b') & \xrightarrow[st_{a',b'}]{} & T(a' \times b') \end{array}$$

Composing the upper part of this diagram with $p_{a,T(b)}$ we can show the following using associativity of composition and definition of $st_{a,b}$:

$$\begin{aligned} T_{a\times b}(f \times g) \circ_{a\times T(b)} \langle st_{a,b} \rangle) \circ_{a,T(b)} \langle p_{a,T(b)} \rangle = \\ T_{a\times b}(f \times g) \circ_{a,T(b)} \langle st_{a,b} \circ_{a,T(b)} \langle p_{a,T(b)} \rangle \rangle = \\ T_{a\times b}(f \times g) \circ_{a,T(b)} \langle T_b(p_{a,b}) \rangle \end{aligned}$$

Expanding $f \times g$ and using unary composition property this is equal to

$$T_b((p_{a',b'} \circ_{a\times b} \langle f \circ_{a\times b} \langle \hat{\pi}_a \rangle, g \circ_{a\times b} \langle \hat{\pi}_b \rangle \rangle) \circ_{a,b} \langle p_{a,b} \rangle)$$

Using associativity and the fact that $\hat{\pi}_a \circ_{a,b} \langle p_{a,b} \rangle = \pi_a$ we get

$$T_b((p_{a',b'} \circ_{a,b} \langle f \circ_{a,b} \langle \pi_a \rangle, g \circ_{a,b} \langle \pi_b \rangle \rangle)$$

Lower part of the diagram can be written as

$$st_{a',b'} \circ_{a\times T(b)} \langle p_{a',T(b')} \circ_{a\times T(b)} \langle f \circ_{a\times T(b)} \langle \hat{\pi}_a \rangle, T_b(g) \circ_{a\times T(b)} \langle \hat{\pi}_{T(b)} \rangle \rangle \rangle$$

Using associativity this is equal to

$$(st_{a',b'} \circ_{a',T(b')} \langle p_{a',T(b')} \rangle) \circ_{a\times T(b)} \langle f \circ_{a\times T(b)} \langle \hat{\pi}_a \rangle, T_b(g) \circ_{a\times T(b)} \langle \hat{\pi}_{T(b)} \rangle \rangle$$

which is by the definition of $st_{a',b'}$ equal to

$$T_{b'}(p_{a',b'}) \circ_{a\times T(b)} \langle f \circ_{a\times T(b)} \langle \hat{\pi}_a \rangle, T_b(g) \circ_{a\times T(b)} \langle \hat{\pi}_{T(b)} \rangle \rangle$$

Composing this with $p_{a,T(b)}$ and using associativity of composition and universal property of $p_{a,T(b)}$ we get:

$$T_{b'}(p_{a',b'}) \circ_{a,T(b)} \langle f \circ_{a,T(b)} \langle \pi_a \rangle, T_b(g) \circ_{a,T(b)} \langle \pi_{T(b)} \rangle \rangle$$

Using unary composition and identity property of T this is equal to

$$T_{b'}(p_{a',b'}) \circ_{a,T(b)} \langle f \circ_{a,T(b)} \langle \pi_a \rangle, T_b(g \circ_{a,b} \langle \pi_b \rangle) \rangle$$

- First diagram for $st_{a,b}$:

$$\begin{array}{ccc} a \times T(b) & \xrightarrow{st_{a,b}} & T(a \times b) \\ & & \downarrow{\scriptstyle T_{a\times b}(\hat{\pi}_b)} \\ & & T(b) \end{array}$$

The following diagram is just a definition of $st_{a,b}$:

$$\begin{array}{ccc} a, T(b) & \xrightarrow{T_b(p_{a,b})} & \\ \downarrow{\scriptstyle p_{a,T(b)}} & & \\ a \times T(b) & \xrightarrow[st_{a,b}]{} & T(a \times b) \end{array}$$

The following diagram commutes because of unary composition property of T:

$$\begin{array}{ccc} a, T(b) & \xrightarrow{\pi_{T(b)}} & \\ \downarrow{\scriptstyle T_b(p_{a,b})} & & \\ T(a \times b) & \xrightarrow[T_{a\times b}(\hat{\pi}_b)]{} & T(b) \end{array}$$

Putting these two diagrams together we get the following:

$$\begin{array}{ccccc} & & a, T(b) & \xrightarrow{\pi_{T(b)}} & \\ & & \downarrow{\scriptstyle T_b(p_{a,b})} & & \\ a \times T(b) & \xrightarrow[st_{a,b}]{p_{a,T(b)}} & T(a \times b) & \xrightarrow[T_{a\times b}(\hat{\pi}_b)]{} & T(b) \end{array}$$

Since the outer diagram commutes by universality of $p_{a,T(b)}$ we conclude that $T_{a\times b}(\hat{\pi}_b)\circ st_{a,b} = \hat{\pi}_{T(b)}$.

