Logical Semantics for Concurrent Lambda-Calculus

een wetenschappelijke proeve op het gebied van Wiskunde en Informatica

Proefschrift

ter verkrijging van de graad van doctor aan de Katholieke Universiteit Nijmegen, volgens besluit van het College van Decanen in het openbaar te verdedigen op 14 oktober 1996 des namiddags te 15.30 precies
door

M. Dezani-Ciancaglini

geboren op 22 december 1946 te Torino

July 2, 1997

Promotor:

prof.dr. H.P. Barendregt

Manuscriptcommissie:

prof.dr. J.W. Klop

dr. L. Ong

prof.dr. G. Plotkin

This research was partially supported by the ESPRIT WG 7232, Gentzen, Common Foundations of Logic and Functional Programming.

Preface

My scholastic carrier is full of gaps. I never went to elementary school and I did not obtain a certificate for it. Neither did I receive a Ph.D. degree. This, however, is quite common for Italian researchers of my generation. In fact the Ph.D. program in Italy started only in the 1980-ties.

To fill the bigger gap, I decided to submit a Ph.D. dissertation for celebrating my first fifty years.

Actually, I am responsible of the Ph.D. program in computer science in Torino. But, in spite of the fact that being a λ-woman I like self-application, I thought it was unfair for me to apply for a Ph.D. in Torino. On the other hand, I had a pleasant experience as promotor of theses in Nijmegen, so I asked for help from my friend and colleague Henk Barendregt. And the enclosed work is the outcome of the full story.

Acknowledgments

I started making a list of the persons which deeply influenced my research, in order to acknowledge all of them. When this list finally contained more than fifty persons, I did throw it away. The main reason is that my memory is not so safe anymore as when I was young, so I worried of committing unforgivable oversights. Maybe a Ph.D. at fifty years is too late!

Therefore I decided to limit my acknowledgments to the persons directly involved with this dissertation, i.e. my Promotor, Henk Barendregt, and the members of the Manuscript Committee, Jan Willem Klop, Luke Ong, and Gordon Plotkin. But I also must mention Corrado Böhm, who has been for me an incomparable guide in the world of research.

Contents

1 Introduction **1**
1.1 Summary 5

2 Classical Lambda-calculus **7**
2.1 Introduction 7
2.2 Conjunctive and Disjunctive λ-calculus 8
2.2.1 λ-calculus with Choice and Parallel Operators 9
2.2.2 The Parallel and Non-deterministic Calculus 10
2.2.3 Synchronous and Asynchronous Calculus 11
2.2.4 Relationships between the two Calculi 13
2.3 Operational Semantics 16
2.3.1 Contextual Semantics 16
2.3.2 Capabilities Semantics 17
2.4 Simple Types and Semimodels 21
2.4.1 The Type Assignment System $\mathcal{B}$ 22
2.4.2 The Set Semimodel 23
2.5 Intersection, Union Types and λ-lattices 25
2.5.1 The Set of Types and its Preorder 25
2.5.2 The Type Assignment System $\mathcal{C}$ 28
2.5.3 The λ-lattices 32
2.5.4 The Filter λ-lattice 33
2.6 Approximation Theorem and Full Abstraction 35
2.6.1 The Approximation Theorem and The Adequacy for the Contextual Semantics 35
2.6.2 Principal Pairs and Full Abstraction for the Capability Semantics 40

3 Lazy Lambda-calculus **47**
3.1 Introduction 47
3.2 The Calculus and its Operational Semantics 52
3.2.1 Partial versus Total Values 52
3.2.2 Convergency 54
3.3 A Logical Presentation 61
3.3.1 The Set of Types and its Preorder 61
3.3.2 The Type Assignment System 63
3.3.3 The Logic Congruence Relation 67
3.4 Models and Completeness 69
3.5 Full Abstraction 75

3.5.1 Characteristic Terms 76
3.5.2 Realizability 77
3.5.3 Full Abstraction Theorem 82

4 A Convex Powerdomain over Lattices **85**
4.1 Introduction 85
4.2 The Powerdomain Construction 91
4.3 The Filter Model 95
4.3.1 The Set of Types 95
4.3.2 Isomorphism between D° and $\mathcal{F}$ 102
4.4 A Calculus of Multivalued Functions 107
4.4.1 Values and Reduction 107
4.4.2 May and Must Convergence 109
4.5 The Type Assignment System 112
4.5.1 Typing Rules 112
4.5.2 Properties of Values 118
4.5.3 Convergence implies Typability 120
4.6 The Interpretation 123
4.7 Adequacy and Completeness 127
4.7.1 Characteristic Terms 127
4.7.2 Typability implies Convergence 132
4.7.3 Full Abstraction Theorem 136

5 Conclusion **139**

Chapter 1

Introduction

Powerful computer architectures make parallelism and concurrency feasible. To exploit these features in existing high-level programming languages, while retaining abstraction and logical clarity in writing programs, it is natural to extend those languages by new concepts and constructs. In particular much work has been done to accommodate parallel and concurrency primitives inside functional programming languages like Concurrent Clean [80], CML [85] and FACILE [37] (see [38] for further work in the area and for references).

This extension gives rise to the problem of introducing non-functional features in the functional framework. To illustrate this, let us consider parallelism first. If the parallel construct is a control primitive which allows the programmer to force the parallel evaluation of two or more arguments to be passed to a function, then the treatment of divergence (and the value passing mechanism) becomes much more complex. For example, a binary function may be undefined if both its arguments are undefined, without being strict neither in the first nor in the second argument. A typical example is Scott's *parallel-or* function (see [88], p. 437), the binary partial function of booleans that returns true if at least one of its arguments is defined and equal to true, and returns false if both arguments are defined and equal to false.

The *parallel-or* can be further analyzed as an example of parallel composition of compatible sequential functions (i.e. sequential functions having an upperbound). Indeed let

$$Lor \equiv \lambda xy.\ \mathbf{if}\ x\ \mathbf{then}\ \mathrm{true}\ \mathbf{else}\ y\ \mathbf{fi}, \qquad Ror \equiv \lambda xy.\ \mathbf{if}\ y\ \mathbf{then}\ \mathrm{true}\ \mathbf{else}\ x\ \mathbf{fi}$$

be the left-sequential and the right-sequential *or*, respectively. Then these functions are compatible, since, in the point-wise ordering induced by the flat domain of booleans, they have an upper bound (actually a join) which is the *parallel-or* function itself. On the other hand, if they can be computed in parallel, returning as soon as either the computation of *Lor* or the computation of *Ror* stops, then we have an implementation of the parallel-or function.

If parallel composition is a binary operator that can be applied to any pair of functions - not necessarily compatible - then the same evaluation mechanism is a non-deterministic device, that can be modeled as a multi-valued function. An example is McCarthy's *amb* function [68]. This kind of multi-valued functions have been widely considered in the literature. In the folklore this form of non-determinism is called angelic non-determinism (and credited to Hoare) because of its behavior with respect to divergence: a parallel composition is convergent if at least one of its operands converges. In terms of Dijkstra's correctness criteria, this corresponds to partial correctness.

Concurrency has been added to functional languages using CCS or CSP-like synchronization and communication primitives. In both cases the interaction with the environment introduces a different form of non-determinism, as unpredictable events may affect the behavior of the system. In particular, non-determinism comes in when a choice occurs among guarded commands having the same guard (see [55]).

This non-determinism has been modeled using internal choice operators, which are correctly considered as abstraction or specification tools. Indeed no programmer may wish to use internal choice to control the evaluation of a program; it has to be thought of instead as a declaration, saying that, whatever the actual alternative will be, the program still satisfies the correctness requirements. Of course the criterion is that of total correctness, so that, with respect to divergence, an internal choice is divergent as soon as one of its operands diverges. In folklore terms, this is demonic non-determinism (credited to Smith). One could instead use a convex powerdomain, like Plotkin's powerdomain, to model non-deterministic choice, obtaining a finer semantics. A survey about non-determinism in functional languages can be found in [91].

When facing these theoretical problems a primary point is to choose the abstraction level of the investigation. One may take a very abstract view and consider them multifunctions, or, equivalently, functions over powerdomains. This study has been pioneered by Plotkin in [82] and pursued by several authors (see [92] and, for a survey, [67]). Here continuity is the only aspect of computations which is retained in the theory, the main point being the treatment of divergence.

An alternative and quite concrete approach is to model functionality, concurrency and parallelism by syntactical tools. This amounts to design theoretical languages that formalize essentially all aspects of the computation and interaction, so that actual programming languages can be seen as sugared syntax of those. In the present case the languages and the related calculi are inspired on one hand by the λ-calculus, both typed and type free, and on the other hand by the process calculi (CCS, CSP, ACP, etc.). In exploiting the "concrete" approach, there are at least two main groups. Following the first, functions and processes are first class objects. The resulting calculus can either be seen as a λ-calculus with processes as possible arguments of functions (as in Nielson's TPL [76]) or as a process algebra with a special form of communication, generalizing the β-reduction of the λ-calculus (as in Thomsen's CHOCS [93, 48, 49]). The second group does not allow processes as arguments of functions: instead channels (or port names) have a first class status and can be sent as values (see e.g. [12]). The most radical step in this direction is to think of processes just as agents that communicate with each others channel names as values. In this way processes are virtually passed by sending the name of a (private) channel to the receiver, thus giving access to the "passed" process: this is Milner's π-calculus [73]. In the latter case, functions and functional application disappear from the calculus syntax, and they are simulated in a rather complex way.

In this thesis we advocate a third approach to the problem of the mathematical study of relevant aspects of concurrent functional languages, which, in some sense, sits in between the abstract denotational method and the concrete, direct description of interaction and communication. In this case one still considers a formal language together with its operational semantics. The latter gives an essential (and effective) description of the evaluation of expressions in the language. The main departure from the concrete approach, however, is the abstraction from communication, concentrating on a syntax which represents different kinds of non-determinism by means of different operators, whose behavior is axiomatically described by the rules of the operational semantics.

In this perspective the interaction between functionality and non-determinism has been studied both in the algebraic framework of rewriting [20, 45, 46, 2], where no abstraction operator is present, and in the λ-calculus framework, either typed [10, 11, 90], or type free [31, 79, 8, 22, 65].

A variety of non-deterministic and parallel operators have been added to the λ-calculus by several authors with different aims. One has been the study of non-determinism in the functional setting (see e.g. [10, 20, 2] and more recently [1, 79]), i.e. the study of (computable) multi-valued functions. This view is strictly connected with the theory of powerdomains introduced in [82, 92].

These efforts receive new interest in connection with recent research activities aiming at a theory of higher-order communicating processes. So it is natural to ask for a theory in which communication embodies functional application. This has been studied by Thomsen in [94] and by Boudol in [21] explicitly, while it is an implicit theme in current research on Milner's π-calculus [73].

Non-determinism and parallelism (usually represented by an interleaving operator) are fundamental concepts in process algebra theory. Combining them with λ-calculus can enlighten the theory of higher-order process algebras. Indeed an open problem with the former theory is the lack of a good denotational semantics. It is encouraging that a main step toward a definition of what is a model of a higher-order process algebra has been done by Hennessy in [48] by resorting to logical models of type-free lazy λ-calculus. On the other hand higher-order process algebras may be helpful in understanding λ-calculus theories capturing evaluation strategies, like lazy and call-by-value λ-calculi, as shown in [72, 94, 87].

Extensions of the λ-calculus with non-deterministic and/or parallel operators have been also considered in order to gain definability of combinators like Plotkin's parallel-or [84]. These extensions increase the power of the λ-calculus to detect convergence internally (easily done by call-by-value mechanisms) also in those cases in which a term converges as soon as at least one of its subterms does, no matter in which order they are evaluated. This amounts to have the definability of all compact points in a standard model, that is, by Milner's theorem, to have a fully abstract interpretation for the language.

In [22] an analysis of parallel-or in terms of an asynchronous parallel operator ($\|$) and call-by-value abstraction is proposed. Because of this asynchronicity, a term $M\|N$ can be reduced independently on both sides; to make it convergent if and only if M or N are, Boudol defines a term to be convergent if at least one of its possible computations (properly reductions) ends, what is called a *may convergence* notion. In the same paper a fully abstract, denotational semantics is provided for this calculus. This semantics is based on the Stone duality paradigm, implicitly introduced for use in denotational semantics in [89] [18]. This paradigm has been explicitly advocated in [5], where the filter model construction of [18] has been put in its right mathematical setting. A full abstraction theorem is then stated and proved.

The investigation carried out in [22] has been pursued further in the present thesis where we consider the calculi obtained by adding a parallel and a non-deterministic operator, both to the classical and to the lazy λ-calculus. Following Ong [78] we named our calculus *concurrent λ-calculus*. To gain the expected behavior, the parallel operator (always denoted by $\|$) is a synchronous operator. The non-deterministic operator (denoted by $+$) is instead an internal choice operator. By synchronicity, a term $M\|N$ is irreducible as soon as M or N is in normal form, and hence there is no need for a may convergence predicate. This choice makes explicit the different meanings of $\|$ and $+$, which are kept distinct by stipulating that a term is convergent if and only if all its reductions eventually stop, that is by using a *must convergence* criterion.

The complex operational semantics of the concurrent λ-calculus asks for an abstract treatment not involving direct reasoning on possible reducts of a given term. The approach taken in this thesis is to use type assignment systems that sufficiently expresses the operational equivalence of terms. We expect that M and N have the same types exactly when they have the same behavior in any context: this is a fully abstract "logical" semantics in the sense of [89], [18] and

[5].

To this aim, we use a system with intersection ($\wedge$) and union (denoted by $\vee$ in chapters 2 and 3, $\oplus$ in chapter 4) types, reflecting the disjunctive and conjunctive operational semantics of $\|$ and $+$. Types are viewed as properties of terms concerning their behavior with respect to the convergence predicate, and type inclusion as the logical implication. The system has a universal type ω, the property which trivially holds of everything; therefore, any type will be less than ω. As usual with type assignment systems for polymorphic λ-calculi, the arrow type expresses functionality: M has type $\sigma \to \tau$ if, for all N having type σ, MN has type τ. With respect to the order, the arrow is co-variant in the second argument and contra-variant in the first argument. Finally $\sigma \wedge \tau$ and $\sigma \vee \tau$ have a conjunctive and disjunctive meaning, respectively, while $\oplus$ models a convex powerdomain.

The lazy semantics distinguishes between functions (even the everywhere undefined function) and the undefined object, representing divergence. This means that the interpretation of the term $\lambda x.\Omega$ is better than the interpretation of Ω. These terms, instead, are equated in the theory of solvability of the classical λ-calculus [17]. On the side of types, this distinction is modeled either by making the inclusion $\omega \to \omega \leq \omega$ proper (in chapters 3 and 4) or by assuming $\omega \to \omega \sim \omega$ (in chapter 2). As a matter of fact, when considering classical λ-calculus, we take the axioms in [18] concerning the arrow. Instead, in the case of the lazy λ-calculus, we save $\sigma \to \omega \leq \omega \to \omega$, which makes $\omega \to \omega$ the type of all functions, but we reject $\omega \leq \omega \to \omega$, which would equate the interpretations of the terms Ω and $\lambda x.\Omega$ (see Corollary 3.5.6*(ii)*).

We now turn to the typing rules for non-deterministic and parallel operators. In the demonic perspective, we know that the term $M+N$ can be reduced to both M and N, so that to ensure correctness we have to prove that both M and N have the same type σ before we can conclude that $M+N$ has type σ (this is also the choice of [1]). Extending the disjunctive semantics of the parallel composition from convergence to arbitrary properties, it follows that one is entitled to type $M\|N$ with σ as soon as M or N (or both) can be typed with σ (see [22] for further explanations). This suggests the following typing rules

$$\frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M + N : \sigma} \qquad \frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M\|N : \sigma} \qquad \frac{\Gamma \vdash N : \sigma}{\Gamma \vdash M\|N : \sigma}\ .$$

This is the choice done in chapter 2.
The inclusion relation $\leq$ among types makes $\wedge$ into the meet and $\vee$ into the join, and we have both a subtyping and an intersection rule, namely

$$\frac{\Gamma \vdash M : \sigma \quad \sigma \leq \tau}{\Gamma \vdash M : \tau} \qquad \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \wedge \tau}.$$

Therefore the rules for $+$ and $\|$ above are equivalent to

$$\frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau}{\Gamma \vdash M + N : \sigma \vee \tau} \qquad \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau}{\Gamma \vdash M\|N : \sigma \wedge \tau}.$$

These are the typing rules of chapter 3.
In chapter 4 (where union is denoted by $\oplus$) we have the same rules, also if the subtyping relation distinguishes between $\vee$ and $\oplus$.

Each type assignment system implicitly suggests a notion of interpretation in which each term can be seen as denoting the set of types it can be assigned. Then one can think of extending the notion of filter models such that they encompass the present calculus and union types. Filter models were introduced in [18] for the classical λ-calculus and based on the intersection type discipline.

In that case, however, discovering that filters of types do actually form a structure (a λ-model) was based on the pre-existing and independent definition of this kind of mathematical structures (see [54, 70]). Here the problem is the opposite: given the logical interpretation induced by our system, we look for a reasonable definition of what is a model of our calculus.

In the extended view of Curry types (see [18, 24]), i.e. when the interpretation of a term is the set of types which can be deduced for it, type theories are an instance of information systems (see [89, 29]). Taking filters of types we have a domain that, seen topologically, is the Stone space generated by the theory of type inclusion (see [59, 5]).

We do carry out the details of the isomorphism between the filter model and the initial solution of the given domain equation only for the convex powerdomain in chapter 4. For the two filter models presented in chapters 2 and 3 we refer to [7]. Instead, we analyze compositionally the interpretation of terms defined by $[\![M]\!] = \{\sigma \mid \vdash M : \sigma\}$ (where M is closed), and devise a category of objects that embodies the minimum needed structure to interpret the calculus.

The content of the present thesis is essentially that of [33], [34] and [8].

1.1. Summary

In Chapter 2 we consider a λ-calculus enriched with both a parallel operator and a non-deterministic choice operator. The notion of reduction extends the classical β-reduction, taking into account the disjunctive character of parallelism and the conjuctive character of non-determinism. An operational semantics based on a generalized notion of solvability naturally arises. A "filter" model of this language is built; a type assignment system with intersection and union types provides a logical description of this model. This filter model turns out to be adequate with respect to the operational semantics; a main tool used in the proof of this result is an approximation theorem, which states that a type can be deduced for a term iff it can be deduced for an approximant of this term.

A natural development is to add angelic parallelism and demonic non-determinism to the lazy λ-calculus. This is the argument of Chapter 3. In the lazy perspective it is interesting to allow both call-by-name and call-by-value variables. The reduction of a call-by-value abstraction requires a refinement of the notion of value, here obtained by distinguishing between partial and total values. A minor modification of the type assignment system, which was introduced in the classical case, provides a logical description of a "filter" model for the present language. In this new "filter" model the union type constructor is crucial to distinguish between call-by-value and call-by-name variables. The main result is the full abstraction of the "filter" model; a crucial step in this proof is the definition of "characteristic terms", which finely reflect the duality between conjunctive and disjunctive operators.

One could be interested in interpreting the non-deterministic choice operator using a partial order relation finer than the demonic one, requiring in particular that this relation behaves like the Egli-Milner one as far as convergence is concerned. All this without loosing the mutual distributivity between the parallel operator and the non-deterministic choice one. This leads to a new powerdomain construction (done in Chapter 4) in the category of complete algebraic lattices (with continuous functions as morphisms) which meets the above requirements. Also in this case the construction is done by means of a suitable type assignment system with intersection and union types. Each one of the following chapters can be read independently, since they are self-contained. Only for some proofs and comparisons they refer to previous chapters.

Chapter 2

Classical Lambda-calculus

2.1. Introduction

The present chapter aims to study the *full* classical λ-calculus extended with $\|$ and $+$. This essentially amounts to allow reduction under abstraction and evaluation of the argument even before passing it.

Since the original paper [6] by Abramsky and Ong, it has been argued that the lazy λ-calculus is a better model of actual implementations of functional programming languages like Scheme. Indeed these languages do not evaluate the bodies of functions before formal parameters have been replaced by the arguments to which functions are applied. Similarly they do not evaluate the arguments before passing them.

There is, however, a missing point in treating functional languages in a lazy perspective. In that setting we are forced to look at functions in a merely extensional way, that is as black boxes whose different behaviors can be detected just testing them against application to suitable arguments and waiting for the output (but also, possibly, waiting forever). As a matter of fact, the semantics of the lazy λ-calculus has been defined in [6] by introducing the notion of functional bisimulation, which is nothing but a sophisticated version of the extensional idea.

The unfolding semantics (sometimes called algebraic semantics) is a well established theory of recursive languages, originated with Tarsky's fixed point theorem and with Kleene's first recursion theorem. This theory has its λ-calculus counterpart in the notion of Böhm tree, which finely recovers topological ideas from the syntactical notions of head normal forms and separability (see [17]). Now it seems that such a theory does not exist in the case of lazy λ-calculus. As a matter of fact, the problem cannot be remedied by resorting to Lévy-Longo trees, since they induce a finer semantics than functional bisimulation (this has been shown in Ong's thesis [77]). This justifies our choice of considering the classical λ-calculus.

In the present chapter we give a semantics based on the notion of unfolding for our parallel and non-deterministic extension of classical λ-calculus. This is not achieved by means of trees, but by using the equivalent notion of approximant originated, in the case of λ-calculus, from the works of Lévy [63] and Wadsworth [95].