- Second diagram for $st_{a,b}$:

$$\begin{array}{ccccc} a\times(b\times T(c)) & \xrightarrow{id_a\times st_{b,c}} & a\times T(b\times c) & \xrightarrow{st_{a,b\times c}} & T(a\times(b\times c)) \\ \downarrow{\scriptstyle \alpha_{a,b,T(c)}} & & & & \downarrow{\scriptstyle T(\alpha_{a,b,c})} \\ (a\times b)\times T(c) & & \xrightarrow{\qquad st_{a\times b,c}\qquad} & & T((a\times b)\times c) \end{array}$$

Need to copy/modify proof from IL case.

□

C.2 Strong endofunctors on monoidal categories

Definition C.2.1 (Strong endofunctor on multicategory M)**.**
A **strong endofunctor** $T : \mathbf{M} \to \mathbf{M}$ on a multicategory $\mathbf{M}$ is the family of functions:

$T : |\mathbf{M}| \to |\mathbf{M}|$

$T_{a_i} : \mathbf{M}([a_1 \ldots a_n], b) \to \mathbf{M}([a_1 \ldots a_{i-1}\, Ta_i\, a_{i+1} \ldots a_n], Tb)$ for each $[a_1 \ldots a_n], a_i, b$

satisfying the following properties:

$$T_a(id_a) = id_{Ta}$$

$$T_{b^i_j}(f\circ\langle g_1,\ldots,g_n\rangle) = T_{a_i}(f)\circ\langle g_1,\ldots,g_{i-1},T_{b^i_j}(g_i),g_{i+1},\ldots,g_n\rangle$$

where $f : [a_1 \ldots a_n] \to c$ and $g_i : [b^i_1 \ldots b^i_{n_i}] \to a_i$

Definition C.2.2.
Given a strong endofunctor $T : \mathbf{M} \to \mathbf{M}$ on a representable monoidal multicategory $\mathbf{M}$ we define a strong endofunctor $\hat{t} : \hat{\mathbf{M}} \to \hat{\mathbf{M}}$ on a corresponding base category $\hat{\mathbf{M}}$ as follows:

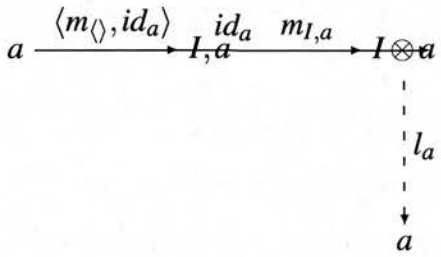
$$\hat{T}(a) = T(a)$$

$$\hat{T}(a \xrightarrow{f} b) = T_a([a] \xrightarrow{f} b)$$

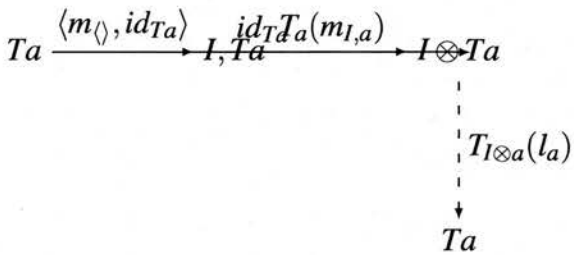
$$\begin{array}{ccc} & & a, Tb \\ & & \downarrow{\scriptstyle T_b(m_{a,b})} \\ a\otimes Tb & \xrightarrow{st_{a,Tb}} & T(a\otimes b) \end{array}$$

Lemma C.2.3. *$\hat{t} : \hat{\mathbf{M}} \to \hat{\mathbf{M}}$ is an endofunctor with strength.*