In the first section of the chapter we introduce the syntax of the calculus and two reduction relations. The first one explicitly makes the $+$ into a choice operator, while the second one, instead, simulates the choice by a distribution law. Adapting to the present case the notion of head reduction and head normal form, we prove that both reductions define the same set of solvable terms, so that in the following we study the second reduction relation which is technically easier to handle.

After a short discussion of the contextual theory induced by the set of solvable terms, we define the concept of approximant and the connected notion of capability (reminiscent of the homonymous notion in [78]), formally setting the unfolding semantics that we study.

In the subsequent two sections we introduce a type assignment system in two steps. The first one considers just Curry types, simply adding to the assignment system the rules for typing $M \| N$ and $M + N$. As a preliminary result we get Plotkin's set semimodel [83] for our calculus and the equational theory on terms which it induces. We then enrich the type syntax with intersection types, union types, and the universal type ω. Types are partially ordered so that they give rise, by the usual filter construction, to a distributive lattice which, as a domain, is an ω-algebraic prime lattice. We refer to [7] for more details and for the description of the domain equation underlying the construction, which involves both lower and upper powerdomain functors, combined with the space of Scott-continuous functions. By adding a subtyping rule and an intersection introduction rule, the type assignment system turns out to be sound and complete with respect to λ-lattices, which are λ-models with a lattice structure.

The last section contains the main results of the chapter, namely the approximation theorem and the full abstraction theorem. Roughly speaking, the approximation theorem says that the set of types of any term is the union of all types that can be given to its approximations, hence being the limit of them in the logical semantics. The full abstraction theorem states that the unfolding semantics and the logical semantics have actually the same theory. Moreover, we get that solvable terms are characterized as those terms which are typable by a type which is not equivalent to ω.
The content of this chapter is essentially [33].

2.2. Conjunctive and Disjunctive λ-calculus

In this section we give the syntax of our calculus and prove the basic properties of two reduction relations. The general theme is that of distinguishing between non-determinism and parallelism.

It is certainly debatable whether these two notions have to be kept distinct, since in many cases parallelism is explained in terms of non-determinism. This is true in particular when the aim of parallelism is the possibility of handling simultaneously several different computations and of terminating as soon as one of these computations terminates.

But if we implement this device using a choice operator, then we must assume the existence of an oracle which, at each stage, will suggest the right decision. In this way the oracle will prevent any non terminating computation, whenever at least one output of the non-deterministic program exists. This is no more necessary if, instead, we use an operator which does not make choices, but which evaluates in a synchronous way its arguments. I.e. an operator which does one reduction step only when both its arguments are reducible, and which stops otherwise.

On the contrary the choice operator comes out as a tool for representing programs whose behavior can be determined, at a certain time, by unpredictable events. In this case the choice has no guidance. Therefore the criterion of taking into account all possible cases when studying the convergence of the program (that is the total correctness criterion) is the most natural one.

We will analyze the distinction between the internal choice operator and the parallel synchronous operator using the logical distinction between disjunction and conjunction in section 2.5.

2.2.1. λ-calculus with Choice and Parallel Operators

Let $\Lambda_{+\|}$ be the set of pure λ-terms enriched with the binary operators $+$ and $\|$, that is the set of expressions generated by the following grammar:

$$M ::= x \mid \lambda x.M \mid MM \mid M+M \mid M\|M$$

where x ranges over a denumerable set *Var* of variables. As usual, $FV(M)$ is the set of variables which occur free in M. To simplify notation we assume that abstraction and application take precedence over $+$ and $\|$.

As usual, if $\longrightarrow_R$ is a one-step reduction relation on $\Lambda_{+\|}$, then $\xrightarrow{*}_R$ and $=_R$ denote the transitive and reflexive, the transitive and reflexive and symmetric closure of $\longrightarrow_R$, respectively. Finally $\xrightarrow{n}_R$ means the n-times self-composition of $\longrightarrow_R$.

To extend the β-reduction relation $\longrightarrow_\beta$ of classical λ-calculus to $\Lambda_{+\|}$, we explicitly mention rules (μ), (ν) and (ξ), instead of considering the closure under contexts of the β-rule. Therefore we implicitly forbid reductions of the form:

$$M \longrightarrow N \Rightarrow op(\ldots, M, \ldots) \longrightarrow op(\ldots, N, \ldots)$$

where *op* is either $+$ or $\|$.

We also define explicitly the sub-relation of $\longrightarrow_\beta$ called in the literature *head reduction* (see [17] also for the subsequent notion of solvable terms).

2.2.1. Definition.

(i) The relation $\longrightarrow_\beta$ is the least binary relation on $\Lambda_{+\|}$ defined by:

$$\begin{array}{llll} (\beta) & (\lambda x.M)N \longrightarrow_\beta M[N/x] & (\mu) & M \longrightarrow_\beta N \Rightarrow LM \longrightarrow_\beta LN \\ (\nu) & M \longrightarrow_\beta N \Rightarrow ML \longrightarrow_\beta NL & (\xi) & M \longrightarrow_\beta N \Rightarrow \lambda x.M \longrightarrow_\beta \lambda x.N. \end{array}$$

(ii) The relation $\longrightarrow^h_\beta$ is the least binary relation on $\Lambda_{+\|}$ satisfying (β) and (ξ) above and

$$(\nu_\beta) \quad M \longrightarrow^h_\beta M' \text{ and } M \notin \mathsf{Abst} \Rightarrow MN \longrightarrow^h_\beta M'N$$

where $\mathsf{Abst} = \{\lambda x.P \mid P \in \Lambda_{+\|}, x \in \mathit{Var}\}$.

In the solvability theory of the classical of λ-calculus, meaningful terms are not just those possessing a normal form with respect to $\longrightarrow_\beta$, but more in general those which determine a terminating $\longrightarrow_\beta$ reduction, when applied to suitable terms. These are characterized as those terms having a normal form with respect to the $\longrightarrow^h_\beta$ relation (see [17] Theorem 8.3.14). This normal form is called *head normal form*, and, in view of the characterization just mentioned, terms possessing a head normal form are called *solvable*.

2.2.2. Definition. The subset of $\Lambda_{+\|}$

$$\mathsf{SOL}_\beta = \{M \mid \exists M'.\ M \xrightarrow{*}{}^h_\beta M' \text{ and } \neg\, \exists N.\ M' \longrightarrow^h_\beta N\}$$

is the set of β-solvable terms.

Note that $\longrightarrow^h_\beta$-reduction is deterministic since any term has at most one head redex because of rule (ν_β). Hence we have immediately:

$$M \in \mathsf{SOL}_\beta \quad \Leftrightarrow \quad \exists n\, \forall m \geq n\, \neg\, \exists N.\ M \xrightarrow{m}{}^h_\beta N.$$

2.2.2. The Parallel and Non-deterministic Calculus

In this subsection we think of $+$ as an *internal choice* operator and of $\|$ as a synchronous *parallel evaluator* of its arguments. Indeed, rule $(+_c)$ allows to freely choose between the arguments of $+$. Instead, $M\|N$ reduces according to rule $(\|_s)$ only when both M and N reduce. Moreover, since every term represents a function in the λ-calculus, we further define $M\|N$ as the function which, when applied to some L, returns $ML\|NL$ (rule $(\|_{app})$). All this is formalized in the following definition.

2.2.3. Definition.

(i) The relation $\longrightarrow_{pn}$ (*Parallel* and *Non-deterministic* reduction) is the least binary relation on $\Lambda_{+\|}$ satisfying (β), (μ), (ν), (ξ) and

$$\begin{array}{ll} (+_c) & M+N \longrightarrow_{pn} M, \quad M+N \longrightarrow_{pn} N \\ (\|_s) & M \longrightarrow_{pn} M',\ N \longrightarrow_{pn} N' \Rightarrow M\|N \longrightarrow_{pn} M'\|N' \\ (\|_{app}) & (M\|N)L \longrightarrow_{pn} ML\|NL. \end{array}$$

(ii) The relation $\longrightarrow^h_{pn}$ (*Parallel* and *Non-deterministic* head reduction) is the least binary relation on $\Lambda_{+\|}$ satisfying (β), (ξ), $(+_c)$, $(\|_s)$, $(\|_{app})$ and

$$(\nu_{pn}) \quad M \longrightarrow^h_{pn} M' \text{ and } M \notin \mathsf{Abst} \cup \mathsf{Par} \Rightarrow MN \longrightarrow^h_{pn} M'N$$

where $\mathsf{Par} = \{P\|Q \mid P, Q \in \Lambda_{+\|}\}$.

Because of rule $(+_c)$, the relation $\longrightarrow_{pn}$ is not confluent. Moreover, because of rule $(\|_s)$, the set of "head redexes" of a term M (that is the set of redexes that will be contracted in the first step of a $\longrightarrow^h_{pn}$ reduction) can be larger than a singleton. These facts imply that a term M may have more than one immediate reduct with respect to $\longrightarrow^h_{pn}$ (but always finitely many).

Consequently there are at least two natural ways of extending the notion of solvability to $\longrightarrow_{pn}$. We could say that M is solvable if at least one $\longrightarrow^h_{pn}$ reduction starting from M ends in a (head) normal form. This definition, however, does not distinguish between $+$ and $\|$ by the property of being solvable. Indeed, both $M+N$ and $M\|N$ would be solvable if and only if either M or N is solvable.

Since we are looking for a semantics keeping distinct $+$ and $\|$ wrt convergence, we define M to be solvable if and only if *all* head reductions starting from it terminate. We immediately have that $M+N$ is solvable if and only if both M *and* N are, while $M\|N$ is solvable if and only if *either* M is solvable *or* N is solvable.

As observed above, the reduction tree of any term under the relation $\longrightarrow^h_{pn}$ is a finitary tree, hence by König's Lemma, it is finite if and only if all its branches have finite lengths, i.e. there is an upper bound to the length of all head reductions of the given term. We use this in the following definition.

2.2.4. Definition. The subset of $\Lambda_{+\|}$

$$\mathsf{SOL}_{pn} = \{M \mid \exists n\ \forall m \geq n \neg \exists N.\ M \xrightarrow{m}{}^h_{pn} N\}$$

is the set of pn-solvable terms.

As observed above, this definition of solvability fits well with the conjunctive behavior of + and the disjunctive behavior of $\|$ since

$$M + N \in \mathsf{SOL}_{pn} \;\Leftrightarrow\; M \in \mathsf{SOL}_{pn} \text{ and } N \in \mathsf{SOL}_{pn}$$

while

$$M\|N \in \mathsf{SOL}_{pn} \;\Leftrightarrow\; M \in \mathsf{SOL}_{pn} \text{ or } N \in \mathsf{SOL}_{pn}.$$

For example, if $\mathbf{I} \equiv \lambda x.x$, and $\Delta \equiv \lambda x.xx$, we have that $\mathbf{I} + \Delta\Delta$ is pn-unsolvable, since $\mathbf{I} + \Delta\Delta \longrightarrow^h_{pn} \Delta\Delta \longrightarrow^h_{pn} \Delta\Delta$. Instead $\mathbf{I}\|\Delta\Delta$ is a normal form, so a fortiori it is pn-solvable. Now $\lambda x.(x\mathbf{I} + x(\Delta\Delta))$ is a pn-solvable term, since it head reduces to $\lambda x.x\mathbf{I}$ and to $\lambda x.x(\Delta\Delta)$. Notice that $\lambda x.x(\Delta\Delta)$ reduces to itself under $\longrightarrow_{pn}$, but it is a head normal form.

2.2.3. Synchronous and Asynchronous Calculus

We introduce a slightly different reduction relation, still extending β-reduction and still ascribing a conjunctive semantics to + and a disjunctive one to $\|$. The aim is that of eliminating rule $(+_c)$. The advantage will be that the existence of a finite reduction path out of a term assures the solvability of the term (see Corollary 2.2.9). In this reduction + is an *asynchronous* evaluator of its operands, while $\|$ is a *synchronous* one. Moreover, both + and $\|$ have the feature of passing to their operands any argument to which they apply.

2.2.5. Definition.

(i) The relation $\longrightarrow_{sa}$ (*Synchronous* and *Asynchronous* reduction) is the least binary relation on $\Lambda_{+\|}$ satisfying $(\beta), (\mu), (\nu), (\xi), (\|_s), (\|_{app})$ and

$$(+_a) \qquad M \longrightarrow_{sa} M' \Rightarrow \begin{cases} M + N \longrightarrow_{sa} M' + N \\ N + M \longrightarrow_{sa} N + M' \end{cases}$$

$$(+_{app}) \quad (M + N)L \longrightarrow_{sa} ML + NL.$$

(ii) The relation $\longrightarrow^h_{sa}$ (*Synchronous* and *Asynchronous* head reduction) is the least binary relation on $\Lambda_{+\|}$ satisfying (β), (ξ), $(+_a)$, $(+_{app})$, $(\|_s)$, $(\|_{app})$ and

$$(\nu_{sa}) \quad M \longrightarrow^h_{sa} M' \text{ and } M \notin \mathsf{Abst} \cup \mathsf{Par} \cup \mathsf{Sum} \Rightarrow MN \longrightarrow^h_{sa} M'N$$

where $\mathsf{Sum} = \{P + Q \mid P, Q \in \Lambda_{+\|}\}$.

Notice that $\longrightarrow_{sa}$ differs from $\longrightarrow_{pn}$, since for example, writing $\mathbf{I} \equiv \lambda x.x$, if $P \longrightarrow_{sa} P'$ and $Q \longrightarrow_{sa} Q'$, then $(P + Q)\|\mathbf{II}$ sa-reduces to $(P' + Q)\|\mathbf{I}$ and to $(P + Q')\|\mathbf{I}$. Instead $(P + Q)\|\mathbf{II}$ pn-reduces to $P\|\mathbf{I}$ and to $Q\|\mathbf{I}$. This example shows also that, even if rule $(+_c)$ has been dropped, the presence of rule $(+_a)$, together with the synchronous character of $\|$, implies that $\longrightarrow_{sa}$ is not Church-Rosser. In fact both $(P' + Q)\|\mathbf{I}$ and $(P + Q')\|\mathbf{I}$ are normal forms, since the reducibility of a parallel composition requires reducibility of both its operands.

For the same reason the head reduction $\longrightarrow^h_{sa}$ is non-deterministic. Consequently, we define the notion of sa-solvability in the same way as we did for pn-solvability.

2.2.6. Definition. The subset of $\Lambda_{+\|}$

$$\mathsf{SOL}_{sa} = \{M \mid \exists n\, \forall m \geq n \neg \exists N.\, M \xrightarrow{m}{}^h_{sa} N\}$$

is the set of sa-solvable terms.

The difference between $+$ and $\|$ with respect to the solvability criterion is still expressed as follows
$$M + N \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; M \in \mathsf{SOL}_{sa} \text{ and } N \in \mathsf{SOL}_{sa}$$
while
$$M \| N \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; M \in \mathsf{SOL}_{sa} \text{ or } N \in \mathsf{SOL}_{sa}.$$

In spite of the lack of the Church-Rosser property, the existence of a finite $\longrightarrow^h_{sa}$-reduction path now implies the finiteness of all $\longrightarrow^h_{sa}$-reduction paths. To prove this we need to prove a more general statement, since a stronger induction hypothesis is used when dealing with rules (ξ) and (ν_{sa}). In particular, (ν_{sa}) forces us to consider term vectors and consequently rule (ξ) forces us to consider substitutions (see Proposition 2.2.8).

The following properties of the reduction relation $\longrightarrow^h_{sa}$ are crucial in subsequent proofs. They are an immediate consequence of the constraint in rule (ν_{sa}).

2.2.7. Proposition.

(i) *If $P \in \mathsf{Abst} \cup \mathsf{Par} \cup \mathsf{Sum}$, then any head reduction out of $PL_0\vec{L}$ will start by reducing the subterm PL_0.*

(ii) *If $P \equiv P_1$ op P_2 (where op is $+$ or $\|$) then any exhaustive head reduction of $PL_0 \cdots L_{k-1}$ will start with k steps leading to $P_1L_0 \cdots L_{k-1}$ op $P_2L_0 \cdots L_{k-1}$.*

As usual a substitution is a map from variables to terms which is the identity for all variables but a finite set.

2.2.8. Proposition. *If $M \longrightarrow^h_{sa} N$, then*
$$\forall (\cdot)^\nabla, \vec{L}.\ N^\nabla \vec{L} \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; M^\nabla \vec{L} \in \mathsf{SOL}_{sa},$$
where $(\cdot)^\nabla$ ranges over substitutions and $\vec{L}$ is a vector of terms.

Proof. By induction on the definition of $\longrightarrow^h_{sa}$.

Case $(+_a)$: then $M \equiv P + Q \longrightarrow^h_{sa} P' + Q \equiv N$ with $P \longrightarrow^h_{sa} P'$. Now
$$\begin{array}{rcll}
(P'^\nabla + Q^\nabla)\vec{L} \in \mathsf{SOL}_{sa} & \Leftrightarrow & P'^\nabla \vec{L} + Q^\nabla \vec{L} \in \mathsf{SOL}_{sa} & \\
& \Leftrightarrow & P'^\nabla \vec{L} \in \mathsf{SOL}_{sa} \text{ and } Q^\nabla \vec{L} \in \mathsf{SOL}_{sa} & \\
& \Leftrightarrow & P^\nabla \vec{L} \in \mathsf{SOL}_{sa} \text{ and } Q^\nabla \vec{L} \in \mathsf{SOL}_{sa} & \text{by induction} \\
& \Leftrightarrow & P^\nabla \vec{L} + Q^\nabla \vec{L} \in \mathsf{SOL}_{sa} & \\
& \Leftrightarrow & (P^\nabla + Q^\nabla)\vec{L} \in \mathsf{SOL}_{sa} &
\end{array}$$

where the $\Leftarrow$ part of the first implication and the last $\Leftrightarrow$ are trivial if the vector $\vec{L}$ is empty. Otherwise they readily follows from 2.2.7*(i)*.

Case $(+_{app})$: then $M \equiv (P+Q)R \longrightarrow^h_{sa} PR + QR \equiv N$. We have:
$$(P^\nabla R^\nabla + Q^\nabla R^\nabla)\vec{L} \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; (P^\nabla + Q^\nabla)R^\nabla \vec{L} \in \mathsf{SOL}_{sa}$$

as in previous case.

Case $(\|_s)$: then $M \equiv P\|Q \longrightarrow^h_{sa} P'\|Q' \equiv N$ with $P \longrightarrow^h_{sa} P'$ and $Q \longrightarrow^h_{sa} Q'$. Then this case is similar to case $(+_a)$, where $+$ is replaced by $\|$ and "$\ldots \in \mathsf{SOL}_{sa}$ and $\ldots \in \mathsf{SOL}_{sa}$" is replaced by "$\ldots \in \mathsf{SOL}_{sa}$ or $\ldots \in \mathsf{SOL}_{sa}$".

Case $(\|_{app})$: same as case $(+_{app})$ where $+$ is replaced by $\|$.

Case (β): then $M \equiv (\lambda x.P)Q \longrightarrow^h_{sa} P[Q/x] \equiv N$. By 2.2.7*(i)*, the first step out of $(\lambda x.P^\nabla)Q^\nabla\vec{L}$ must be a β-reduction. Then

$$M^\nabla\vec{L} \equiv (\lambda x.P^\nabla)Q^\nabla\vec{L} \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; N^\nabla\vec{L} \equiv P^\nabla[Q^\nabla/x]\vec{L} \in \mathsf{SOL}_{sa}.$$

Note that, being x bound in $\lambda x.P$, we can freely assume that the substitution $(\cdot)^\nabla$ does not affect it.

Case (ξ): then $M \equiv \lambda x.P \longrightarrow^h_{sa} \lambda x.P' \equiv N$ with $P \longrightarrow^h_{sa} P'$. If the vector $\vec{L}$ is empty, then the thesis follows from the induction hypothesis. Otherwise, taking the non empty vector $L_0\vec{L}$, the first step out of $(\lambda x.P'^\nabla)L_0\vec{L}$ will be a β-reduction by 2.2.7*(i)*. Then:

$$\begin{array}{rcll} (\lambda x.P'^\nabla)L_0\vec{L} \in \mathsf{SOL}_{sa} & \Leftrightarrow & (P'^\nabla[L_0/x])\vec{L} \in \mathsf{SOL}_{sa} & \\ & \Leftrightarrow & (P^\nabla[L_0/x])\vec{L} \in \mathsf{SOL}_{sa} & \text{by induction} \\ & \Leftrightarrow & (\lambda x.P^\nabla)L_0\vec{L} \in \mathsf{SOL}_{sa}, & \end{array}$$

where in the induction hypothesis the substitution is the composition of $(\cdot)^\nabla$ and $[L_0/x]$. As in case (β) we assume that $(\cdot)^\nabla$ does not substitute for x.