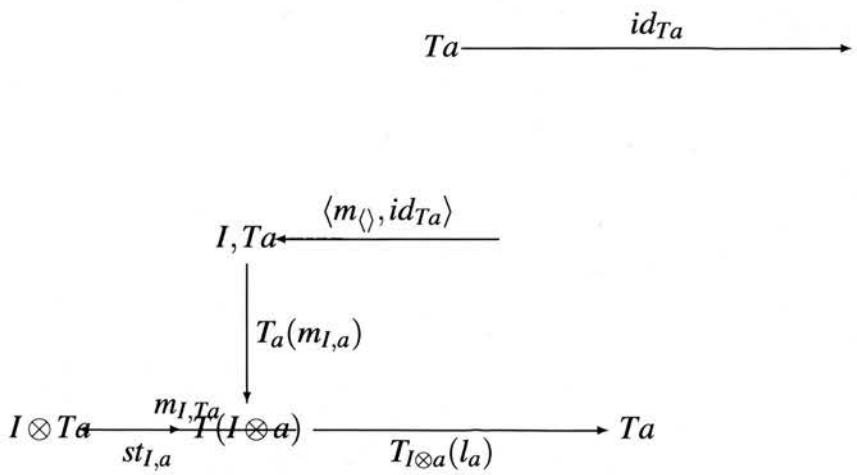
Proof. Take the definition of l_a:



Apply the functor T to the position a. Since T preserves the identity and the composition we get

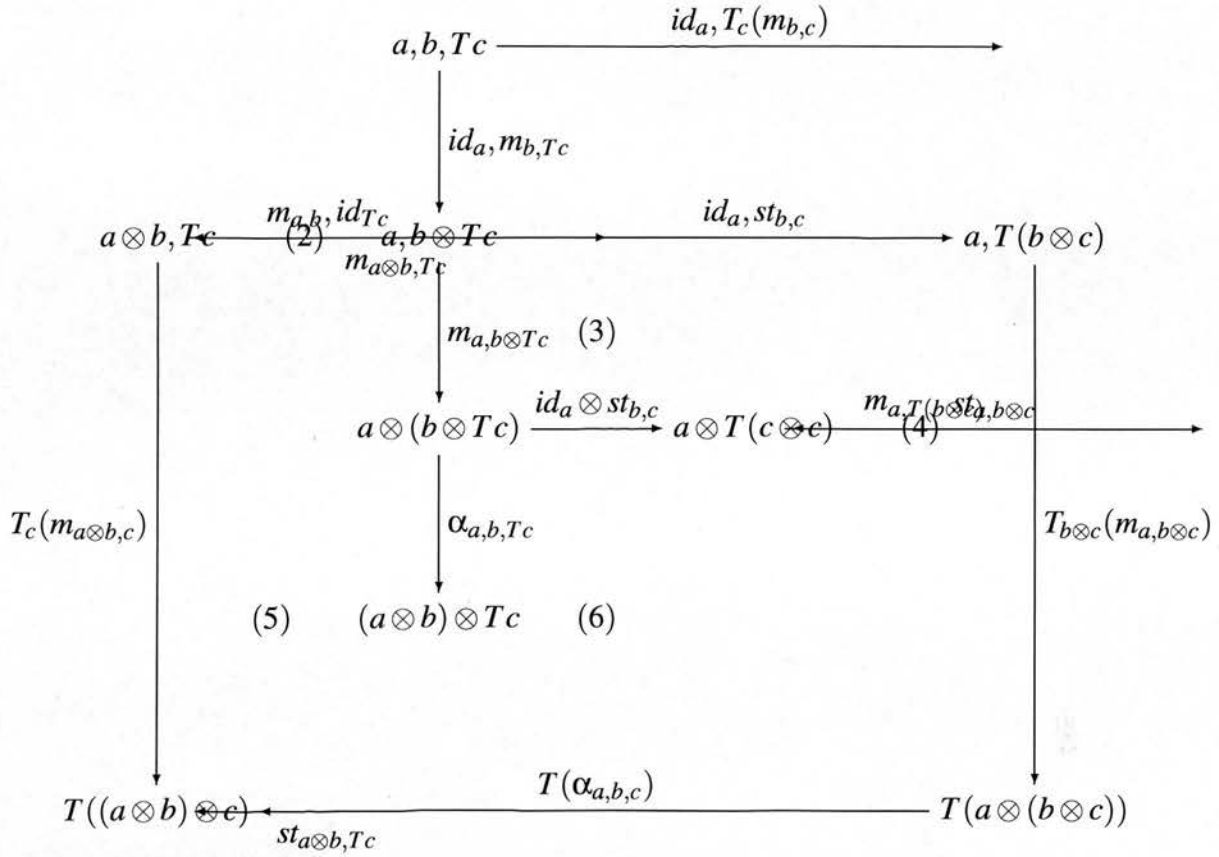


Combining this diagram with the definition of strength we have the following commuting diagram:



This proves the first property of st.To prove the second property consider the following

diagram:



We need to show that diagram (6) commutes. Diagrams (4) and (5) commute by the definition of $st_{a,b\otimes c}$ and $st_{a\otimes b,Tc}$ respectively . Diagram (2) commutes by the definition of $\alpha_{a,b,Tc}$. Diagram (3) commutes by the definition of $id_a \otimes st_{b,c}$. The outermost diagram commutes as it is the application of functor T to the commutative diagram defining $\alpha_{a,b,c}$. Notice that $m = m_{a,b\otimes Tc} \circ \langle id_a, m_{b,c}\rangle$ is universal. Diagram (6) commutes if

$$T_{b\otimes c}(m_{a,b\otimes c}) \circ \langle id_a, T_c(m_{b,c})\rangle = st_{a,b\otimes c} \circ \langle id_a \otimes st_{b,c}\rangle \circ \langle m\rangle$$
$$T_c(m_{a\otimes b,c}) \circ \langle m_{a,b}, id_c\rangle = st_{a\otimes b,Tc} \circ \langle \alpha_{a,b,Tc}\rangle \circ \langle m\rangle$$

First line holds because diagrams (1),(3) and (4) commute. Second line holds because diagrams (2) and (5) commute. □

Bibliography

[Acz80] Peter Aczel. Frege structures and the notions of proposition, truth and set. In H.J. Keisler J. Barwise and K. Kunen, editors, *The Kleene Symposium*, pages 31–59. North-Holland Publishing Company, 1980.

[Bar96a] Andrew Barber. Dual intuitionistic linear logic. *LFCS technical report ECS-LFCS-96-347, School of Informatics, Edinburgh University*, 1996.

[Bar96b] Andrew Barber. *Linear Type Theories, Semantics and Action Calculi.* PhD thesis, Edinburgh University, Laboratory for Foundations of Computer Science, 1996.

[BBHdP93a] P.N. Benton, G.M. Bierman, J.M.E. Hyland, and V.C.V. de Paiva. Linear lambda calculus and categorical models revisited. In E. Börger et al., editor, *Selected papers from Computer Science Logic'92*, volume 702 of *Lecture Notes in computer Science*, 1993.

[BBHdP93b] P.N. Benton, G.M. Bierman, J.M.E. Hyland, and V.C.V. de Paiva. A term calculus for intuitionistic linear logic. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in computer Science*, pages 75–90, 1993.

[BC92] Stephen Bellantoni and Stephen Cook. A new recursion-theoretic characterization of the polytime functions. *Computational complexity*, 2:97–110, 1992.

[Ben95] P.N. Benton. A mixed linear and non-linear logics; proofs, terms and models. In *Proceedings of Computer Science Logic '94*, volume 933 of *Lecture notes in Computer Science*, 1995.

[Bie94] Garret Bierman. *On Intuitionistic Linear Logic.* PhD thesis, University of Cambridge, Computing Laboratory, 1994.

[Cob65] A. Cobham. The intrinsic computational difficulty of functions. In Y.Bar-Hillel, editor, *Logic, Methodology and Philosophy of Science II*, pages 24–30, 1965.

[CS92] Robin Cockett and Dwight Spencer. Strong categorical datatypes I. In R.A.G. Seely, editor, *Category theory 1991*, volume 13 of *CMS Conference Proceedings*, pages 141–169, 1992.

[CS95] Robin Cockett and Dwight Spencer. Strong categorical datatypes. II. A term logic for categorical programming. *Theoret. Comput. Sci.*, 139(1-2):69–113, 1995.