Case (ν_{sa}): then $M \equiv PQ \longrightarrow^h_{sa} P'Q \equiv N$ with $P \longrightarrow^h_{sa} P'$. Then, by the induction hypothesis, taking the vector $Q^\nabla\vec{L}$, we immediately have that

$$N^\nabla\vec{L} \equiv P'^\nabla Q^\nabla\vec{L} \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; P^\nabla Q^\nabla\vec{L} \equiv M^\nabla\vec{L} \in \mathsf{SOL}_{sa}. \qquad \Box$$

2.2.9. Corollary. *$M \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; \exists M'.\ M \xrightarrow{*}{}^h_{sa} M'$ and $\neg\,\exists N.\ M' \longrightarrow^h_{sa} N$.*

Proof. $\Rightarrow$ is trivial.
The proof of $\Leftarrow$ follows by straightforward induction on the length of the reduction $M \xrightarrow{*}{}^h_{sa} M'$ using Proposition 2.2.8 with the identical substitution and the empty vector. $\Box$

2.2.4. Relationships between the two Calculi

Even if the reductions $\longrightarrow^h_{pn}$ and $\longrightarrow^h_{sa}$ are different, as it is clear also from Corollary 2.2.9, since it does not hold for $\longrightarrow^h_{pn}$, they are equivalent in the sense that they determine the same set of solvable terms, i.e. SOL_{pn} and SOL_{sa} coincide.

To show this we need a definition and some Lemmas, all proved by induction on the structure of one-step head reductions.

2.2.10. Definition. Define SOL^n_{sa} as the set of terms whose longest $\longrightarrow^h_{sa}$ reduction has at most n steps, i.e.:

$$\mathsf{SOL}^n_{sa} = \{M \mid \forall m \geq n\ \neg\,\exists N.\ M \xrightarrow{m}{}^h_{sa} N\}.$$

Comparing this with Definition 2.2.6 it is clear that $\mathsf{SOL}_{sa} = \bigcup_{n\geq 0} \mathsf{SOL}^n_{sa}$.

The first lemma connects the reduction $\longrightarrow^h_{pn}$ with the sets SOL^n_{sa}.

2.2.11. LEMMA. *If $M \longrightarrow^h_{pn} N$ then, for all $\vec{L}$ and substitutions $(\cdot)^\nabla$:*

$$M^\nabla \vec{L} \in \mathsf{SOL}^n_{sa} \Rightarrow \exists m \leq n.\ N^\nabla \vec{L} \in \mathsf{SOL}^m_{sa}.$$

Moreover, if $N^\nabla \vec{L} \notin \mathsf{SOL}^{n-1}_{sa}$, then we used rule $(+_c)$ in deriving that $M \longrightarrow^h_{pn} N$.

PROOF. By induction on $\longrightarrow^h_{pn}$.

Case $(+_c)$: then $M \equiv P + Q \longrightarrow^h_{pn} P \equiv N$, say.
If $(P+Q)^\nabla \vec{L} \equiv (P^\nabla + Q^\nabla)\vec{L} \in \mathsf{SOL}^n_{sa}$ and r is the length of $\vec{L}$, then by 2.2.7*(ii)* any $\longrightarrow^h_{sa}$ reduction out of $(P^\nabla + Q^\nabla)\vec{L}$ will produce $P^\nabla \vec{L} + Q^\nabla \vec{L}$ in r steps. Hence $P^\nabla \vec{L} + Q^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa}$ and, a fortiori, $P^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa}$. If $\vec{L}$ is empty, we get $P^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$.

Case $(\|_s)$: then $M \equiv P\|Q \longrightarrow^h_{pn} P'\|Q' \equiv N$ with $P \longrightarrow^h_{pn} P'$ and $Q \longrightarrow^h_{pn} Q'$.
Now if r is the length of $\vec{L}$, then

$$\begin{array}{rcl} M^\nabla \vec{L} \equiv (P^\nabla \| Q^\nabla)\vec{L} \in \mathsf{SOL}^n_{sa} & \Rightarrow & P^\nabla \vec{L} \| Q^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa} \\ & \Rightarrow & P^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa} \text{ or } Q^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa} \\ & \Rightarrow & \exists m \leq n-r.\ P'^\nabla \vec{L} \in \mathsf{SOL}^m_{sa} \text{ or } Q'^\nabla \vec{L} \in \mathsf{SOL}^m_{sa} \\ & & \hfill \text{by induction} \\ & \Rightarrow & \exists m \leq n-r.\ (P'^\nabla \| Q'^\nabla)\vec{L} \equiv N^\nabla \vec{L} \in \mathsf{SOL}^{m+r}_{sa} \end{array}$$

and clearly $m + r \leq n$.
Notice that if $\vec{L}$ is empty, we can have $m = n$. In this case $P^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$ or $Q^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$. So we have by induction that we used rule $(+_c)$ in deriving $P \longrightarrow^h_{pn} P'$ or $Q \longrightarrow^h_{pn} Q'$. Therefore rule $(+_c)$ has also been used in deriving $M \longrightarrow^h_{pn} N$.

Case $(\|_{app})$: then $M \equiv (P\|Q)R \longrightarrow^h_{pn} PR\|QR \equiv N$.
If $((P\|Q)R)^\nabla \vec{L} \equiv (P^\nabla \| Q^\nabla)R^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$, then we immediately have

$$(P^\nabla R^\nabla \| Q^\nabla R^\nabla)\vec{L} \equiv (PR\|QR)^\nabla \vec{L} \in \mathsf{SOL}^{n-1}_{sa}\ .$$

Case (β): then $M \equiv (\lambda x.P)Q \longrightarrow^h_{pn} P[Q/x] \equiv N$.
Now for all $(\cdot)^\nabla$ $((\lambda x.P)Q)^\nabla \equiv (\lambda x.P^\nabla)Q^\nabla$, up to renaming of the bound variable x. For all $\vec{L}$, any $\longrightarrow^h_{sa}$ reduction out of $(\lambda x.P^\nabla)Q^\nabla \vec{L}$ will start by

$$(\lambda x.P^\nabla)Q^\nabla \vec{L} \longrightarrow^h_{sa} P^\nabla[Q^\nabla/x]\vec{L}$$

hence, if $(\lambda x.P^\nabla)Q^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$, then $P^\nabla[Q^\nabla/x]\vec{L} \in \mathsf{SOL}^{n-1}_{sa}$.

Case (ξ): then $M \equiv \lambda x.P \longrightarrow^h_{pn} \lambda x.P' \equiv N$, with $P \longrightarrow^h_{pn} P'$.
Now, up to renaming of the bound variable x, $(\lambda x.P)^\nabla \equiv \lambda x.P^\nabla$. Assume that $(\lambda x.P^\nabla)\vec{L} \in \mathsf{SOL}^n_{sa}$, then if $\vec{L}$ is empty the thesis follows immediately by induction. Otherwise the first step of any $\longrightarrow^h_{sa}$ will be

$$(\lambda x.P^\nabla)Q\vec{L} \longrightarrow^h_{sa} P^\nabla[Q/x]\vec{L},$$

so that $P^\nabla[Q/x]\vec{L} \in \mathsf{SOL}^{n-1}_{sa}$. From the induction hypothesis there exists $m \leq n-1$ such that

$$P'^\nabla[Q/x]\vec{L} \in \mathsf{SOL}^m_{sa}$$

which implies that

$$(\lambda x.P'^\nabla)Q\vec{L} \in \mathsf{SOL}^{m+1}_{sa}$$

and clearly $m + 1 \leq n$.

Case (ν_{pn}): then $M \equiv PQ \longrightarrow^h_{pn} P'Q \equiv N$ with $P \longrightarrow^h_{pn} P'$, where $P \notin \mathsf{Abst} \cup \mathsf{Par}$.
Now, if $(PQ)^\nabla \vec{L} \equiv P^\nabla Q^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$, then by induction and considering the vector $Q^\nabla \vec{L}$ we have $P'^\nabla Q^\nabla \vec{L} \in \mathsf{SOL}^m_{sa}$ for some $m \leq n$ and we are done.
If $m = n$, then by induction we used rule $(+_c)$ in deriving $P \longrightarrow^h_{pn} P'$. Therefore we used rule $(+_c)$ also in deriving $M \longrightarrow^h_{pn} N$. □

2.2.12. Lemma. *If $M \longrightarrow^h_{sa} M'$ then, for all $\vec{L}$ and substitution $(\cdot)^\nabla$:*

$$M'^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow \exists N.\ N^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ and } M \longrightarrow^h_{pn} N\ .$$

Proof. By induction on $\longrightarrow^h_{sa}$.

Case $(+_a)$: then assume that $M \equiv P + Q \longrightarrow^h_{sa} P' + Q \equiv M'$ with $P \longrightarrow^h_{sa} P'$. Now

$$(P'^\nabla + Q^\nabla)\vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow P'^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ or } Q^\nabla \vec{L} \notin \mathsf{SOL}_{sa}\ .$$

If $P'^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, choosing $N \equiv P$, we have

$$M \equiv P + Q \longrightarrow^h_{pn} N$$

and by Proposition 2.2.8

$$P'^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow P^\nabla \vec{L} \notin \mathsf{SOL}_{sa}.$$

Otherwise, if $Q^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, we take $N \equiv Q$ and we have $M \equiv P + Q \longrightarrow^h_{pn} N$. The case $M \equiv P + Q \longrightarrow^h_{sa} P + Q' \equiv M'$, with $Q \longrightarrow^h_{sa} Q'$, is symmetric.

Case $(+_{app})$: then $M \equiv (P + Q)R \longrightarrow^h_{sa} PR + QR \equiv M'$. Now

$$(P^\nabla R^\nabla + Q^\nabla R^\nabla)\vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow P^\nabla R^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ or } Q^\nabla R^\nabla \vec{L} \notin \mathsf{SOL}_{sa}\ .$$

If $P^\nabla R^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, then it suffices to choose $N \equiv PR$, with $(P + Q)R \longrightarrow^h_{pn} PR$. Otherwise $Q^\nabla R^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, so that we choose $N \equiv QR$ and we conclude similarly.

Case $(\|_s)$: then $M \equiv P\|Q \longrightarrow^h_{sa} P'\|Q' \equiv M'$ with $P \longrightarrow^h_{sa} P'$ and $Q \longrightarrow^h_{sa} Q'$. If $(P'^\nabla \| Q'^\nabla)\vec{L} \notin \mathsf{SOL}_{sa}$, then both $P'^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$ and $Q'^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, so that, by induction

$$\exists N_1, N_2.\ N_1^\nabla \vec{L}, N_2^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ and } P \longrightarrow^h_{pn} N_1 \text{ and } Q \longrightarrow^h_{pn} N_2\ .$$

Therefore we choose $N \equiv N_1 \| N_2$.

Case $(\|_{app})$: then $M \equiv (P\|Q)R \longrightarrow^h_{sa} PR\|QR \equiv M'$. Hence we take $N \equiv M'$.

Case (β): then $M \equiv (\lambda x.P)Q \longrightarrow^h_{sa} P[Q/x] \equiv M'$. Clearly the choice $N \equiv M'$ works.

Case (ξ): then $M \equiv \lambda x.P \longrightarrow^h_{sa} \lambda x.P' \equiv M'$ with $P \longrightarrow^h_{sa} P'$.
If the vector $\vec{L}$ is empty, then the thesis follows from the induction hypothesis. Otherwise consider the non empty vector $L_0\vec{L}$:

$$\begin{array}{rcl} (\lambda x.P'^\nabla)L_0\vec{L} \notin \mathsf{SOL}_{sa} & \Rightarrow & P'^\nabla[L_0/x]\vec{L} \notin \mathsf{SOL}_{sa} \\ & & \text{by } 2.2.7\textit{(i)} \\ & \Rightarrow & \exists N'.\ N'^\nabla[L_0/x]\vec{L} \notin \mathsf{SOL}_{sa} \text{ and } P \longrightarrow^h_{pn} N' \\ & & \text{by induction.} \end{array}$$

Then $\lambda x.P \longrightarrow^h_{pn} \lambda x.N'$ and we take $N \equiv \lambda x.N'$.

Case (ν_{sa}): then $M \equiv PQ \longrightarrow_{sa}^{h} P'Q \equiv M'$ with $P \longrightarrow_{sa}^{h} P'$ and $P \notin \mathsf{Abst} \cup \mathsf{Par} \cup \mathsf{Sum}$. From the induction hypothesis

$$P'^{\nabla} Q^{\nabla} \vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow \exists N'.\, N'^{\nabla} Q^{\nabla} \vec{L} \notin \mathsf{SOL}_{sa} \text{ and } P \longrightarrow_{pn}^{h} N'\ .$$

Then $PQ \longrightarrow_{pn}^{h} N'Q$ by (ν_{pn}) since in particular $P \notin \mathsf{Abst} \cup \mathsf{Par}$. Therefore we take $N \equiv N'Q$. □

We are now ready to prove that $\longrightarrow_{pn}^{h}$ and $\longrightarrow_{sa}^{h}$ determine the same set of solvable terms. To prove this, we will apply the previous Lemmas, using the identical substitution and the empty vector of terms.

2.2.13. THEOREM. $\mathsf{SOL}_{sa} = \mathsf{SOL}_{pn}$.

PROOF. First we show that $\mathsf{SOL}_{sa} \subseteq \mathsf{SOL}_{pn}$. Toward a contradiction suppose that $M \in \mathsf{SOL}_{sa}$ but $M \notin \mathsf{SOL}_{pn}$. If $M \in \mathsf{SOL}_{sa}$, then there exists n such that $M \in \mathsf{SOL}_{sa}^{n}$. The hypothesis that $M \notin \mathsf{SOL}_{pn}$ implies that there is a sequence $\{M_i\}_{i\in\omega}$ such that $M_0 \equiv M$ and, for all i, $M_i \longrightarrow_{pn}^{h} M_{i+1}$. By Lemma 2.2.11 there is a k such that $M_k \in \mathsf{SOL}_{sa}^{0}$, i.e. M_k is in normal form wrt $\longrightarrow_{sa}^{h}$. This is because the only case in which the n of SOL_{sa}^{n} does not decrease is when in the $\longrightarrow_{pn}^{h}$ reduction rule $(+_c)$ is used. But the number of consecutive steps of this kind is bounded by the number of the occurrences of $+$ in the term to be reduced.
It is easy to see that, if M_k can be further reduced under $\longrightarrow_{pn}^{h}$, then only steps involving the use of $(+_c)$ are possible, which again are bounded by the number of $+$'s in M_k. So any sequence of $\longrightarrow_{pn}^{h}$ reductions out of M has to be finite: a contradiction.

To show that $\mathsf{SOL}_{pn} \subseteq \mathsf{SOL}_{sa}$ assume, toward a contradiction, that $M \in \mathsf{SOL}_{pn}$ and $M \notin \mathsf{SOL}_{sa}$. Then there exists M_1 such that $M \longrightarrow_{sa}^{h} M_1$ and $M_1 \notin \mathsf{SOL}_{sa}$. By Lemma 2.2.12 this implies that there exists N such that $M \longrightarrow_{pn}^{h} N$ and still $N \notin \mathsf{SOL}_{sa}$. Iterating the same reasoning, we build an infinite $\longrightarrow_{pn}^{h}$ reduction out of M, so that $M \notin \mathsf{SOL}_{pn}$: a contradiction.
□

Since our aim is that of developing an unfolding semantics for our calculus, we are interested essentially in the set of solvable terms. So Theorem 2.2.13 gives us the possibility of choosing freely between the reduction relations $\longrightarrow_{pn}$ and $\longrightarrow_{sa}$. For technical reasons we will concentrate in the following on $\longrightarrow_{sa}$. Consequently we will write simply $\longrightarrow$ for it, and SOL for the set of solvable terms.

2.3. Operational Semantics

In the previous section the semantics of our calculi has been described by means of reduction relations. Here we develop a theory to compare terms with respect to their functional behaviors. We do this in two different ways. The first one is by means of contexts. The second one is more refined and compares terms by means of their "approximants", where the set of approximants of a term can be viewed as a generalization to our calculus of the notion of Böhm tree.

2.3.1. Contextual Semantics

Following the standard approach for defining equational theories from convergence predicates (originated with Morris' thesis [75]; see also [17] 16.5.5), we state:

2.3.1. DEFINITION. For any $M, N \in \Lambda_{+\|}$ we define:

$$M \sqsubseteq^{\mathcal{O}} N \Leftrightarrow \forall C[\,].\ C[M] \in \mathsf{SOL} \Rightarrow C[N] \in \mathsf{SOL}.$$

Accordingly,

$$M \simeq^{\mathcal{O}} N \Leftrightarrow M \sqsubseteq^{\mathcal{O}} N \sqsubseteq^{\mathcal{O}} M.$$

Clearly, the relation $\sqsubseteq^{\mathcal{O}}$ is a precongruence. The set SOL, when restricted to pure λ-terms, is the set of terms having a head normal form, that is those terms which are solvable in the classical sense. Hence the restriction of $\simeq^{\mathcal{O}}$ to pure λ-terms is the λ-theory of D_∞ by a well known result of Hyland [56] and Wadsworth [95].

2.3.2. PROPOSITION. *The following (in)-equations hold:*

(i) $(\lambda x.M)N \simeq^{\mathcal{O}} M[N/x]$;	*(vii)* $\lambda x.(M\|N) \simeq^{\mathcal{O}} \lambda x.M\|\lambda x.N$;
(ii) $(M+N)L \simeq^{\mathcal{O}} ML + NL$;	*(viii)* $M + N \sqsubseteq^{\mathcal{O}} M, N$;
(iii) $L(M+N) \sqsubseteq^{\mathcal{O}} LM + LN$;	*(ix)* $L \sqsubseteq^{\mathcal{O}} M, N \Rightarrow L \sqsubseteq^{\mathcal{O}} M + N$;
(iv) $(M\|N)L \simeq^{\mathcal{O}} ML\|NL$;	*(x)* $M, N \sqsubseteq^{\mathcal{O}} M\|N$;
(v) $LM\|LN \sqsubseteq^{\mathcal{O}} L(M\|N)$;	*(xi)* $M, N \sqsubseteq^{\mathcal{O}} L \Rightarrow M\|N \sqsubseteq^{\mathcal{O}} L$.
(vi) $\lambda x.(M+N) \simeq^{\mathcal{O}} \lambda x.M + \lambda x.N$;	

where the inequalities (iii) and (v) are in general proper.

PROOF. We consider only the interesting cases.

To prove that the inequality *(iii)* is proper, let $\Delta \equiv \lambda x.xx$, $M \equiv \lambda x.x(\lambda yzv.v)\Delta$ and $N \equiv \lambda x.\Delta$. ΔM and ΔN both β-reduce to Δ and therefore $\Delta M + \Delta N$ is solvable. Instead, $\Delta(M+N)$ reduces to $\Delta + \Delta\Delta + \Delta + \Delta$, which is unsolvable.

To prove that the inequality *(v)* is proper, let Δ be as above, $\mathbf{I} \equiv \lambda x.x$, $\mathbf{K} \equiv \lambda xy.x$, $T \equiv \lambda x.x\Delta\mathbf{I}\Delta$ and $R \equiv \lambda x.x\Delta\Delta$. Now $(T+R)(\mathbf{I}\|\mathbf{K})$ is solvable since it reduces to $(\Delta\|\Delta\Delta) + (\Delta\Delta\|\Delta)$. Instead, $(T+R)\mathbf{I}\|(T+R)\mathbf{K}$ reduces to $(\Delta + \Delta\Delta)\|(\Delta\Delta + \Delta)$ and therefore it is unsolvable.

(ix). First, we prove the idempotence of $+$. $P + P \sqsubseteq^{\mathcal{O}} P$ follows immediately from *(viii)*. We have that $P \sqsubseteq^{\mathcal{O}} P + P$ follows from *(iii)* choosing $L \equiv \mathbf{K}P$. Now, given an arbitrary context $C[\,]$, let $C'[\,] \equiv C[[\,] + L]$ and $C''[\,] \equiv C[M + [\,]]$. If $L \sqsubseteq^{\mathcal{O}} M, N$, then

$$\begin{array}{rcl} C[L] \in \mathsf{SOL} & \Rightarrow & C[L+L] \equiv C'[L] \in \mathsf{SOL} \Rightarrow C'[M] \equiv C''[L] \in \mathsf{SOL} \\ & \Rightarrow & C''[N] \equiv C[M+N] \in \mathsf{SOL}. \end{array}$$

(xi). Similarly, we prove the idempotence of $\|$ using *(x)* and *(v)*. Now, given an arbitrary context $C[\,]$, let $C'[\,] \equiv C[[\,]\|N]$ and $C''[\,] \equiv C[L\|[\,]]$. If $M, N \sqsubseteq^{\mathcal{O}} L$, then

$$\begin{array}{rcl} C[M\|N] \equiv C'[M] \in \mathsf{SOL} & \Rightarrow & C'[L] \equiv C''[N] \in \mathsf{SOL} \Rightarrow C''[L] \equiv C[L\|L] \in \mathsf{SOL} \\ & \Rightarrow & C[L] \in \mathsf{SOL}. \qquad \Box \end{array}$$

2.3.2. Capabilities Semantics

$\simeq^{\mathcal{O}}$ is an extensional theory by definition, and in fact $\lambda x.(M+N) \simeq^{\mathcal{O}} \lambda x.M + \lambda x.N$ holds. However, if $+$ is interpreted as an operation to form "sets" of values and λx is the standard functional abstraction, then this equality identifies any set of functions with a single multivalued function (see [65, 64]). This is not very natural if one considers that $L(M+N) \not\simeq^{\mathcal{O}} LM + LN$. This problem becomes more evident when modeling the calculus by means of type assignment systems, as we shall do in the forthcoming sections.

For these reasons we introduce a finer, non extensional semantics which is still based on the notion of head normal form and solvability, but uses ideas underlyning Böhm trees. More precisely, we first show the shape of head normal forms in the present setting. Then we associate to each term the set of head normal forms (the *capabilities*) which can be obtained out of it using a more liberal reduction relation ($\longrightarrow_a$, see Definition 2.3.5). Lastly we define a notion of approximation patterned after [95] and we compare terms via the approximate normal forms of their capabilities.

It is easy to verify that the terms irreducible according to $\longrightarrow^h$ (i.e. the head normal forms) satisfy the conditions of the following proposition.

2.3.3. PROPOSITION. *The set $\mathcal{H}$ of* head normal forms *is the least one such that:*

(a) $M_1, \ldots, M_n \in \Lambda_{+\|},\ x \in Var \ \Rightarrow\ xM_1 \ldots M_n \in \mathcal{H} \quad (n \geq 0)$;

(b) $H \in \mathcal{H}, x \in Var \ \Rightarrow\ \lambda x.H \in \mathcal{H}$;

(c) $H_1, H_2 \in \mathcal{H} \ \Rightarrow\ H_1 + H_2 \in \mathcal{H}$;

(d) $H \in \mathcal{H}, M \in \Lambda_{+\|} \ \Rightarrow\ H\|M, M\|H \in \mathcal{H}$.

2.3.4. DEFINITION. The set $\mathcal{H}(M)$ of *head normal forms* of M is defined by:

$$\mathcal{H}(M) \ = \ \{H \in \mathcal{H} \mid M \xrightarrow{*}{}^{h} H\}.$$

For example, let us consider the terms $F\mathsf{0}$ and $G\mathsf{0}$, where

$$F \equiv \Theta(\lambda fx.(x + f(\mathsf{Succ}\ x))), \quad G \equiv \Theta(\lambda fx.(x\|f(\mathsf{Succ}\ x))),$$

$\Theta \equiv (\lambda zx.x(zzx))(\lambda zx.x(zzx))$ is the Turing fixed point combinator, $\mathsf{0}$ and Succ are the zero and successor of Church numerals respectively. Let n be the Church numeral for the natural number n, then it is easy to check that for any n

$$F\mathsf{0} \xrightarrow{*}{}^{h} \mathsf{0} + \mathsf{1} + \ldots + \mathsf{n} + F(\mathsf{Succ}\ \mathsf{n})$$

which is never in $\mathcal{H}$. So $\mathcal{H}(F\mathsf{0}) = \emptyset$.
On the other hand $\mathcal{H}(G\mathsf{0}) = \{\mathsf{0}\|G(\mathsf{Succ}\ \mathsf{0})\}$. However, if we consider its reducts with respect to $\longrightarrow$, then we see that for any n, putting $G' \equiv (\lambda fx.(x\|f(\mathsf{Succ}\ x)))$, we have:

$$\begin{array}{lll} G\mathsf{0} & \xrightarrow{*} & G'G\mathsf{0} \\ & \cdots & \\ & \xrightarrow{*} & \underbrace{G'(\cdots(G'}_{n+1}G)\cdots)\mathsf{0} \\ & \xrightarrow{*} & \mathsf{0}\|\underbrace{G'(\cdots(G'}_{n}G)\cdots)\mathsf{1} \end{array}$$

giving rise to an infinite set of (distinct) head normal forms, none of which even reduces to a head normal form of the shape

$$\mathsf{0}\|\mathsf{1}\|\ldots\|\mathsf{n}\|G(\mathsf{Succ}\ \mathsf{n}),$$

because of the synchronous character of $\|$. This is unfortunate, since the last term is a better candidate for describing the behavior of $G\mathsf{0}$ when it is applied to an argument.

Being $\mathcal{H}$ the set of normal forms wrt $\longrightarrow^h$, by Corollary 2.2.9, it follows that

$$\mathsf{SOL} = \{M \in \Lambda_{+\|} \mid \mathcal{H}(M) \neq \emptyset\}.$$

Observe that $H \in \mathcal{H}(M+N)$ implies $H \equiv H_1 + H_2$ where $H_1 \in \mathcal{H}(M)$ and $H_2 \in \mathcal{H}(N)$, while $H \in \mathcal{H}(M\|N)$ implies $H \equiv L_1\|L_2$ where $L_1 \in \mathcal{H}(M)$ or $L_2 \in \mathcal{H}(N)$, only.

2.3.1. Remark. Since $\longrightarrow^h \subseteq \longrightarrow$, it holds

$$\mathsf{SOL} \subseteq \{M \in \Lambda_{+\|} \mid \exists H \in \mathcal{H}.\ M \xrightarrow{*} H\}.$$

Also the vice-versa is true, because, by a standardization argument, $M \xrightarrow{*} H$ implies that

$$\exists N.\ M \xrightarrow{*}^h N \text{ and } N \xrightarrow{*}^i H$$

where $\longrightarrow^i$ is obtained out of $\longrightarrow$ by forbidding the $\longrightarrow^h$ steps. In other words, only internal redexes are reduced according to $\longrightarrow^i$. But we omit this quite long proof, since we do not need this result.
Notice that, due to the lack of the Church-Rosser property, our language does not fit the conditions of [40], therefore we cannot directly use that proof method.

As it is clear from Proposition 2.3.3, we have shifted to the head normal forms the distinction between the conjunctive behavior of $+$ and the disjunctive nature of $\|$. We capitalize on this fact and we remedy the drawback outlined in the above example by abstracting away from the synchronous reduction of $\|$.

2.3.5. Definition.

(i) Let $\longrightarrow_a$ be the least binary relation on $\Lambda_{+\|}$ which is defined as $\longrightarrow$ adding the clause:

$$M \longrightarrow_a M' \Rightarrow M\|N \longrightarrow_a M'\|N \text{ and } N\|M \longrightarrow_a N\|M'.$$

(ii) The set $\mathcal{C}(M)$ of the capabilities of M is defined by:

$$\mathcal{C}(M) \;=\; \{H \mid \exists H' \in \mathcal{H}(M).\ H' \xrightarrow{*}_a H\}.$$

As examples, consider the terms $F\mathsf{0}$ and $G\mathsf{0}$ and observe that $\mathcal{C}(F\mathsf{0}) = \emptyset$, while

$$\mathsf{0}\|\mathsf{1}\|\ldots\|\mathsf{n}\|G(\mathsf{Succ\ n}) \in \mathcal{C}(G\mathsf{0}) \text{ for all } n \geq 0.$$

We now introduce the formal definition of approximate normal form. This will be useful for comparing the capabilities of terms through their approximate normal forms (see Definition 2.3.9).

2.3.6. Definition. Let $\Lambda_{+\|\Omega}$ be the language obtained from $\Lambda_{+\|}$ by adding the constant Ω. The set of *approximate normal forms* $\mathcal{A} \subset \Lambda_{+\|\Omega}$ is the least one such that:

(i) $\Omega \in \mathcal{A}$;

(ii) $A_1, \ldots, A_n \in \mathcal{A} \Rightarrow xA_1 \ldots A_n \in \mathcal{A}$ $(n \geq 0)$;

(iii) $A \in \mathcal{A} \Rightarrow \lambda x.A \in \mathcal{A}$;

(iv) $A_1, A_2 \in \mathcal{A} \Rightarrow A_1 + A_2, A_1 \| A_2 \in \mathcal{A}$.

We define a preorder relation on approximate normal forms which generalizes the classical one taking into account the intended meanings of $+$ and $\|$. Moreover an η-redex is always less than its contractum according to this preorder.

2.3.7. Definition. Over the set $\mathcal{A}$ define $\preceq$ as the least preorder which makes $\mathcal{A}$ into a distributive lattice with $+$ as meet, $\|$ as join and Ω as bottom, and such that:

(i) $\lambda x.\Omega \preceq \Omega$;

(ii) $A \preceq A' \Rightarrow \lambda x.A \preceq \lambda x.A'$;

(iii) $A_1 \preceq A'_1, \ldots, A_n \preceq A'_n \Rightarrow xA_1 \ldots A_n \preceq xA'_1 \ldots A'_n$;

(iv) $\lambda x.(A \| A') \preceq \lambda x.A \| \lambda x.A'$;

(v) $\lambda y.xA_1 \ldots A_n y \preceq xA_1 \ldots A_n$, if $y \notin FV(xA_1 \ldots A_n)$.

Let $\sim$ be the equivalence relation induced by $\preceq$.

As usual, we associate to each term M an approximate normal form $\phi(M)$ obtained by replacing Ω to all subterms which are not head normal forms.

2.3.8. Definition. Let $\phi: \Lambda_{+\|} \to \mathcal{A}$ be the following map:

(i) $\phi(\lambda x_1 \ldots x_n.xM_1 \ldots M_m) = \lambda x_1 \ldots x_n.x\phi(M_1) \ldots \phi(M_m)$;

(ii) $\phi(\lambda x_1 \ldots x_n.H + H') = \lambda x_1 \ldots x_n.\phi(H) + \phi(H')$, if $H, H' \in \mathcal{H}$;

(iii) $\left.\begin{array}{l} \phi(\lambda x_1 \ldots x_n.M \| H) = \lambda x_1 \ldots x_n.\phi(M) \| \phi(H) \\ \phi(\lambda x_1 \ldots x_n.H \| M) = \lambda x_1 \ldots x_n.\phi(H) \| \phi(M) \end{array}\right\}$ if $H \in \mathcal{H}$;

(iv) $\phi(M) = \Omega$, if $M \notin \mathcal{H}$.

Now we relate the capabilities of two terms by comparing their approximate normal forms in a cofinal way.

2.3.9. Definition. For any $M, N \in \Lambda_{+\|}$ we define:

$$M \sqsubseteq^{\mathcal{A}} N \Leftrightarrow \forall H \in \mathcal{C}(M)\ \exists H' \in \mathcal{C}(N).\ \phi(H) \preceq \phi(H').$$

Accordingly,

$$M \simeq^{\mathcal{A}} N \Leftrightarrow M \sqsubseteq^{\mathcal{A}} N \sqsubseteq^{\mathcal{A}} M.$$

The possibility of taking an element out of the set of capabilities allows us to choose any term obtainable by reducing according to $\longrightarrow_a$. Notice that $\longrightarrow_a$ is the most permissive among the reduction relations we introduced. The fact of considering then the approximate normal form of this term means (as usual) that we disregard redexes.

If one defines the set of approximants of a term as the downward closure of the set of approximate normal forms of its capabilities, one immediately obtains that the relation $\sqsubseteq^{\mathcal{A}}$ coincides with the inclusion between sets of approximants.

2.3.10. Definition. Let $M \in \Lambda_{+\|}$, then the *set* $\mathcal{A}(M)$ *of approximants of* M is defined by:

$$\mathcal{A}(M) = \{A \in \mathcal{A} \mid \exists H \in \mathcal{C}(M).\ A \preceq \phi(H)\} \cup \{\Omega\}.$$

For example,

$$\mathsf{0}\|\mathsf{1}\| \ldots \|\mathsf{n}\|\Omega \in \mathcal{A}(G\mathsf{0}) \text{ for all } n \geq 0.$$

The following properties of the sets of approximants follow immediately from previous definitions.

2.3.11. Proposition.

(i) $\mathcal{A}(M + N) = \mathcal{A}(M) \cap \mathcal{A}(N)$;

(ii) $\mathcal{A}(M\|N) = \{H\|H' \mid H \in \mathcal{A}(M)$ *and* $H' \in \mathcal{A}(N)\}$;

(iii) $M \sqsubseteq^{\mathcal{A}} N \Leftrightarrow \mathcal{A}(M) \subseteq \mathcal{A}(N)$;

(iv) $M \xrightarrow{*}{}^{h} N \Rightarrow \mathcal{A}(N) \subseteq \mathcal{A}(M)$.

2.3.2. Remark. 2.3.11 *(iv)* is weak. Indeed a stronger connection between the reduction relation and the sets of approximate normal forms holds, i.e.:

$$M =_a N \Rightarrow \mathcal{A}(M) = \mathcal{A}(N).$$

This will follow from the subject conversion of $\mathcal{C}$ (Theorem 2.5.9) and the full abstraction of the filter model (Theorem 2.6.23).

Now we can prove for our calculus a standard property of λ-calculus: a term is solvable iff it has an approximant different from Ω.

2.3.12. Proposition.

(i) $M \in \mathsf{SOL} \iff \mathcal{C}(M) \neq \emptyset$;

(ii) $M \in \mathsf{SOL} \iff \mathcal{A}(M) \neq \{\Omega\}$.

Proof. *(i)* follows from Definitions 2.3.4, 2.3.5*(ii)* and Corollary 2.2.9.
(ii) is a consequence of *(i)* and of Definition 2.3.10. □

One would expect $\sqsubseteq^{\mathcal{A}}$ to be a refinement of $\sqsubseteq^{\mathcal{O}}$; this is in fact true. A direct proof based on an approximation theorem á la Wadsworth [95] is possible, but we will obtain it for free from the adequacy and full abstraction results of Section 2.6.

2.4. Simple Types and Semimodels

In this section we type the terms of our calculus by means of simple types and we define a set semimodel in the sense of [83].

2.4.1. The Type Assignment System $\mathcal{B}$

Curry types are thought of as properties of terms. The properties in which we are mainly interested concern solvability. This guides the choice of typing rules for $+$ and $\|$.
Indeed to assure that $M + N$ normalizes with respect to $\longrightarrow^h$, we have to prove that both M and N have the same property. Generalizing to arbitrary properties we type $M + N$ with σ if both M and N can be typed with σ. This is also the choice of [1].
Conversely, for $M\|N$ to be normalizable it suffices that either M or N normalizes. Extending this notion to arbitrary properties, it follows that one is entitled to type $M\|N$ with σ as soon as either M or N (or both) can be typed with σ. See [22] for further motivations.

Let the set *Type* of types be defined by

$$\sigma ::= t \mid \sigma \to \sigma,$$

where t ranges over a denumerable collection of type variables. A *statement* is an expression of the form $M{:}\,\sigma$, where M is a λ-term and σ a type. A *basis* Γ is a set of statements such that subjects are pairwise distinct variables. As usual, $FV(\Gamma)$ is the set of term variables in Γ.

2.4.1. Definition (The System $\mathcal{B}$). The axioms and rules of the basic assignment system $\mathcal{B}$ are the following:

$$(\mathrm{Ax})\quad \Gamma, x{:}\,\sigma \vdash x{:}\,\sigma$$

$$(\to \mathrm{I})\quad \frac{\Gamma, x{:}\,\sigma \vdash M{:}\,\tau}{\Gamma \vdash \lambda x.M{:}\,\sigma \to \tau} \qquad (\to \mathrm{E})\quad \frac{\Gamma \vdash M{:}\,\sigma \to \tau \qquad \Gamma \vdash N{:}\,\sigma}{\Gamma \vdash MN{:}\,\tau}$$

$$(+\,\mathrm{I})\quad \frac{\Gamma \vdash M{:}\,\sigma \qquad \Gamma \vdash N{:}\,\sigma}{\Gamma \vdash M + N{:}\,\sigma} \qquad (\|\,\mathrm{I})\quad \frac{\Gamma \vdash M{:}\,\sigma}{\Gamma \vdash M\|N{:}\,\sigma} \qquad \frac{\Gamma \vdash N{:}\,\sigma}{\Gamma \vdash M\|N{:}\,\sigma}$$

If $\Gamma \vdash M{:}\,\sigma$ is provable in $\mathcal{B}$, we write $\Gamma \vdash_{\mathcal{B}} M{:}\,\sigma$.

In this system, as in Curry's original one, there is a correspondence between the main constructor of the subject of the conclusion in each rule and the rule itself; this does not hold for the type. However, classical terms (i.e. those without occurrences of $+$ and $\|$) have just their simple types. This property results in a simple theory of the type assignment system.

A routine induction on derivations in $\mathcal{B}$ shows:

2.4.2. Lemma (Structural Properties of Deductions in $\mathcal{B}$).

(i) $\Gamma \vdash_{\mathcal{B}} x{:}\,\tau \;\Leftrightarrow\; x{:}\,\tau \in \Gamma$;

(ii) $\Gamma \vdash_{\mathcal{B}} \lambda x.M{:}\,\sigma \to \tau \;\Leftrightarrow\; \Gamma, x{:}\,\sigma \vdash_{\mathcal{B}} M{:}\,\tau$;

(iii) $\Gamma \vdash_{\mathcal{B}} MN{:}\,\tau \;\Leftrightarrow\; \Gamma \vdash_{\mathcal{B}} M{:}\,\sigma \to \tau$ *and* $\Gamma \vdash_{\mathcal{B}} N{:}\,\sigma$ *for some* σ;

(iv) $\Gamma \vdash_{\mathcal{B}} M + N{:}\,\sigma \;\Leftrightarrow\; \Gamma \vdash_{\mathcal{B}} M{:}\,\sigma$ *and* $\Gamma \vdash_{\mathcal{B}} N{:}\,\sigma$;

(v) $\Gamma \vdash_{\mathcal{B}} M\|N{:}\,\sigma \;\Leftrightarrow\; \Gamma \vdash_{\mathcal{B}} M{:}\,\sigma$ *or* $\Gamma \vdash_{\mathcal{B}} N{:}\,\sigma$.

Using this lemma it is easy to prove the following corollary by induction on the definition of $\longrightarrow_a$. We consider this reduction, since it includes $\longrightarrow$ (which includes $\longrightarrow^h$).

2.4.3. COROLLARY (Subject Reduction of $\mathcal{B}$). $\Gamma \vdash_{\mathcal{B}} M{:}\sigma$ *and* $M \xrightarrow{*}_a N \Rightarrow \Gamma \vdash_{\mathcal{B}} N{:}\sigma$.

As an immediate consequence of 2.4.3, we have the subject reduction property of $\mathcal{B}$ for $\xrightarrow{*}$.

2.4.1. REMARK. As stated in [31], also $\xrightarrow{*}_{pn}$ enjoys the subject reduction property.

2.4.2. The Set Semimodel

For the classical λ-calculus, a filter model construction with simple types, even considering as a "filter" any set of types, does not yield a λ-model (see e.g. [54]). Indeed the best one can obtain is a *semimodel* in the sense of [83]. I.e. a model in which interreducible terms are equal, but in general convertible terms are not (M, N are interreducible iff $M \xrightarrow{*} N$ and $N \xrightarrow{*} M$). Adapting Plotkin's definition to the present context (see also [1]) we introduce the following notion:

2.4.4. DEFINITION. A *semimodel* is a structure

$$\mathcal{P} = \langle P, \sqsubseteq, \cdot, \sqcap, \sqcup, [\![\cdot]\!]^{\mathcal{P}} \rangle$$

where $\langle P, \sqsubseteq \rangle$ is a poset, and $\cdot, \sqcap, \sqcup$ are binary monotonic operations that satisfy the following requirements:

$$d \sqcap e \sqsubseteq d, d \sqcap e \sqsubseteq e, d \sqsubseteq d \sqcup e, e \sqsubseteq d \sqcup e$$
$$\text{and} \qquad (d \sqcup d') \cdot e \sqsubseteq (d \cdot e) \sqcup (d' \cdot e).$$

Finally $[\![\cdot]\!]^{\mathcal{P}}{:}\, \Lambda_{+\|} \times \mathit{Env} \to P$, where $\mathit{Env} = \{\rho \mid \rho{:}\, \mathit{Var} \to P\}$, is such that:

(a) $[\![M + N]\!]^{\mathcal{P}}_\rho = [\![M]\!]^{\mathcal{P}}_\rho \sqcap [\![N]\!]^{\mathcal{P}}_\rho$;

(b) $[\![M \| N]\!]^{\mathcal{P}}_\rho = [\![M]\!]^{\mathcal{P}}_\rho \sqcup [\![N]\!]^{\mathcal{P}}_\rho$;

(c) $[\![x]\!]^{\mathcal{P}}_\rho = \rho(x)$;

(d) $[\![MN]\!]^{\mathcal{P}}_\rho = [\![M]\!]^{\mathcal{P}}_\rho \cdot [\![N]\!]^{\mathcal{P}}_\rho$;

(e) $\forall d \in P.\ [\![\lambda x.M]\!]^{\mathcal{P}}_\rho \cdot d \sqsubseteq [\![M]\!]^{\mathcal{P}}_{\rho[d/x]}$;

(f) $\forall x \in \mathrm{FV}(M).\ \rho(x) = \rho'(x) \Rightarrow [\![M]\!]^{\mathcal{P}}_\rho = [\![M]\!]^{\mathcal{P}}_{\rho'}$;

(g) $(\forall d \in P.\ [\![M]\!]^{\mathcal{P}}_{\rho[d/x]} \sqsubseteq [\![N]\!]^{\mathcal{P}}_{\rho[d/x]}) \Rightarrow [\![\lambda x.M]\!]^{\mathcal{P}}_\rho \sqsubseteq [\![\lambda x.N]\!]^{\mathcal{P}}_\rho$.

Semimodels interpret the reduction relation, as stated in the following proposition, which can be proved by induction on the definition of $\longrightarrow_a$. In the case of $(M+N)L \longrightarrow_a ML + NL$ this follows from the monotonicity of the application which implies $(d \sqcap d') \cdot e \sqsubseteq (d \cdot e) \sqcap (d' \cdot e)$.

2.4.5. PROPOSITION. $M \xrightarrow{*}_a N \Rightarrow \forall \rho.\ [\![M]\!]^{\mathcal{P}}_\rho \sqsubseteq [\![N]\!]^{\mathcal{P}}_\rho$ *for all semimodels* $\mathcal{P}$.