[CU93] Stephen Cook and Alasdair Urquhart. Functional interpretation of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, pages 103–200, 1993.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[Hag87a] T. Hagino. *A categorical Programming Language*. PhD thesis, Edinburgh University, 1987.

[Hag87b] T. Hagino. A typed lambda-calculus with categorical type constructors. In D.E. Rydeheard D.H. Pitt, A. Poigné, editor, *Category and Computer Science*, volume 283 of *Lecture Notes in Computer Science*, pages 140–157. Springer, 1987.

[Her00] Claudio Hermida. Representable multicategories. *Advances in Mathematics*, pages 164–225, 2000.

[HM94] Joshua Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, 110(2):327–365, 1994.

[Hof97] Martin Hofmann. An application of category theory semantics to the characterisation of complexity classes using higher order function algebras. *Bulletin of Symbolic Logic*, 3(4):469–485, 1997.

[Hof99] Martin Hofmann. *Type Systems for polynomial-time computations*. Habilitationsschrift Vom Fachbereich Mathematik der Technischen Universitat Darmstadt, 1999. Also available as Edinburgh University, LFCS report ECS-LFCS-99-406.

[HPW96] James Harland, David Pym, and Michael Winikoff. Programming in lygon: an overview. In *Algebraic Methodology and Software Technology*, volume 1101 of *Lecture Notes in computer Science*, pages 391–405, 1996.

[Jac95] B. Jacobs. Parameters and parametrization in specification using distributive categories. *Fund. Informaticae*, 24(3):209–250, 1995.

[Jac01] Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and Foundations of Mathematics*. Elsevier, 2001.

[Laf88] Y. Lafont. Linear abstract machine. *Theoretical Computer Science*, 59:157–180, 1988.

[Lam68] Joachim Lambek. Deductive systems and categories i. *J. Math Systems Theory*, 2:278–318, 1968.

[Lam69] Joachim Lambek. Deductive systems and categories ii. *Springer Lecture Notes in Mathematics*, 86:76–122, 1969.

[Lam89] Joachim Lambek. Multicategories revisited. In J.W.Gray and A.Scedrov, editors, *Categories in Computer Science and Logic*, 1989.

[Lei93] Daniel Leivant. Stratified functional programs and computational complexity. In *Proceedings of 20th IEEE Symposium on Principles of Programming Languages*, 1993.

[Lei94] Daniel Leivant. Ramified recurrence and computational complexity i: Word recurrence and poly-time. In P.Clote and J. Remmek, editors, *Feasible Mathematics II*, 1994.

[Lei04] Tom Leinster. *Higher Operads, Higher Categories*. Cambridge University Press, 2004.

[LM93] Daniel Leivant and Jean-Yves Marion. Lambda calculus characterisation of polytime. *Fundamentae Informaticae*, 19:167–184, 1993.

[LS86] J. Lambek and P.J. Scott. *Introduction to higher order categorical logic*, volume 7 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1986.

[Mil94] Dale Miller. A multiple-conclusion meta-logic. In S. Abramsky, editor, *Proceedings of the 9th symposium on Logic in Computer Science*, pages 272–281, 1994.

[Ott95] James Otto. *Complexity Doctrines*. PhD thesis, Department of Mathematics and Statistics, McGill University, 1995.

[PH94] David Pym and James Harland. The uniform proof-theoretic foundation of linear logic programming. *Journal of Logic and Computation*, 4(2):175–207, 1994.

[Pit95] Andrew Pitts. Categorical logic. *Handbook of logic in computer Science*, VI, 1995.

[Rom89] Leopoldo Roman. Cartesian categories and natural numbers object. *Journal of Pure and Applied Algebra*, 58:267–278, 1989.

[RP89] Leopoldo Roman and Robert Paré. Monoidal categories and natural numbers object. *Studia Logica*, XLVIII:361–376, 1989.

[SB02] Helmut Schwichtenberg and Stephen J. Bellantoni. Feasible computation with higher types. In Helmut Schwichtenberg and Ralph Steinbrüggen, editors, *Proof and System-Reliability*, pages 399–415, 2002.

[See89] R.A.G. Seely. Linear logic, *-autonomous categories and cofree algebras. In *Conference on Categories in Computer Science and Logic*, volume 92 of *AMS Contemporary Mathematics*, pages 371–382, 1989.