Notice that Proposition 2.4.5 holds even if $\xrightarrow{*}_a$ is replaced by $\xrightarrow{*}_{pn}$. In the case of the classical λ-calculus one has $\Leftrightarrow$ (see [83]). Here, instead, completeness with respect to reduction does not hold: e.g. we have, by definition, that $\forall \rho \in \mathit{Env}.\ [\![M]\!]_\rho \sqsubseteq [\![M \| N]\!]_\rho$ but we do not have $M \xrightarrow{*}_a M \| N$. This does not seem to be unfortunate; indeed we are looking for a partial order (and its relative equivalence) which is, in a sense, more abstract than reducibility.

As expected, the type assignment $\mathcal{B}$ induces a semimodel.

2.4.6. PROPOSITION. *For $a, b \subseteq Type$, let $a \cdot b = \{\tau \in Type \mid \exists \sigma \in b.\, \sigma \to \tau \in a\}$ and*

$$[\![M]\!]^{\mathcal{B}}_{\rho} = \{\sigma \mid \Gamma \vdash_{\mathcal{B}} M{:}\sigma \textit{ for some } \Gamma \subseteq \{x{:}\tau \mid \tau \in \rho(x)\}\}.$$

The structure

$$\langle \wp(Type), \subseteq, \cdot, \cap, \cup, [\![\cdot]\!]^{\mathcal{B}} \rangle$$

is a semimodel (the set semimodel*).*

The interpretation of the parallel and non-deterministic constructors in the set semimodel can also be easily stated using set theoretic operators. I.e., for all ρ:

$$[\![M + N]\!]^{\mathcal{B}}_{\rho} = [\![M]\!]^{\mathcal{B}}_{\rho} \cap [\![N]\!]^{\mathcal{B}}_{\rho} \text{ and } [\![M \| N]\!]^{\mathcal{B}}_{\rho} = [\![M]\!]^{\mathcal{B}}_{\rho} \cup [\![N]\!]^{\mathcal{B}}_{\rho}.$$

To interpret types over a given semimodel we use the *simple semantics* of types (see [52, 83]).

2.4.7. DEFINITION. A *type structure* over $\mathcal{P} = \langle P, \sqsubseteq, \cdot, \sqcap, \sqcup, [\![\cdot]\!]^{\mathcal{P}} \rangle$ is a pair $\langle \mathcal{T}, \Rightarrow \rangle$ where:

(i) $\mathcal{T} \subseteq \{X \in \wp(P) \mid X$ is not empty, upper closed and $d, e \in X$ imply $d \sqcap e \in X\}$;

(ii) $\Rightarrow$ is a binary function over $\mathcal{T}$ such that

(a) $X \Rightarrow Y \subseteq \{d \in P \mid \forall e \in X.\, d \cdot e \in Y\}$,
(b) $d \in X$ and $[\![M]\!]^{\mathcal{P}}_{\rho[d/x]} \in Y$ imply $[\![\lambda x.M]\!]^{\mathcal{P}}_{\rho} \in X \Rightarrow Y$,
for all $X, Y \in \mathcal{T}$.

2.4.8. DEFINITION.

(i) A *type environment* is a map η from type variables to $\mathcal{T}$.

(ii) $[\![\sigma]\!]^{\mathcal{T}}_{\eta} \in \mathcal{T}$ is defined by

$$[\![t]\!]^{\mathcal{T}}_{\eta} = \eta(t) \quad \text{and} \quad [\![\sigma \to \tau]\!]^{\mathcal{T}}_{\eta} = [\![\sigma]\!]^{\mathcal{T}}_{\eta} \Rightarrow [\![\tau]\!]^{\mathcal{T}}_{\eta}.$$

(iii) A basis Γ *satisfies* ρ and η iff, for all $x{:}\tau \in \Gamma$, $\rho(x) \in [\![\tau]\!]^{\mathcal{T}}_{\eta}$.

(iv) $\Gamma \models M{:}\sigma \;\Leftrightarrow\; \forall \mathcal{P}, \langle \mathcal{T}, \Rightarrow \rangle$ over $\mathcal{P}$, ρ, η. Γ satisfies $\rho, \eta \Rightarrow [\![M]\!]^{\mathcal{P}}_{\rho} \in [\![\sigma]\!]^{\mathcal{T}}_{\eta}$.

2.4.9. THEOREM (Completeness of $\mathcal{B}$). $\Gamma \vdash_{\mathcal{B}} M{:}\sigma \;\Leftrightarrow\; \Gamma \models M{:}\sigma$.

PROOF. This proof essentially adapts Plotkin's completeness proof in [83].
($\Rightarrow$) Simple induction on the derivation of $\Gamma \vdash M{:}\sigma$. If the last applied rule is ($\to$ I), the thesis follows from 2.4.7*(ii) (b)*. For rule (+ I) use 2.4.7*(i)*.
($\Leftarrow$) Using the set semimodel. If we define:

$$\chi_{\sigma} = \{a \subseteq Type \mid \sigma \in a\}, \mathcal{T} = \{\chi_{\sigma}\}_{\sigma \in Type}, \text{ and } \chi_{\sigma} \Rightarrow \chi_{\tau} = \chi_{\sigma \to \tau},$$

then the pair $\langle \mathcal{T}, \Rightarrow \rangle$ is a type structure for the set semimodel.
We take ρ and η such that $\rho(x) = \{\sigma \mid x{:}\sigma \in \Gamma\}$ for every term variable x and $\eta(t) = \chi_t$ for every type variable t. Then we have $[\![\sigma]\!]^{\mathcal{T}}_{\eta} = \chi_{\sigma}$ for all $\sigma \in Type$ and $[\![M]\!]^{\mathcal{B}}_{\rho} \in [\![\sigma]\!]^{\mathcal{T}}_{\eta}$, which imply $\Gamma \vdash_{\mathcal{B}} M{:}\sigma$. □

The set semimodel allows to define a preorder over terms which is a precongruence:

$$M \sqsubseteq^{\mathcal{B}} N \Leftrightarrow_{def} \forall\rho.\ [\![M]\!]^{\mathcal{B}}_{\rho} \subseteq [\![N]\!]^{\mathcal{B}}_{\rho}.$$

We list in the following proposition the main (in)-equations holding in the set semimodel semantics.

2.4.10. PROPOSITION. *Let* $\simeq^{\mathcal{B}} = \sqsubseteq^{\mathcal{B}} \cap \sqsupseteq^{\mathcal{B}}$, *then:*

(i) $(\lambda x.M)N \sqsubseteq^{\mathcal{B}} M[N/x]$;
(ii) $(M+N)L \sqsubseteq^{\mathcal{B}} ML+NL$;
(iii) $L(M+N) \sqsubseteq^{\mathcal{B}} LM+LN$;
(iv) $(M\|N)L \simeq^{\mathcal{B}} ML\|NL$;
(v) $L(M\|N) \simeq^{\mathcal{B}} LM\|LN$;
(vi) $\lambda x.(M+N) \simeq^{\mathcal{B}} \lambda x.M+\lambda x.N$;
(vii) $\lambda x.(M\|N) \simeq^{\mathcal{B}} \lambda x.M\|\lambda x.N$;
(viii) $M+N \sqsubseteq^{\mathcal{B}} M,N$;
(ix) $L \sqsubseteq^{\mathcal{B}} M,N \Rightarrow L \sqsubseteq^{\mathcal{B}} M+N$;
(x) $M,N \sqsubseteq^{\mathcal{B}} M\|N$;
(xi) $M,N \sqsubseteq^{\mathcal{B}} L \Rightarrow M\|N \sqsubseteq^{\mathcal{B}} L$,

where the inequalities (i), (ii) and (iii) are in general proper.

PROOF. By the Completeness of $\mathcal{B}$ (Theorem 2.4.9) we have

$$M \sqsubseteq^{\mathcal{B}} N \Leftrightarrow \forall\Gamma,\sigma.\ \Gamma \vdash_{\mathcal{B}} M{:}\,\sigma \Rightarrow \Gamma \vdash_{\mathcal{B}} N{:}\,\sigma.$$

The positive statements are straightforward consequences of the structural properties of deductions (Lemma 2.4.2). To prove that the inequality *(i)* is proper observe that *(i)* essentially claims that the set semimodel is not a λ-model. To see *(ii)*, let

$$\Gamma = \{x{:}\,\sigma_1 \to \tau, y{:}\,\sigma_2 \to \tau, z{:}\,\sigma_1, v{:}\,\sigma_2\}$$

where $\sigma_1 \not\equiv \sigma_2$. Then $\Gamma \vdash_{\mathcal{B}} x(z\|v)+y(z\|v){:}\,\tau$, but $\Gamma \not\vdash_{\mathcal{B}} (x+y)(z\|v){:}\,\tau$ since $x+y$ has no type. Similarly, for *(iii)*, we have that $\Gamma \vdash_{\mathcal{B}} (x\|y)z+(x\|y)v{:}\,\tau$, but $\Gamma \not\vdash_{\mathcal{B}} (x\|y)(z+v){:}\,\tau$ since $z+v$ has no type. $\square$

Comparing the properties of $\sqsubseteq^{\mathcal{B}}$ with those of $\sqsubseteq^{\mathcal{O}}$ (Proposition 2.3.2) and of $\sqsubseteq^{\mathcal{A}}$ (Proposition 2.5.19) it turns out that the set semimodel does agree neither with the operational semantics á la Morris nor with the inclusion of sets of approximants. This failure suggests us to look at a more expressive type assignment system.

2.5. Intersection, Union Types and λ-lattices

In this section we extend the notion of filter model introduced in [18] to our calculus, the aim being this time to interpret the terms of $\Lambda_{+\|}$ in such a way that the usual λ-calculus equations hold and which fits better the operational behavior of $+$ and $\|$.

2.5.1. The Set of Types and its Preorder

Let us redefine the syntax of types as follows:

$$\sigma ::= t \mid \omega \mid \sigma \to \sigma \mid \sigma \wedge \sigma \mid \sigma \vee \sigma,$$

and call again *Type* the resulting set. In writing types, we assume that $\wedge$ and $\vee$ take precedence over $\to$.

It is clear that to build a filter model a critical choice is that of the preorder between types, since this preorder will appear in a subtyping rule.

2.5.1. DEFINITION.

(i) Let $\leq$ be the smallest preorder over types s.t. $\langle \mathit{Type}, \leq\rangle$ is a distributive lattice (taking the quotient), in which $\wedge$ is the meet, $\vee$ is the join and ω is the top, and moreover the arrow satisfies:

(a) $\omega \leq \omega \to \omega$;

(b) $(\sigma \to \mu) \wedge (\sigma \to \tau) \leq \sigma \to \mu \wedge \tau$;

(c) $\sigma' \leq \sigma, \tau \leq \tau' \Rightarrow \sigma \to \tau \leq \sigma' \to \tau'$.

(ii) Let $\sigma = \tau$ mean $\sigma \leq \tau \leq \sigma$.

The subtype relation $\leq$ can be presented axiomatically by adding the inequalities (a)-(c) to any standard axiomatization of distributive lattices. For proof purpose we assume that such a presentation has been fixed.

We need some properties of the $\leq$ relation, whose proof requires a stratification of *Type*.

2.5.2. DEFINITION (Stratification of *Type*). Let us define three subsets T_0, T_1, T_2 of *Type* recursively:

- $t \in T_0$;
- $\omega \in T_2$;
- $\sigma \in T_2, \tau \in T_1 \Rightarrow \sigma \to \tau \in T_0$;
- $n \geq 1, \sigma_1, \ldots, \sigma_n \in T_0 \Rightarrow \sigma_1 \vee \ldots \vee \sigma_n \in T_1$;
- $n \geq 1, \sigma_1, \ldots, \sigma_n \in T_1 \Rightarrow \sigma_1 \wedge \ldots \wedge \sigma_n \in T_2$.

2.5.1. REMARK. Notice that the set T_2, when restricted to types without $\vee$ occurrences, coincides with the set of normal type schemes of [53] and the set of strict types of [13]. Normal type schemes in [53] were introduced to prove the properties stated in Lemma 2.5.6 (for types without $\vee$). Strict types, instead, have been introduced with a different preorder to obtain a syntax directed type assignment system in [13, 15].

Taking $n = 1$ in the clauses above, one sees that $T_0 \subseteq T_1 \subseteq T_2$, and such inclusions are clearly proper.

Over each of these sets we introduce a preorder.

2.5.3. DEFINITION. $\leq_i \subseteq T_i \times T_i$ is the least preorder such that:

$(\leq_0)$ $\sigma \leq_0 \tau \;\Leftrightarrow\; \sigma \equiv \tau$ or ($\sigma \equiv \sigma' \to \sigma''$ and $\tau \equiv \tau' \to \tau''$ and $\tau' \leq_2 \sigma'$ and $\sigma'' \leq_1 \tau''$);

$(\leq_1)$ $\sigma_1 \vee \ldots \vee \sigma_n \leq_1 \tau_1 \vee \ldots \vee \tau_m \;\Leftrightarrow\; \forall i \leq n\, \exists j \leq m.\, \sigma_i \leq_0 \tau_j$;

$(\leq_2)$ $\sigma \leq_2 \tau \;\Leftrightarrow\; \tau \equiv \omega$ or ($\sigma \equiv \sigma_1 \wedge \ldots \wedge \sigma_n, \tau \equiv \tau_1 \wedge \ldots \wedge \tau_m$ and $\forall j \leq m\, \exists i \leq n.\, \sigma_i \leq_1 \tau_j$).

For each type in *Type* we can find an equivalent type in T_2; this means that we can limit ourself to consider types in T_2, provided that there is a map $(\;)^*$ associating to each type in $Type$ a standard form in T_2.

Notation. In writing $\tau^* \equiv \bigwedge_{i \in I} \tau_i$ we assume that $\tau_i \in T_1$ for all $i \in I$.

2.5.4. DEFINITION. The map $(\;)^*: \mathit{Type} \to T_2$ is defined by:

$t^* \equiv t,\ \omega^* \equiv \omega$

$$(\sigma \to \tau)^* \equiv \begin{cases} \bigwedge_{i\in I}(\sigma^* \to \tau_i) & \text{if } \tau^* \equiv \bigwedge_{i\in I}\tau_i \text{ and } \tau^* \not\equiv \omega \\ \omega & \text{otherwise} \end{cases}$$

$$(\sigma \vee \tau)^* \equiv \begin{cases} \bigwedge_{i\in I}\bigwedge_{j\in J}(\sigma_i \vee \tau_j) & \text{if } \sigma^* \equiv \bigwedge_{i\in I}\sigma_i,\ \sigma^* \not\equiv \omega \text{ and } \tau^* \equiv \bigwedge_{j\in J}\tau_j,\ \tau^* \not\equiv \omega \\ \omega & \text{otherwise} \end{cases}$$

$$(\sigma \wedge \tau)^* \equiv \begin{cases} \sigma^* & \text{if } \tau^* \equiv \omega \\ \tau^* & \text{if } \sigma^* \equiv \omega \\ \sigma^* \wedge \tau^* & \text{otherwise.} \end{cases}$$

2.5.5. Proposition. *For all $\sigma, \tau \in$ Type:*

(i) $\sigma = \sigma^*$;

(ii) $\sigma, \tau \in T_i, \sigma \leq_i \tau \Rightarrow \sigma \leq \tau$ *for* $i = 0, 1, 2$;

(iii) $\sigma \leq \tau \Rightarrow \sigma^* \leq_2 \tau^*$.

Proof. *(i)* By induction on the definition of the map $(\ )^*$.

(ii) By induction on the definition of $\leq_i$.

(iii) By induction on the formal derivation of $\sigma \leq \tau$. □

2.5.6. Lemma.

(i) $\mu \wedge \nu \leq \sigma \to \tau$ *and* $\mu \neq \omega$ *and* $\nu \neq \omega \Rightarrow \exists \tau_1, \tau_2.\ \tau = \tau_1 \wedge \tau_2$ *and* $\mu \leq \sigma \to \tau_1$ *and* $\nu \leq \sigma \to \tau_2$;

(ii) $\bigwedge_{i\in I}(\mu_i \to \nu_i) \leq \sigma \to \tau$ *and* $\tau \neq \omega \Rightarrow \exists J \subseteq I.\ \sigma \leq \bigwedge_{j\in J}\mu_j$ *and* $\bigwedge_{j\in J}\nu_j \leq \tau$.

Proof. *(i)* : let

$$(\mu \wedge \nu)^* = \bigwedge_{i\in I}\mu_i \wedge \bigwedge_{j\in J}\nu_j \text{ and } (\sigma \to \tau)^* = \bigwedge_{k\in K}(\sigma^* \to \pi_k),$$

supposing $\mu^* = \bigwedge_{i\in I}\mu_i, \nu^* = \bigwedge_{j\in J}\nu_j$ and $\tau^* = \bigwedge_{k\in K}\pi_k$. Using 2.5.5*(i)*, *(ii)*, *(iii)* and the definition of $\leq_2$, we have that

$$\forall k.\ (\exists i.\ \mu_i \leq_1 \sigma^* \to \pi_k) \text{ or } (\exists j.\ \nu_i \leq_1 \sigma^* \to \pi_k).$$

Therefore we can choose τ_1 as the intersection of the π_k which satisfy the first inequality and τ_2 as the intersection of the remaining π_k. If one of these intersections is empty, we choose ω for the corresponding τ_i $(i = 1, 2)$.

(ii) : let $\nu_i^* = \bigwedge_{l\in L}\nu_{i,l}$ (where L depends on i) and $\tau^* = \bigwedge_{k\in K}\tau_k$. Then, by 2.5.5 *(iii)* and Definition 2.5.4,

$$\bigwedge_{i\in I}(\mu_i \to \nu_i) \leq \sigma \to \tau \Rightarrow \bigwedge_{i\in I}\bigwedge_{l\in L}(\mu_i^* \to \nu_{i,l}) \leq_2 \bigwedge_{k\in K}(\sigma^* \to \tau_k).$$

It follows that

$$\forall k\ \exists i, l.\ \mu_i^* \to \nu_{i,l} \leq_1 \sigma^* \to \tau_k,$$

which in this case is equivalent to

$$\forall k\ \exists i, l.\ \mu_i^* \to \nu_{i,l} \leq_0 \sigma^* \to \tau_k,$$

and hence

$$\forall k\ \exists i, l.\ \sigma^* \leq_2 \mu_i^* \text{ and } \nu_{i,l} \leq_1 \tau_k.$$

So we conclude

$$\forall k\ \exists i.\ \sigma \leq \mu_i \text{ and } \bigwedge_{l \in L} \nu_{i,l} \leq_2 \tau_k.$$

Taking J as the set of i's which satisfy these inequalities for some $k \in K$, we are done.□

2.5.2. The Type Assignment System $\mathcal{C}$

We introduce now a type assignment system for our extended language of types. We add a rule (ω) which takes into account the universal character of ω, and two standard rules of introduction of $\wedge$ and $\vee$. Moreover we use the preorder on types defined in previous section in a subtyping rule.

Notice that a rule of $\wedge$ elimination is derivable, while a rule of $\vee$ elimination would be unsound (see Remark 2.5.2*(ii)*).

2.5.7. Definition. The system $\mathcal{C}$ is obtained by adding to the basic system $\mathcal{B}$ the following axiom and rules:

$$(\omega) \quad \Gamma \vdash M{:}\omega \qquad\qquad (\wedge \mathrm{I}) \ \frac{\Gamma \vdash M{:}\sigma \quad \Gamma \vdash M{:}\tau}{\Gamma \vdash M{:}\sigma \wedge \tau}$$

$$(\vee \mathrm{I}) \ \frac{\Gamma \vdash M{:}\sigma}{\Gamma \vdash M{:}\sigma \vee \tau} \qquad \frac{\Gamma \vdash M{:}\tau}{\Gamma \vdash M{:}\sigma \vee \tau} \qquad (\leq) \ \frac{\Gamma \vdash M{:}\sigma \quad \sigma \leq \tau}{\Gamma \vdash M{:}\tau}$$

If $\Gamma \vdash M{:}\sigma$ is provable in the system $\mathcal{C}$ we write $\Gamma \vdash M{:}\sigma$.

Notation. In the following we shall sometimes refer to a stronger basis which can be formed out of two given bases. This is done by taking the intersection of the types which are predicates of the same variable:

$$\begin{array}{rcl} \Gamma \uplus \Gamma' & = & \{x{:}\sigma \wedge \tau \mid x{:}\sigma \in \Gamma \text{ and } x{:}\tau \in \Gamma'\} \\ & \cup & \{x{:}\sigma \mid x{:}\sigma \in \Gamma \text{ and } x \notin FV(\Gamma')\} \\ & \cup & \{x{:}\tau \mid x{:}\tau \in \Gamma' \text{ and } x \notin FV(\Gamma)\}. \end{array}$$

Accordingly we define:

$$\Gamma \Subset \Gamma' \ \Leftrightarrow\ \exists \Gamma''. \Gamma \uplus \Gamma'' = \Gamma'.$$

2.5.2. Remark.

(i) Of course rule $(\vee \mathrm{I})$ is derivable. The following rules are admissible:

$$\frac{\Gamma \vdash M{:}\sigma \quad \Gamma \vdash N{:}\tau}{\Gamma \vdash M + N{:}\sigma \vee \tau} \qquad \frac{\Gamma \vdash M{:}\sigma \quad \Gamma \vdash N{:}\tau}{\Gamma \vdash M \| N{:}\sigma \wedge \tau}$$

$$\frac{\Gamma \vdash M{:}\sigma \wedge \tau}{\Gamma \vdash M{:}\sigma} \qquad \frac{\Gamma \vdash M{:}\sigma \wedge \tau}{\Gamma \vdash M{:}\tau} \qquad \frac{\Gamma, x{:}\sigma \vdash M{:}\tau \quad \sigma' \leq \sigma}{\Gamma, x{:}\sigma' \vdash M{:}\tau}.$$

(ii) A natural rule of $\vee$ elimination in the present setting would be:

$$(\vee \mathrm{E}) \quad \frac{\Gamma, x{:}\,\sigma \vdash M{:}\,\mu \qquad \Gamma, x{:}\,\tau \vdash M{:}\,\mu \qquad \Gamma \vdash N{:}\,\sigma \vee \tau}{\Gamma \vdash M[N/x]{:}\,\mu}.$$

This is a rule of the system proposed in [16], where only pure λ-terms are considered. In presence of $+$ and of the corresponding typing rule, however, rule $(\vee\mathrm{E})$ causes the loss of the subject reduction property (established below in Theorem 2.5.9).
Moreover with $(\vee\mathrm{E})$ we would lose also the property (proved in Corollary 2.6.9) that unsolvable terms have only types equivalent to ω.
We give an example showing both failures. Let $\mathbf{I}, \mathbf{K}, \Delta$ be as in the proof of Proposition 2.3.2, and $\mathbf{O} \equiv \lambda xy.y$, then we have:

$$\begin{array}{ll} & x{:}\,(\nu \to \omega \to \nu) \wedge \nu \vdash xx\mathbf{KI}(\Delta\Delta){:}\,\mu, \\ & x{:}\,\omega \to \nu \to \nu \vdash xx\mathbf{KI}(\Delta\Delta){:}\,\mu, \\ \text{and} & \vdash \mathbf{K} + \mathbf{O}{:}\,((\nu \to \omega \to \nu) \wedge \nu) \vee (\omega \to \nu \to \nu), \\ \text{where} & \mu \equiv t \to t, \;\; \nu \equiv \mu \to \omega \to \mu. \end{array}$$

This can be easily checked considering that

$$\vdash \mathbf{I}{:}\,\mu, \;\; \vdash \mathbf{K}{:}\,(\nu \to \omega \to \nu) \wedge \nu \text{ and } \vdash \mathbf{O}{:}\,\omega \to \nu \to \nu.$$

Therefore using $(\vee\mathrm{E})$ we could derive:

$$M \equiv (\mathbf{K} + \mathbf{O})(\mathbf{K} + \mathbf{O})\mathbf{KI}(\Delta\Delta){:}\,\mu.$$

But M reduces to $\mathbf{I} + \Delta\Delta + \mathbf{I} + \mathbf{I}$ and therefore it is unsolvable. We lose subject reduction, since only type ω can be deduced for $\Delta\Delta$, and hence for $\mathbf{I} + \Delta\Delta + \mathbf{I} + \mathbf{I}$. Moreover M is unsolvable but it has type $\mu \not\sim \omega$.

(iii) Notice that

$$\sigma \vee \tau \to \mu \leq (\sigma \to \mu) \wedge (\tau \to \mu),$$

but the converse does not hold. The equality is derivable in the system proposed in [16]. In the present system, by postulating

$$(\sigma \to \mu) \wedge (\tau \to \mu) \leq \sigma \vee \tau \to \mu$$

we would have the same problems we discussed in *(ii)* with rule $(\vee\mathrm{E})$. In fact the following derivation would be possible:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x{:}\sigma \vdash M{:}\,\mu}{\Gamma \vdash \lambda x.M{:}\,\sigma \to \mu}(\to \mathrm{I}) \quad \dfrac{\Gamma, x{:}\tau \vdash M{:}\,\mu}{\Gamma \vdash \lambda x.M{:}\,\tau \to \mu}(\to \mathrm{I})}{\Gamma \vdash \lambda x.M{:}\,(\sigma \to \mu) \wedge (\tau \to \mu)}(\wedge \mathrm{I})}{\Gamma \vdash \lambda x.M{:}\,\sigma \vee \tau \to \mu}(\leq) \qquad \Gamma \vdash N{:}\,\sigma \vee \tau}{\Gamma \vdash (\lambda x.M)N{:}\,\mu}(\to \mathrm{E})$$

If we compare this derivation with the $(\vee\mathrm{E})$ rule we see that from the same premises we obtain the same type for a β-expansion of the subject.

(iv) In a λ-calculus enriched with constants (and with the corresponding constant types) in the standard way, the typing rules for $+$ and $\|$ give a sort of abstract interpretation [57, 28]. As an example we would have that $1 + \texttt{true}$ has type *integer* $\vee$ *boolean* and $1 \| \texttt{true}$ has both types *integer* and *boolean*.

For the present type assignment system the proof of structural properties is a bit more involved than in case of system $\mathcal{B}$. If $x:\sigma \in \Gamma$, then we define $\Gamma(x) =_{Df} \sigma$.

2.5.8. LEMMA (Structural Properties of Deductions in $\mathcal{C}$).

(i) *If* $\tau \neq \omega$, *then* $\Gamma \vdash x{:}\tau \;\Leftrightarrow\; \Gamma(x) \leq \tau$;

(ii) $\Gamma \vdash \lambda x.M{:}\tau \;\Leftrightarrow$
$$\exists\, \mu_1,\ldots,\mu_n,\nu_1,\ldots,\nu_n.\ \Gamma \vdash \lambda x.M{:}\ \textstyle\bigwedge_{i=1}^{n}(\mu_i \to \nu_i) \text{ and } \bigwedge_{i=1}^{n}(\mu_i \to \nu_i) \leq \tau;$$

(iii) $\Gamma \vdash \lambda x.M{:}\sigma \to \tau \;\Leftrightarrow\; \Gamma, x{:}\sigma \vdash M{:}\tau$;

(iv) $\Gamma \vdash MN{:}\tau \;\Leftrightarrow\; \exists \sigma.\ \Gamma \vdash M{:}\sigma \to \tau$ *and* $\Gamma \vdash N{:}\sigma$;

(v) $\Gamma \vdash M + N{:}\sigma \;\Leftrightarrow\; \Gamma \vdash M{:}\sigma$ *and* $\Gamma \vdash N{:}\sigma$;

(vi) $\Gamma \vdash M \| N{:}\tau \;\Leftrightarrow\; \exists \sigma, \sigma'.\ \sigma \wedge \sigma' \leq \tau$ *and* $\Gamma \vdash M{:}\sigma$ *and* $\Gamma \vdash N{:}\sigma'$.

PROOF. *(i)* and *(iv)*: it is easy to extend to union types the proof given in [18].
In *(ii)*, *(iii)* *(v)* and *(vi)*, $\Leftarrow$ is immediate. We show $\Rightarrow$.

(ii) If $\tau = \omega$ we can take $n = 1$, $\mu_1 = \nu_1 = \omega$, since $\Gamma \vdash \lambda x.M{:}\omega \to \omega$ is provable in $\mathcal{C}$. Otherwise choose a derivation of $\Gamma \vdash \lambda x.M{:}\tau$. Being $\tau \neq \omega$ rule $(\to \mathrm{I})$ has been used. Let
$$\Gamma \vdash \lambda x.M{:}\mu_1 \to \nu_1, \ldots, \Gamma \vdash \lambda x.M{:}\mu_n \to \nu_n$$
be the conclusions of all $(\to \mathrm{I})$ rules having $\lambda x.M$ as subject in this derivation. Now $\lambda x.M$ is the same subject of the conclusion of the derivation itself; hence below such rules only $(\leq)$ and $(\wedge\, \mathrm{I})$ rules are possible. This implies that
$$(\mu_1 \to \nu_1) \wedge \cdots \wedge (\mu_n \to \nu_n) \leq \tau.$$

(iii) Assume $\tau \neq \omega$. Let $\mu_1,\ldots,\mu_n,\nu_1,\ldots,\nu_n$ be as in the proof of *(ii)*. Then by *(ii)* itself:
$$(\mu_1 \to \nu_1) \wedge \cdots \wedge (\mu_n \to \nu_n) \leq \sigma \to \tau$$
so that, by Lemma 2.5.6 *(ii)*,
$$\exists J \subseteq \{1,\cdots,n\}.\ \sigma \leq \bigwedge_{j \in J} \mu_j \text{ and } \bigwedge_{j \in J} \nu_j \leq \tau.$$
On the other hand the premises of the $(\to \mathrm{I})$ rules are of the shape $\Gamma, x{:}\mu_i \vdash M{:}\nu_i$ and have been derived for $1 \leq i \leq n$. Hence $\Gamma, x{:}\sigma \vdash M{:}\tau$.

(v) Let a deduction of $\Gamma \vdash M + N{:}\sigma$ be given and let
$$\Gamma \vdash M + N{:}\sigma_1, \ldots, \Gamma \vdash M + N{:}\sigma_n$$
be all the statements in this deduction on which $\Gamma \vdash M + N{:}\sigma$ depends and which are conclusions of rule $(+\mathrm{I})$. Then
$$\sigma_1 \wedge \cdots \wedge \sigma_n \leq \sigma \text{ and } \Gamma \vdash M{:}\sigma_i, \Gamma \vdash N{:}\sigma_i,$$
for $1 \leq i \leq n$. So we can derive $\Gamma \vdash M{:}\sigma$ and $\Gamma \vdash N{:}\sigma$ using $(\wedge\mathrm{I})$ and $(\leq)$.

(vi) Finally, given a deduction of $\Gamma \vdash M \| N : \tau$, let

$$\Gamma \vdash M \| N : \sigma_1, \ldots, \Gamma \vdash M \| N : \sigma_n$$

be all the statements in this deduction on which $\Gamma \vdash M \| N : \tau$ depends and which are conclusions of rule ($\|$I). Then

$$\sigma_1 \wedge \cdots \wedge \sigma_n \leq \tau \text{ and } \forall i \leq n.\ (\Gamma \vdash M : \sigma_i \text{ or } \Gamma \vdash N : \sigma_i).$$

We assume, without loss of generality, that, for some h, $\Gamma \vdash M : \sigma_i$ for $1 \leq i \leq h$ and $\Gamma \vdash N : \sigma_j$ for $h+1 \leq j \leq n$. It follows that, by rule ($\wedge$I), $\Gamma \vdash M : \sigma$ and $\Gamma \vdash N : \sigma'$ are provable, where $\sigma \equiv \sigma_1 \wedge \ldots \wedge \sigma_h$, $\sigma' \equiv \sigma_{h+1} \wedge \ldots \wedge \sigma_n$ and $\sigma \wedge \sigma' \leq \tau$. □

The invariance of types under subject conversion with respect to $=_a$ is now an easy consequence of the previous Lemmas. We consider $=_a$, since it includes $=$.

2.5.9. Theorem (Subject Conversion of $\mathcal{C}$).

$$\Gamma \vdash M : \sigma \textit{ and } M =_a N \Rightarrow \Gamma \vdash N : \sigma.$$

Proof. It suffices to prove the thesis when $M =_a N$ is replaced by $M \xrightarrow{*}_a N$ (subject reduction) and by $N \xrightarrow{*}_a M$ (subject expansion). We show this by induction on the definition of $\longrightarrow_a$.

The most interesting case is $(P\|Q)L \longrightarrow_a PL\|QL$. Let $\Gamma \vdash (P\|Q)L : \tau$; then we have, by Lemma 2.5.8*(iv)*, that $\Gamma \vdash L : \sigma$ and $\Gamma \vdash P\|Q : \sigma \to \tau$ for some σ. This implies, by Lemma 2.5.8*(vi)*, that there exist μ, ν such that

$$\Gamma \vdash P : \mu, \Gamma \vdash Q : \nu \text{ and } \mu \wedge \nu \leq \sigma \to \tau.$$

Assuming $\mu \neq \omega$ and $\nu \neq \omega$ we have, by Lemma 2.5.6*(i)*,

$$\exists\, \tau_1, \tau_2.\ \tau = \tau_1 \wedge \tau_2 \text{ and } \mu \leq \sigma \to \tau_1 \text{ and } \nu \leq \sigma \to \tau_2.$$

It follows that $\Gamma \vdash P : \sigma \to \tau_1$ and $\Gamma \vdash Q : \sigma \to \tau_2$, so we conclude $\Gamma \vdash PL\|QL : \tau$.
The case in which $\mu = \omega$ or $\nu = \omega$ is similar and simpler.
Vice-versa, let $\Gamma \vdash PL\|QL : \tau$. By Lemma 2.5.8*(vi)* there are μ, ν such that

$$\Gamma \vdash PL : \mu, \Gamma \vdash QL : \nu \text{ and } \mu \wedge \nu \leq \tau.$$

This implies by Lemma 2.5.8*(iv)* that there are σ_1, σ_2 such that

$$\Gamma \vdash P : \sigma_1 \to \mu, \Gamma \vdash L : \sigma_1 \text{ and } \Gamma \vdash Q : \sigma_2 \to \nu, \Gamma \vdash L : \sigma_2.$$

Therefore, by rules ($\|$ I), ($\wedge$I), and ($\leq$)

$$\Gamma \vdash P\|Q : \sigma_1 \wedge \sigma_2 \to \mu \wedge \nu \text{ and } \Gamma \vdash L : \sigma_1 \wedge \sigma_2,$$

so that $\Gamma \vdash (P\|Q)L : \tau$. □

2.5.3. Remark. As an immediate consequence of Theorem 2.5.9, we have the subject conversion of $\mathcal{C}$ also for the relation $=$. Instead, as stated in [31], only subject reduction of $\mathcal{C}$ holds for the reduction $\longrightarrow_{pn}$. This is clear by looking at rule ($+_{\mathcal{C}}$), because this rule properly increases the set of types of the subject.

2.5.3. The λ-lattices

As the set semimodel suggests, when interpreting our calculus we naturally get lattices. We make precise now what is a model of this calculus. We do this by incorporating the notion of lattice into that of λ-model of [54].

2.5.10. DEFINITION. A *λ-lattice* is a structure $\mathcal{D} = \langle D, \sqsubseteq, \cdot, \sqcap, \sqcup, [\![\cdot]\!]^{\mathcal{D}}\rangle$ where:

(i) $\langle D, \sqsubseteq, \sqcap, \sqcup\rangle$ is a lattice;

(ii) $\cdot : D \times D \to D$ is monotonic;

(iii) $\forall d, d', e \in D.\ (d \sqcup d') \cdot e \sqsubseteq (d \cdot e) \sqcup (d' \cdot e)$ and $(d \cdot e) \sqcap (d' \cdot e) \sqsubseteq (d \sqcap d') \cdot e$;

(iv) $[\![\cdot]\!]^{\mathcal{D}} : Env \times \Lambda_{+\|} \to D$, where $Env = \{\rho \mid \rho \colon Var \to D\}$, is such that:

  (a) $[\![M + N]\!]^{\mathcal{D}}_\rho = [\![M]\!]^{\mathcal{D}}_\rho \sqcap [\![N]\!]^{\mathcal{D}}_\rho$;

  (b) $[\![M \| N]\!]^{\mathcal{D}}_\rho = [\![M]\!]^{\mathcal{D}}_\rho \sqcup [\![N]\!]^{\mathcal{D}}_\rho$;

  (c) $[\![x]\!]^{\mathcal{D}}_\rho = \rho(x)$;

  (d) $[\![MN]\!]^{\mathcal{D}}_\rho = [\![M]\!]^{\mathcal{D}}_\rho \cdot [\![N]\!]^{\mathcal{D}}_\rho$;

  (e) $\forall d \in D.\ [\![\lambda x.M]\!]^{\mathcal{D}}_\rho \cdot d = [\![M]\!]^{\mathcal{D}}_{\rho[d/x]}$;

  (f) $\forall x \in \mathrm{FV}(M).\ \rho(x) = \rho'(x) \Rightarrow [\![M]\!]^{\mathcal{D}}_\rho = [\![M]\!]^{\mathcal{D}}_{\rho'}$;

  (g) $(\forall d \in D.\ [\![M]\!]^{\mathcal{D}}_{\rho[d/x]} = [\![N]\!]^{\mathcal{D}}_{\rho[d/x]}) \Rightarrow [\![\lambda x.M]\!]^{\mathcal{D}}_\rho = [\![\lambda x.N]\!]^{\mathcal{D}}_\rho$.

Clauses *(iv)* from (c) to (g) define syntactical λ-models (see [54]). They have been written to state explicitly that the map $[\![\cdot]\!]^{\mathcal{D}}$ satisfies these clauses not just on the classical λ-terms, but on the whole set $\Lambda_{+\|}$.

It is interesting to relate semimodels and λ-lattices considering the role of the order in the structure. Indeed by Proposition 2.4.5 the meaning of a term in a semimodel increases along reduction. In the case of λ-lattices, instead, we have:

2.5.11. PROPOSITION. *$M =_a N \Rightarrow \forall\rho.\ [\![M]\!]^{\mathcal{D}}_\rho = [\![N]\!]^{\mathcal{D}}_\rho$ for all λ-lattices $\mathcal{D}$.*

PROOF. By induction on the definition of $\longrightarrow_a$ using the conditions of 2.5.10*(iii)*. The proof is a straightforward variant of the analogous proof for classical λ-calculus (see [54] or [17] 5.3.4). □

Moreover it is not difficult to show that we have:

$$M \xrightarrow{*}_{pn} N \Rightarrow \forall\rho.\ [\![M]\!]^{\mathcal{D}}_\rho \sqsubseteq [\![N]\!]^{\mathcal{D}}_\rho$$

for all λ-lattices $\mathcal{D}$, where $\sqsubseteq$ can be proper. Indeed $M + N \xrightarrow{*}_{pn} M$ and in general $[\![M+N]\!]^{\mathcal{D}}_\rho \sqsubset [\![M]\!]^{\mathcal{D}}_\rho$.

As immediate consequence of Proposition 2.3.2 we obtain a term model based on the contextual semantics which is a λ-lattice.

2.5.12. PROPOSITION. *For $M, N \in \Lambda_{+\|}$ define $[M] = \{M' \in \Lambda_{+\|} \mid M \simeq^{\mathcal{O}} M'\}$, $[M] \cdot [N] = [MN]$,$[M] \sqcup [N] = [M \| N]$, $[M] \sqcap [N] = [M + N]$, and $[M] \sqsubseteq [N]$ iff $M \sqsubseteq^{\mathcal{O}} N$. These definitions induce a λ-lattice, where $[\![M]\!]_\rho = [M[\vec{N}/\vec{x}]]$ when $FV(M) = \vec{x}$ and $\rho(\vec{x}) = [\vec{N}]$.*

The existence of the term model implies an adequacy result.

2.5.13. COROLLARY. $\forall M, N \in \Lambda_{+\|}.\ (\forall\lambda\text{-}lattice\ \mathcal{D}, \forall\rho.\ [\![M]\!]^{\mathcal{D}}_\rho \sqsubseteq [\![N]\!]^{\mathcal{D}}_\rho) \Rightarrow M \sqsubseteq^{\mathcal{O}} N$.

2.5.4. The Filter λ-lattice

Given the usual notion of *filter*, rules (ω), $(\leq)$ and $(\wedge\mathrm{I})$ imply that, for any Γ and M, $\{\sigma \mid \Gamma \vdash M{:}\sigma\}$ is a filter. A filter model construction as in [18] can be carried out. If X is a subset of any poset, then let $\uparrow X$ be its upward closure.

2.5.14. THEOREM. *Let $\mathcal{F}(Type)$ be the set of filters over Type and define, for $f, f' \in \mathcal{F}(Type)$:*

$$f\bar{\cup}f' = \uparrow\{\sigma \wedge \tau \mid \sigma \in f, \tau \in f'\}, \quad f \cdot f' = \{\tau \mid \exists \sigma \in f'.\, \sigma \to \tau \in f\}.$$

Then $f\bar{\cup}f', f \cdot f' \in \mathcal{F}(Type)$. Moreover the structure

$$\langle \mathcal{F}(Type), \subseteq, \cdot, \cap, \bar{\cup}, [\![\cdot]\!]^{\mathcal{C}} \rangle,$$

where

$$[\![M]\!]^{\mathcal{C}}_{\rho} = \{\sigma \mid \Gamma \vdash M{:}\sigma \textit{ for some } \Gamma \subseteq \{x{:}\tau \mid \tau \in \rho(x)\}\},$$

is a λ-lattice (the filter λ-lattice*).*

PROOF. $f\bar{\cup}f'$ is the least filter including $f \cup f'$, therefore it is the join wrt inclusion in the set of filters. Since filters are closed under intersection, $\langle \mathcal{F}(Type), \cap, \bar{\cup} \rangle$ is a lattice, so that *(i)* of Definition 2.5.10 is satisfied.

It is easy to see that $f \cdot f'$ is a filter too: hence "$\cdot$" is well defined. Moreover "$\cdot$" is clearly monotonic in both its arguments. So that also *(ii)* of Definition 2.5.10 holds.

Now we prove the first clause of *(iii)*. By definition we know

$$\begin{array}{rcl} \tau \in (f_0\bar{\cup}f_1) \cdot f_2 & \Rightarrow & \exists\, \sigma \in f_2.\, \sigma \to \tau \in f_0\bar{\cup}f_1 \\ & \Rightarrow & \exists\, \sigma \in f_2, \mu \in f_0, \nu \in f_1.\, \mu \wedge \nu \leq \sigma \to \tau. \end{array}$$

The more interesting case is $\mu \neq \omega$ and $\nu \neq \omega$. By 2.5.6*(i)* there are τ_1, τ_2 such that $\tau = \tau_1 \wedge \tau_2$ and $\mu \leq \sigma \to \tau_1, \nu \leq \sigma \to \tau_2$. Therefore by definition $\tau_1 \in f_0 \cdot f_2$ and $\tau_2 \in f_1 \cdot f_2$, so we can conclude $\tau \in (f_0 \cdot f_2)\bar{\cup}(f_0 \cdot f_1)$.
The proof of the other clause of *(iii)* is similar and simpler.

Lastly we prove *(iv)*. Lemma 2.5.8*(v)* implies that

$$[\![M + N]\!]^{\mathcal{C}}_{\rho} = [\![M]\!]^{\mathcal{C}}_{\rho} \cap [\![N]\!]^{\mathcal{C}}_{\rho}$$

and Lemma 2.5.8*(vi)* implies that

$$[\![M \| N]\!]^{\mathcal{C}}_{\rho} = [\![M]\!]^{\mathcal{C}}_{\rho}\bar{\cup}[\![N]\!]^{\mathcal{C}}_{\rho}$$

for all ρ. Hence clauses (a) and (b) follow.
The clauses from (c) to (g) follow easily from points $(i), (ii), (iii)$ and *(iv)* of Lemma 2.5.8 as in the case of classical λ-calculus. $\square$

2.5.15. DEFINITION. Let $\mathcal{D} = \langle D, \sqsubseteq, \cdot, \sqcap, \sqcup, [\![\cdot]\!]^{\mathcal{D}} \rangle$ be a λ-lattice. Then a *type structure* over $\mathcal{D}$ is a pair $\langle \mathcal{T}, \Rightarrow \rangle$ such that $\mathcal{T}$ is a sublattice of the lattice of filters over D, $D \in \mathcal{T}$, and $\Rightarrow$ is a binary function over $\mathcal{T}$ such that $X \Rightarrow Y = \{d \in D \mid \forall e \in X.\, d \cdot e \in Y\}$, for all $X, Y \in \mathcal{T}$. Moreover $\mathcal{T}$ is closed under $\cap$, and $\bar{\cup}$ defined by $X\bar{\cup}Y = \uparrow \{d \sqcap d' \mid d \in X, d' \in Y\}$, where we overload $\bar{\cup}$.
The map $[\![\cdot]\!]^{\mathcal{T}}$, interpreting types over $\mathcal{T}$, is defined as in Definition 2.4.8*(iii)*, adding three clauses:

(iii) $[\![\omega]\!]^{\mathcal{T}}_{\eta} = D$;

(iv) $[\![\sigma \wedge \tau]\!]^{\mathcal{T}}_{\eta} = [\![\sigma]\!]^{\mathcal{T}}_{\eta} \cap [\![\tau]\!]^{\mathcal{T}}_{\eta}$;

(v) $[\![\sigma \vee \tau]\!]^{\mathcal{T}}_{\eta} = [\![\sigma]\!]^{\mathcal{T}}_{\eta} \bar{\cup} [\![\tau]\!]^{\mathcal{T}}_{\eta}$.

In the filter λ-lattice defined in Theorem 2.5.14, the interpretation of a type turns out to be a filter of filters of types. Since the lattice of types is distributive, the lattice of filters forming the filter λ-lattice is distributive too, hence the upward closure in clause *(v)* above is redundant in this case. The following proposition is proved by routine calculations.

2.5.16. PROPOSITION. *Let* $\chi_{\sigma} = \{f \in \mathcal{F}(Type) \mid \sigma \in f\}$. *Then* $\langle\{\chi_{\sigma} \mid \sigma \in Type\}, \Rightarrow\rangle$ *is a type structure over the filter λ-lattice. Moreover it satisfies the following equations:*

(i) $\chi_{\omega} = \mathcal{F}(Type)$;

(ii) $\chi_{\sigma\to\tau} = \chi_{\sigma} \Rightarrow \chi_{\tau}$;

(iii) $\chi_{\sigma\wedge\tau} = \chi_{\sigma} \cap \chi_{\tau}$;

(iv) $\chi_{\sigma\vee\tau} = \chi_{\sigma} \bar{\cup} \chi_{\tau} = \{f \cap f' \mid f \in \chi_{\sigma}, f' \in \chi_{\tau}\}$.

As for system $\mathcal{B}$, the immediate consequence of Theorem 2.5.14 and of Proposition 2.5.16 is completeness. Redefining $\models$ for λ-lattices in the same way as it has been defined for semimodels in 2.4.7, this is stated as follows.

2.5.17. COROLLARY (Completeness of $\mathcal{C}$). $\Gamma \vdash M{:}\sigma \;\Leftrightarrow\; \Gamma \models M{:}\sigma$.

The filter λ-lattice naturally induces a preorder on terms.

2.5.18. DEFINITION. $M \sqsubseteq^{\mathcal{C}} N \Leftrightarrow_{def} \forall\rho.\, [\![M]\!]^{\mathcal{C}}_{\rho} \subseteq [\![N]\!]^{\mathcal{C}}_{\rho}$.

We state some (in)-equations which show that $\sqsubseteq^{\mathcal{C}}$ discriminates terms which are equated by $\sqsubseteq^{\mathcal{O}}$. This implies that the filter λ-lattice is not fully abstract with respect to the contextual semantics.

2.5.19. PROPOSITION. *The following (in)-equations hold:*

(i) $(\lambda x.M)N \simeq^{\mathcal{C}} M[N/x]$;

(ii) $(M+N)L \simeq^{\mathcal{C}} ML + NL$;

(iii) $L(M+N) \sqsubseteq^{\mathcal{C}} LM + LN$;

(iv) $(M\|N)L \simeq^{\mathcal{C}} ML\|NL$;

(v) $LM\|LN \sqsubseteq^{\mathcal{C}} L(M\|N)$;

(vi) $\lambda x.(M+N) \sqsubseteq^{\mathcal{C}} \lambda x.M + \lambda x.N$;

(vii) $\lambda x.(M\|N) \simeq^{\mathcal{C}} \lambda x.M\|\lambda x.N$;

(viii) $M + N \sqsubseteq^{\mathcal{C}} M, N$;

(ix) $L \sqsubseteq^{\mathcal{C}} M, N \Rightarrow L \sqsubseteq^{\mathcal{C}} M + N$;

(x) $M, N \sqsubseteq^{\mathcal{C}} M\|N$;

(xi) $M, N \sqsubseteq^{\mathcal{C}} L \Rightarrow M\|N \sqsubseteq^{\mathcal{C}} L$.

where the inequalities (iii), (v) and (vi) are in general proper.

PROOF. Points $(i), (ii), (iv), (viii), (ix), (x)$ and (xi) hold by definition of λ-lattice. For the other points, the positive statements are easy consequences of Lemma 2.5.8.

The examples given in the proof of Proposition 2.3.2 show that the inequalities *(iii)* and *(v)* are proper. Indeed we have that both $\Delta M + \Delta N$ and $(T+R)(\mathbf{I}\|\mathbf{K})$ have type $\sigma \wedge (\sigma \to \tau) \to \tau$. On the contrary, ω is the only type which can be deduced for $\Delta(M+N)$ and for $(T+R)\mathbf{I}\|(T+R)\mathbf{K}$.

To prove that the inequality *(vi)* is proper we have for example $\vdash \lambda x.x + \lambda x.xx : (\mu \to \mu) \vee (\sigma \wedge (\sigma \to \tau) \to \tau)$, but this type cannot be deduced for $\lambda x.(x + xx)$. □

Notice that the filter model turns out to be a (properly) semilinear applicative structure as defined in [64, 65], because of 2.5.19*(ii)* and *(iii)*. This was not true for the set semimodel. It is worth to stress that, without the union type constructor, this cannot be achieved (see [1]). From this fact and from Proposition 2.4.10 it is also clear that the theories induced by $\simeq^{\mathcal{B}}$ and $\simeq^{\mathcal{C}}$ are incomparable.

2.6. Approximation Theorem and Full Abstraction

In this section we prove the main results of the present chapter, i.e.:

- the filter λ-lattice is adequate with respect to the contextual semantics;
- the filter λ-lattice is fully abstract with respect to the capabilities semantics.

A main tool in these proofs is the notion of approximant. The first result essentially follows from the Approximation Theorem for the filter λ-lattice. For the second result we introduce a one-to-one correspondence between approximate normal forms (considered modulo $\sim$) and suitable pairs $<$ *basis, type* $>$ (where types are considered modulo $=$). This correspondence essentially shows that the discrimination power of approximants and types is the same.

2.6.1. The Approximation Theorem and The Adequacy for the Contextual Semantics

In this section we prove that the set of types which can be deduced for any term coincides with the union of the sets of types deducible for its approximants. Since in the filter λ-lattice these sets are thought of as the "meanings" of terms, this shows that the meaning of any term is the join of the meanings of its approximants.

Let us call $\mathcal{C}\Omega$ the type system resulting from $\mathcal{C}$ when subjects are from $\Lambda_{+\|\Omega}$. Since no explicit typing rule is added for the constant Ω, if $\Gamma \vdash \Omega{:}\,\sigma$, then $\sigma = \omega$. vice-versa, a straightforward induction shows that, if A is an approximate normal form and $A \not\sim \Omega$, then there are a basis Γ and a type $\sigma \neq \omega$, such that $\Gamma \vdash A{:}\,\sigma$. All the properties of the system $\mathcal{C}$ proved in previous section extends easily to $\mathcal{C}\Omega$. So we will freely use them in the following proofs.

The Approximation Theorem is proved by means of a variant of Tait's "computability" technique. We define sets of "approximable" and "computable" terms (Definition 2.6.1). The computable terms are defined by induction on types, and every computable term is shown to be approximable (Lemma 2.6.4*(ii)*). Using induction on typings, we show that every term is computable for the appropriate type (Lemma 2.6.7).

2.6.1. Definition. We define two predicates $\mathsf{App}(\Gamma, \sigma, M)$ and $\mathsf{Comp}(\Gamma, \sigma, M)$ as follows:

(i) $\mathsf{App}(\Gamma, \sigma, M) \;\Leftrightarrow\; \exists A \in \mathcal{A}(M).\ \Gamma \vdash A{:}\,\sigma$;

(ii) *(a)* $\mathsf{Comp}(\Gamma, \omega, M)$ is always true;

(b) $\mathsf{Comp}(\Gamma, t, M) \;\Leftrightarrow\; \mathsf{App}(\Gamma, t, M)$;

(c) $\mathsf{Comp}(\Gamma, \sigma \to \tau, M) \;\Leftrightarrow\; \forall \Gamma', N.\mathsf{Comp}(\Gamma', \sigma, N) \Rightarrow \mathsf{Comp}(\Gamma \uplus \Gamma', \tau, MN)$;

(d) $\mathsf{Comp}(\Gamma, \sigma \wedge \tau, M) \;\Leftrightarrow\; \mathsf{Comp}(\Gamma, \sigma, M)$ and $\mathsf{Comp}(\Gamma, \tau, M)$;

(e) $\mathsf{Comp}(\Gamma, \sigma \vee \tau, M) \;\Leftrightarrow\; \mathsf{App}(\Gamma, \sigma \vee \tau, M)$.

We can easily prove that Comp agrees with some head reductions. More precisely we have:

2.6.2. LEMMA. *Let M be a redex and N its immediate contractum. Then, for any Γ, σ,*

$$\mathsf{Comp}(\Gamma, \sigma, N\vec{L}) \;\Rightarrow\; \mathsf{Comp}(\Gamma, \sigma, M\vec{L})$$

where $\vec{L}$ is any vector of terms.

PROOF. The proof is by induction on σ.
If $\sigma \equiv t$ or $\sigma \equiv \sigma_1 \vee \sigma_2$ the thesis follows immediately from 2.3.11*(iv)* since the hypothesis on M and N implies $M\vec{L} \xrightarrow{*}{}^{h} N\vec{L}$, so that $\mathcal{A}(N\vec{L}) \subseteq \mathcal{A}(M\vec{L})$.
If $\sigma \equiv \sigma_1 \wedge \sigma_2$ the thesis follows by induction.
If $\sigma \equiv \sigma_1 \to \sigma_2$, let P be such that $\mathsf{Comp}(\Gamma', \sigma_1, P)$ so that by definition $\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, N\vec{L}P)$. This implies by induction $\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, M\vec{L}P)$, so we can conclude $\mathsf{Comp}(\Gamma, \sigma, M\vec{L})$, by the arbitrariness of the term P. $\square$

Really Comp is invariant under $=_a$, but we do not prove this, since we would need $M =_a N \Rightarrow \mathcal{A}(M) = \mathcal{A}(N)$ (see Remark 2.3.2).

We show some properties of types which are deducible for approximate normal forms.

2.6.3. LEMMA. *Let $A, A' \in \mathcal{A}$.*

(i) $\Gamma \vdash A{:}\sigma$ *and* $A \preceq A' \Rightarrow \Gamma \vdash A'{:}\sigma$.

(ii) *Let $z \notin FV(M)$ and suppose that z does not occur in the basis Γ.*
If $A \in \mathcal{A}(Mz)$, then

$$\Gamma, z{:}\sigma \vdash A{:}\tau \Rightarrow \exists \widehat{A} \in \mathcal{A}(M).\ \Gamma \vdash \widehat{A}{:}\sigma \to \tau.$$

PROOF. *(i)* By induction on $\preceq$. The more interesting case is $A \equiv \lambda y.xA_1 \dots A_n y$ and $A' \equiv xA_1 \dots A_n$, where $y \notin FV(xA_1 \dots A_n)$. By 2.5.8*(ii)* $\Gamma \vdash A{:}\sigma$ implies $\Gamma \vdash A{:} \bigwedge_{i=1}^{m}(\mu_i \to \nu_i)$, for some $\mu_1, \dots, \mu_m, \nu_1, \dots, \nu_m$ such that $\bigwedge_{i=1}^{m}(\mu_i \to \nu_i) \leq \sigma$. From $\Gamma \vdash A{:}\mu_i \to \nu_i$ by 2.5.8*(iii)* we have $\Gamma, y{:}\mu_i \vdash xA_1 \dots A_n y{:}\nu_i$. Therefore by 2.5.8*(iv)* and 2.5.8*(i)* $\Gamma, y{:}\mu_i \vdash xA_1 \dots A_n{:}\mu_i \to \nu_i$ hold for $1 \leq i \leq m$. Since $y \notin FV(xA_1 \dots A_n)$, we can prove using $(\wedge \mathrm{I})$ and $(\leq)$ that $\Gamma \vdash A'{:}\sigma$.

(ii) $\mathcal{A}(M)$ is the downward closure of

$$\mathcal{A}'(M) = \{\phi(H) \mid H \in \mathcal{C}(M)\}$$

with respect to $\preceq$. It follows that, by *(i)*, it suffices to show the thesis when $A \in \mathcal{A}'(Mz)$. If $A \in \mathcal{A}'(Mz)$ then, for some H, H',

$$A \equiv \phi(H') \text{ and } Mz \xrightarrow{*}{}^{h} H \xrightarrow{*}{}_{a} H'.$$

The proof is by induction on the length k of the reduction $\xrightarrow{*}{}^{h}$. If $k = 0$ then $Mz \equiv H \equiv xM_1 \dots M_n z$. Hence $H' \equiv xM_1' \dots M_n' z$ where $M_i' \xrightarrow{*}{}_{a} M_i$ for $i \leq n$. Therefore $A \equiv x\phi(M_1') \dots \phi(M_n')z$, so that we take $\widehat{A} \equiv x\phi(M_1') \dots \phi(M_n') \in \mathcal{A}'(M)$. We have $\Gamma \vdash \widehat{A}{:}\sigma \to \tau$ using 2.5.8*(i)* and *(iv)*.
If $k > 0$, then

$$Mz \xrightarrow{*}{}^{h} M'z \longrightarrow^{h} L \xrightarrow{*}{}^{h} H \xrightarrow{*}{}_{a} H'$$

where $M \xrightarrow{*}^{h} M'$ and M', L have one of the following shapes:

(a) $M' \equiv \lambda x.P$ and $L \equiv P[z/x]$;
(b) $M' \equiv P + Q$ and $L \equiv Pz + Qz$;
(c) $M' \equiv P \| Q$ and $L \equiv Pz \| Qz$.

Case (a). Then $A \in \mathcal{A}'(P[z/x])$, which implies $\lambda z.A \in \mathcal{A}'(\lambda z.P[z/x])$. Now $\lambda z.P[z/x] \equiv \lambda x.P$ since by hypothesis $z \notin FV(P)$. From $\Gamma, z{:}\sigma \vdash A{:}\tau$ we derive by ($\to$ I) $\Gamma \vdash \lambda z.A{:}\sigma \to \tau$. So we can choose $\widehat{A} \equiv \lambda z.A$, since $\mathcal{A}'(\lambda x.P) \subseteq \mathcal{A}'(M)$ by 2.3.11*(iv)*.

Case (b). In this case $H \equiv H_1 + H_2$, $H' \equiv H_1' + H_2'$, and $Pz \xrightarrow{*}^{h} H_1 \xrightarrow{*}_{a} H_1'$, $Qz \xrightarrow{*}^{h} H_2 \xrightarrow{*}_{a} H_2'$. Moreover

$$A \equiv \phi(H_1') + \phi(H_2'),$$

where $\phi(H_1') \in \mathcal{A}'(Pz)$ and $\phi(H_2') \in \mathcal{A}'(Qz)$. Now $\Gamma, z{:}\sigma \vdash A{:}\tau$ implies, by Lemma 2.5.8*(v)*, $\Gamma, z{:}\sigma \vdash \phi(H_i'){:}\tau$ for $i = 1, 2$. Notice that the lengths of the reductions $Pz \xrightarrow{*}^{h} H_1, Qz \xrightarrow{*}^{h} H_2$ are lower than k. Then by induction there are $A_1 \in \mathcal{A}'(P)$ and $A_2 \in \mathcal{A}'(Q)$ such that $\Gamma \vdash A_i{:}\sigma \to \tau$, for $i = 1, 2$ hence

$$\Gamma \vdash A_1 + A_2{:}\sigma \to \tau.$$

Therefore we can choose $\widehat{A} \equiv A_1 + A_2$; in fact $\widehat{A} \in \mathcal{A}'(M)$, since $M \xrightarrow{*}^{h} P + Q$.

Case (c). Similar to case (b), using Lemma 2.5.8*(vi)* and $M \xrightarrow{*}^{h} P \| Q$. □

We can now show that computability implies approximability.

2.6.4. LEMMA. *For all* Γ, σ, $\vec{L}$ *and* M:

(i) $\mathsf{App}(\Gamma, \sigma, x\vec{L}) \Rightarrow \mathsf{Comp}(\Gamma, \sigma, x\vec{L})$;

(ii) $\mathsf{Comp}(\Gamma, \sigma, M) \Rightarrow \mathsf{App}(\Gamma, \sigma, M)$.

PROOF. (i) and (ii) can be simultaneously proved by induction on σ. We show (ii) in the case $\sigma \equiv \sigma_1 \to \sigma_2$, only.

Let $\Gamma' = \Gamma, z{:}\sigma_1$ where $z \notin FV(M)$ and suppose $\mathsf{Comp}(\Gamma, \sigma_1 \to \sigma_2, M)$; then

$$\begin{array}{lll} & \mathsf{Comp}(\{z{:}\sigma_1\}, \sigma_1, z) & \text{by } (i) \\ \Rightarrow & \mathsf{Comp}(\Gamma', \sigma_2, Mz) & \text{by definition} \\ \Rightarrow & \mathsf{App}(\Gamma', \sigma_2, Mz) & \text{by induction} \\ \Rightarrow & \exists A \in \mathcal{A}(M).\, \Gamma \vdash A{:}\sigma_1 \to \sigma_2 & \text{by Lemma 2.6.3}(ii). \end{array}$$

□

The following two Lemmas state that computability agrees with the typing rules ($\leq$), (+ I) and ($\|$ I).

2.6.5. LEMMA. *For all* σ *and* τ:

(i) $\sigma \leq \tau \Rightarrow \forall \Gamma, M.\, \mathsf{App}(\Gamma, \sigma, M) \Rightarrow \mathsf{App}(\Gamma, \tau, M)$;

(ii) $\sigma \leq \tau \Rightarrow \forall \Gamma, M.\ \mathsf{Comp}(\Gamma, \sigma, M) \Rightarrow \mathsf{Comp}(\Gamma, \tau, M)$.

PROOF. If $A \in \mathcal{A}(M)$ is such that $\Gamma \vdash A{:}\sigma$ then by rule $(\leq)$ $\Gamma \vdash A{:}\tau$, hence (i).

(ii) is easily proved, using *(i)* and Lemma 2.6.4*(ii)*, by induction on any standard axiomatic presentation of $\leq$. In particular, for the basic case $\sigma \leq \sigma \vee \tau$ we have:

$$\begin{array}{rcll} \mathsf{Comp}(\Gamma, \sigma, M) & \Rightarrow & \mathsf{App}(\Gamma, \sigma, M) & \text{by Lemma 2.6.4}\textit{(ii)} \\ & \Rightarrow & \mathsf{App}(\Gamma, \sigma \vee \tau, M) & \text{by } \textit{(i)} \\ & \Rightarrow & \mathsf{Comp}(\Gamma, \sigma \vee \tau, M) & \text{by definition.} \qquad \square \end{array}$$

2.6.6. LEMMA. *For all* Γ, σ, τ *and terms* M, N*:*

(i) $\mathsf{Comp}(\Gamma, \sigma, M)$ *and* $\mathsf{Comp}(\Gamma, \tau, N) \Rightarrow \mathsf{Comp}(\Gamma, \sigma \vee \tau, M + N)$*;*

(ii) $\mathsf{Comp}(\Gamma, \sigma, M) \Rightarrow \mathsf{Comp}(\Gamma, \sigma, M \| N)$.

PROOF. *(i)* By Lemma 2.6.4*(ii)*, the hypothesis implies $\mathsf{App}(\Gamma, \sigma, M)$ and $\mathsf{App}(\Gamma, \tau, N)$, that is, for some $A \in \mathcal{A}(M)$ and $A' \in \mathcal{A}(N)$, $\Gamma \vdash A{:}\sigma$ and $\Gamma \vdash A'{:}\tau$. This implies, by rules $(\leq)$ and $(+\ \mathrm{I})$, that $\Gamma \vdash A + A'{:}\sigma \vee \tau$. Since $A + A' \in \mathcal{A}(M + N)$, it follows that $\mathsf{App}(\Gamma, \sigma \vee \tau, M + N)$, hence the thesis by definition.

(ii) By induction on σ. If σ has the shape t or $\sigma_1 \vee \sigma_2$, then $\mathsf{Comp}(\Gamma, \sigma, M)$ implies (by definition) $\mathsf{App}(\Gamma, \sigma, M)$, that is, for some $A \in \mathcal{A}(M)$, $\Gamma \vdash A{:}\sigma$. Hence, by rule $(\|\ \mathrm{I})$, $\Gamma \vdash A \| \Omega{:}\sigma$. Since $A \| \Omega \in \mathcal{A}(M \| N)$ for any N, we conclude that $\mathsf{App}(\Gamma, \sigma, M \| N)$ holds. This implies the thesis.
If $\sigma \equiv \sigma_1 \wedge \sigma_2$, then the thesis is an immediate consequence of the induction hypothesis.
Finally, if $\sigma \equiv \sigma_1 \to \sigma_2$, let P be any term such that $\mathsf{Comp}(\Gamma', \sigma_1, P)$, so that by definition $\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, MP)$. By induction,

$$\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, MP \| Q),$$

for any Q, hence for any N we can take $Q \equiv NP$ so that

$$\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, MP \| NP).$$

Lemma 2.6.2 implies $\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, (M \| N)P)$, so we can conclude:

$$\mathsf{Comp}(\Gamma, \sigma_1 \to \sigma_2, M \| N). \qquad \square$$

2.6.7. LEMMA. *Let* $\Gamma = \{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\}$ *and* $\Gamma \vdash M{:}\tau$.
Assume that, for each $i \leq n$, $\mathsf{Comp}(\Gamma_i, \sigma_i, N_i)$*; then, taking* $\Gamma' = \biguplus_{i=1}^{n} \Gamma_i$*,*

$$\mathsf{Comp}(\Gamma', \tau, M[N_1/x_1, \ldots, N_n/x_n]).$$

PROOF. By induction on the derivation of $\Gamma \vdash M{:}\tau$.
Cases (Ax) and (ω) are immediate.
Cases $(\to\ \mathrm{E})$ and $(\wedge\,\mathrm{I})$ follow by induction. Cases $(+\ \mathrm{I})$ and $(\leq)$ follow from the induction hypothesis and Lemmas 2.6.6*(i)* and 2.6.5*(ii)* respectively.
If we are in case $(\|\ \mathrm{I})$, then $M \equiv P \| Q$ for some P and Q and, say, $\Gamma \vdash P{:}\tau$ has been derived. From the induction hypothesis, $\mathsf{Comp}(\Gamma', \tau, P[\vec{N}/\vec{x}])$, so that by Lemma 2.6.6*(ii)*,

$$\mathsf{Comp}(\Gamma', \tau, P[\vec{N}/\vec{x}] \| Q[\vec{N}/\vec{x}]),$$

i.e. $\mathsf{Comp}(\Gamma', \tau, (P\|Q)[\vec{N}/\vec{x}])$.
Finally, in case ($\rightarrow$ I) suppose that $M \equiv \lambda y.P$, $\tau \equiv \tau_1 \rightarrow \tau_2$ and $\Gamma, y{:}\tau_1 \vdash P{:}\tau_2$ has been derived. Now, if $\mathsf{Comp}(\Gamma'', \tau_1, Q)$, from the induction hypothesis

$$\mathsf{Comp}(\Gamma' \uplus \Gamma'', \tau_2, P[Q/y, \vec{N}/\vec{x}]).$$

There is no theoretical loss in assuming that $y \notin FV(\vec{N})$ so that

$$P[Q/y, \vec{N}/\vec{x}] \equiv P[\vec{N}/\vec{x}][Q/y] \text{ and } (\lambda y.P[\vec{N}/\vec{x}])Q \equiv ((\lambda y.P)[\vec{N}/\vec{x}])Q.$$

By 2.6.2, it follows that $\mathsf{Comp}(\Gamma' \uplus \Gamma'', \tau_2, ((\lambda y.P)[\vec{N}/\vec{x}])Q)$, and hence

$$\mathsf{Comp}(\Gamma', \tau_1 \rightarrow \tau_2, (\lambda y.P)[\vec{N}/\vec{x}])$$

being the computable term Q arbitrary. □

2.6.8. Theorem (Approximation Theorem). *For any term M, basis Γ and type σ:*

$$\Gamma \vdash M{:}\sigma \Leftrightarrow \exists A \in \mathcal{A}(M).\, \Gamma \vdash A{:}\sigma.$$

Proof. ($\Rightarrow$) Since, for any variable x and type τ, $\mathsf{App}(\{x{:}\tau\}, \tau, x)$ holds, then by Lemma 2.6.4*(i)*, $\mathsf{Comp}(\{x{:}\tau\}, \tau, x)$ holds. Taking in Lemma 2.6.7 the identical substitution, the hypothesis implies $\mathsf{Comp}(\Gamma, \sigma, M)$, and the thesis follows from Lemma 2.6.4*(ii)*.
($\Leftarrow$) Easy from subject conversion (Theorem 2.5.9) and the definition of $\mathcal{A}$. □

From the Approximation Theorem it follows that any term which is typable with a type $\neq \omega$ has an approximant which differs from Ω, i.e. it is solvable. Vice-versa, by Proposition 2.3.12*(ii)* any solvable term has an approximant different from Ω and therefore it can be typed with a type $\neq \omega$.

2.6.9. Corollary.

$$\mathsf{SOL} = \{M \in \Lambda_{+\|} \mid \exists \Gamma, \sigma \neq \omega.\, \Gamma \vdash M{:}\sigma\}.$$

The Approximation Theorem is useful to state properties of the precongruence induced on terms by the filter λ-lattice. In fact we immediately have that the filter λ-lattice is adequate with respect to the observational semantics based on contexts.

2.6.10. Theorem (First Adequacy Theorem). *The filter λ-lattice is adequate for the contextual theory based on solvability, i.e.:*

$$M \sqsubseteq^{\mathcal{C}} N \Rightarrow M \sqsubseteq^{\mathcal{O}} N.$$

Proof. Since $\sqsubseteq^{\mathcal{C}}$ is a precongruence, we immediately have that

$$M \sqsubseteq^{\mathcal{C}} N \Rightarrow \forall C[\,].\, C[M] \sqsubseteq^{\mathcal{C}} C[N].$$

It follows that, by Corollary 2.6.9,

$$\begin{array}{rcl} C[M] \in \mathsf{SOL} & \Rightarrow & \exists \Gamma, \sigma \neq \omega.\, \Gamma \vdash C[M]{:}\sigma \\ & \Rightarrow & \exists \Gamma, \sigma \neq \omega.\, \Gamma \vdash C[N]{:}\sigma \\ & \Rightarrow & C[N] \in \mathsf{SOL}. \quad \square \end{array}$$

2.6.2. Principal Pairs and Full Abstraction for the Capability Semantics

To prove adequacy for the semantics based on capabilities and approximants, a suitable extension of the notion of principal type scheme (as given in [26, 86, 14]) is in order. Since we need to consider open terms, we introduce the notion of *principal pair* consisting of a type and a basis. Such a notion is based on a stratification of the set of approximate normal forms, to be compared with the stratification of *Type* introduced in Definition 2.5.2.

2.6.11. DEFINITION (Stratification of $\mathcal{A}$). Let us define three subsets $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ of $\mathcal{A}$ recursively:

- $\Omega \in \mathcal{A}_2$;
- $A \in \mathcal{A}_1 \Rightarrow \lambda x.A \in \mathcal{A}_0$;
- $m \geq 0, A_1, \ldots, A_m \in \mathcal{A}_2 \Rightarrow xA_1 \ldots A_m \in \mathcal{A}_0$ (the λ-free approximate normal forms);
- $n \geq 1, A_1, \ldots, A_n \in \mathcal{A}_0 \Rightarrow A_1 + \ldots + A_n \in \mathcal{A}_1$;
- $n \geq 1, A_1, \ldots, A_n \in \mathcal{A}_1 \Rightarrow A_1 \| \ldots \| A_n \in \mathcal{A}_2$.

Taking $n = 1$ in the clauses above, one sees that $\mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \mathcal{A}$, and such inclusions are clearly proper. For example, we have that $x + y \in \mathcal{A}_1$, but $x + y \notin \mathcal{A}_0$, $x\|y \in \mathcal{A}_2$, but $x\|y \notin \mathcal{A}_1$, $\lambda x.(x\|y) \in \mathcal{A}$, but $\lambda x.(x\|y) \notin \mathcal{A}_2$. Over each of these sets we introduce a preorder.

2.6.12. DEFINITION. $\preceq_i \subseteq \mathcal{A}_i \times \mathcal{A}_i$ is the least preorder such that:

($\preceq_0$) $A \preceq_0 A'$ if and only if one of the following holds:

- $A \equiv \lambda x.B, A' \equiv \lambda x.B'$ and $B \preceq_1 B'$;
- $A \equiv xB_1 \ldots B_n, A' \equiv xB'_1 \ldots B'_n$ and $\forall i \leq n. B_i \preceq_2 B'_i$;
- A' is λ-free, $x \notin FV(A')$ and $A \preceq_0 \lambda x.A'x$.

($\preceq_1$) $A_1 + \ldots + A_n \preceq_1 B_1 + \ldots + B_m \;\Leftrightarrow\; \forall j \leq m \;\exists i \leq n.\; A_i \preceq_0 B_j$.

($\preceq_2$) $A \preceq_2 A'$ if and only if one of the following holds:

- $A \equiv \Omega$;
- $A \equiv B_1 \| \ldots \| B_n, A' \equiv B'_1 \| \ldots \| B'_m$ and $\forall i \leq n \;\exists j \leq m.\; B_i \preceq_1 B'_j$.

As in the case of types, for each approximate normal form we can find an equivalent element of $\mathcal{A}_2$. The following definition has to be compared with 2.5.4.

Notation. In writing $A^* \equiv \|_{i \in I} A_i$ we assume that $A_i \in \mathcal{A}_1$ for all $i \in I$.

2.6.13. DEFINITION. Let $(\;)^* : \mathcal{A} \to \mathcal{A}_2$ be defined by:

- $\Omega^* = \Omega$
- $(xA_1 \ldots A_n)^* = xA_1^* \ldots A_n^* \qquad (n \geq 0)$
- $(\lambda x.A)^* = \begin{cases} \lambda x.A_1 \| \ldots \| \lambda x.A_n & \text{if } A^* = A_1 \| \ldots \| A_n \text{ and } A^* \not\equiv \Omega \qquad (n \geq 1) \\ \Omega & \text{otherwise} \end{cases}$
- $(A + A')^* = \begin{cases} \|_{i \in I,\, j \in J}(B_i + B'_j) & \text{if } A^* = \|_{i \in I} B_i,\ A^* \not\equiv \Omega \text{ and } A'^* = \|_{j \in J} B'_j,\ A'^* \not\equiv \Omega \\ \Omega & \text{otherwise} \end{cases}$
- $(A\|A')^* = \begin{cases} A^* & \text{if } A'^* \equiv \Omega \\ A'^* & \text{if } A^* \equiv \Omega \\ A^*\|A'^* & \text{otherwise.} \end{cases}$

For example, $(\lambda x.(x\|y) + z\|\Omega)^* = (\lambda x.x + z)\|(\lambda x.y + z)$.

The proof of the following proposition is analogous to the proof of Proposition 2.5.5.

2.6.14. PROPOSITION. *For all $A, A' \in \mathcal{A}$:*

(i) $A \sim A^*$;

(ii) $A, A' \in \mathcal{A}_i, A \preceq_i A' \Rightarrow A \preceq A'$ *for* $i = 0, 1, 2$;

(iii) $A \preceq A' \Rightarrow A^* \preceq_2 A'^*$.

The following definition of principal pair is a generalization to our calculus of that one given in [51], [26], [86], and [14], where it was used to prove the principal type property for various intersection type disciplines.

Let $Basis$ be the set of all bases and $TV(<\Gamma;\sigma>)$ be the set of type variables which occur in Γ or in σ.

2.6.15. DEFINITION.

(i) The mapping $pp : \mathcal{A}_2 \rightarrow Basis \times T_2$ is inductively defined by:

(a) $pp(\Omega) = <\emptyset;\omega>$;

(b) if $pp(A_i) = <\Gamma_i;\sigma_i>$, $TV(<\Gamma_i;\sigma_i>) \cap TV(<\Gamma_j;\sigma_j>) = \emptyset$ for $1 \leq i \neq j \leq n$ and t is fresh, then

$$pp(xA_1 \ldots A_n) = <(\biguplus_{i\leq n} \Gamma_i) \uplus \{x{:}\sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow t\};t> \ (n \geq 0);$$

(c) if $pp(A) = <\Gamma, x:\tau;\sigma>$, then

$$pp(\lambda x.A) = <\Gamma;\tau \rightarrow \sigma>;$$

(d) if $pp(A) = <\Gamma;\sigma>$ and $x \notin FV(\Gamma)$, then

$$pp(\lambda x.A) = <\Gamma;\omega \rightarrow \sigma>;$$

(e) if $pp(A_i) = <\Gamma_i;\sigma_i>$ $(i = 1, 2)$ and $TV(<\Gamma_1;\sigma_1>) \cap TV(<\Gamma_2;\sigma_2>) = \emptyset$, then

$$pp(A_1 + A_2) = <\Gamma_1 \uplus \Gamma_2;\sigma_1 \vee \sigma_2>;$$

(f) if $pp(A_i) = <\Gamma_i;\sigma_i>$ $(i = 1, 2)$ and $TV(<\Gamma_1;\sigma_1>) \cap TV(<\Gamma_2;\sigma_2>) = \emptyset$, then

$$pp(A_1\|A_2) = <\Gamma_1 \uplus \Gamma_2;\sigma_1 \wedge \sigma_2>.$$

(ii) The set Π of principal pairs is the range of the mapping pp.

(iii) A type σ is *principal* iff $<\Gamma,\sigma> \in \Pi$ for some basis Γ. A basis Γ is *principal* iff $<\Gamma,\sigma> \in \Pi$ for some type σ.

To build a unique principal pair, in clause 2.6.15*(i) (b)* we assume to pick up fresh type variables in a deterministic way.
For example we have

$$pp(xyy + (\lambda z.y \| \lambda z.z)) =< x : t_1 \to t_2 \to t_3, y : t_1 \wedge t_2 \wedge t_4 \ ; \ t_3 \vee ((\omega \to t_4) \wedge (t_5 \to t_5)) > .$$

From the definition it follows immediately that the principal pair of an approximate normal form can be deduced for it. Moreover it is easy to prove that the mapping *pp* agrees with the stratification of types and approximate normal forms.

2.6.16. PROPOSITION. *If $pp(A) =< \Gamma; \sigma >$ then $\Gamma \vdash A : \sigma$ and $A \in \mathcal{A}_i$ iff $\sigma \in T_i$ where $i = 0, 1, 2$.*

$\diamond$ turns out to be a very restricted set with closure properties which follow easily from its definition.

2.6.17. PROPOSITION. *Let $< \Gamma; \sigma > \in \diamond$.*

(i) *Each type variable occurs exactly twice in Γ, σ.*

(ii) *All types which occur in a principal basis belong to T_2. Moreover they are intersections of arrow types belonging to T_0 and terminating with a type variable.*

(iii) *If $x{:}\tau_1 \to \ldots \to \tau_n \to \mu \in \Gamma$, then for all $1 \leq i \leq n$ there is $\Gamma_i \Subset \Gamma$ such that $< \Gamma_i; \tau_i > \in \diamond$.*

(iv) *If $\sigma \equiv \mu \to \tau$, then $< \Gamma, x : \mu \ ; \tau > \in \diamond$ for all variables x.*

(v) *If $\sigma \equiv \sigma_1 \vee \sigma_2$ or $\sigma \equiv \sigma_1 \wedge \sigma_2$, then there are $\Gamma_1, \Gamma_2 \Subset \Gamma$ such that $< \Gamma_i; \sigma_i > \in \diamond$ $(i = 1, 2)$.*

The types which can be deduced for a variable from a principal basis are of limited shape.

2.6.18. LEMMA. *Let Γ be a principal basis.*

(i) *If $\tau \in T_1$ and $\Gamma \vdash x{:}\tau$, then $\Gamma(x) = \mu \wedge \nu$ for some μ, ν such that $\mu \in T_0$ and $\mu \leq_1 \tau$.*

(ii) *If $\mu \vee \nu \in T_1$ and $\Gamma \vdash x{:}\tau_1 \to \ldots \to \tau_n \to \mu \vee \nu$, then either $\Gamma \vdash x{:}\tau_1 \to \ldots \to \tau_n \to \mu$ or $\Gamma \vdash x{:}\tau_1 \to \ldots \to \tau_n \to \nu$ $(n \geq 0)$.*

PROOF.

(i) Notice that $\tau \in T_1$ implies $\tau^* \equiv \tau$.

	$\Gamma \vdash x{:}\tau$	
$\Rightarrow$	$\Gamma(x) \leq \tau$	by Lemma 2.5.8*(i)*
$\Rightarrow$	$\Gamma(x) \leq_2 \tau$	by 2.6.17*(ii)* and Proposition 2.5.5*(iii)*
$\Rightarrow$	$\exists \mu \in T_1, \nu.\ \Gamma(x) = \mu \wedge \nu$ and $\mu \leq_1 \tau$	by Definition 2.5.3 since $\tau \in T_1$
$\Rightarrow$	$\exists \mu \in T_0, \nu.\ \Gamma(x) = \mu \wedge \nu$ and $\mu \leq_1 \tau$	by 2.6.17*(ii)*.

(ii) From *(i)* there are $\sigma_1 \in T_0, \sigma_2$ such that $\Gamma(x) = \sigma_1 \wedge \sigma_2$ and $\sigma_1 \leq_1 \tau_1^* \to \ldots \to \tau_n^* \to \mu \vee \nu$. Let $\sigma_1 \equiv \xi_1 \to \ldots \to \xi_n \to \xi$, where $\xi \in T_0$ by 2.6.17*(ii)*. Then $\xi \leq_1 \mu \vee \nu$ which implies, by Definition 2.5.3, either $\xi \leq_1 \mu$ or $\xi \leq_1 \nu$. $\square$

The principal pair carries out the same information of the corresponding approximate normal form. This implies that $pp(A)$ can be deduced only for approximate normal forms which are better than A according to the preorder $\preceq$. The proof of this fact will be done in Lemma 2.6.21 using some preliminary properties (Lemmas 2.6.19 and 2.6.20).

2.6.19. LEMMA. *Let $A \in \mathcal{A}$, Γ be a principal basis, and $\Gamma \vdash A{:}\sigma$.*

(i) $\sigma \in T_1$ *implies* $\exists A' \in \mathcal{A}_1, A'' \in \mathcal{A}.\ A \sim A' \| A''$ *and* $\Gamma \vdash A'{:}\sigma$.

(ii) $A \in \mathcal{A}_0$, *and* $\sigma \equiv \sigma_1 \vee \sigma_2 \in T_1$ *imply either* $\Gamma \vdash A{:}\sigma_1$ *or* $\Gamma \vdash A{:}\sigma_2$.

PROOF. *(i)* If $A \in \mathcal{A}_1$ it is trivial choosing $A' \equiv A$ and $A'' \equiv \Omega$. Otherwise, let $A \equiv A_1 \| \ldots \| A_m$, where $A_i \in \mathcal{A}_1$ $(1 \leq i \leq m)$. Then $\Gamma \vdash A{:}\sigma \Rightarrow \exists \tau_1, \ldots, \tau_m.\ \Gamma \vdash A_i{:}\tau_i$ and $\tau_1 \wedge \ldots \wedge \tau_m \leq \sigma$ by Lemma 2.5.8*(v)*.
Let $\tau_i^* = \bigwedge_{l \in L} \nu_{i,l}$ (where L depends on i). Then $\tau_1 \wedge \ldots \wedge \tau_m \leq \sigma$ and $\sigma \in T_1$ imply that there exist i, l such that $\nu_{i,l} \leq_1 \sigma$, by Definition 2.5.3 and Proposition 2.5.5. Hence $\Gamma \vdash A_i{:}\sigma$.

(ii) By cases on $A \in \mathcal{A}_0$.

- $A \equiv \Omega$ is trivial.
- $A \equiv xA_1 \ldots A_m$ $(m \geq 0)$ implies by 2.5.8*(iv)* $\Gamma \vdash x{:}\tau_1 \to \ldots \to \tau_m \to \sigma$ for some $\tau_1, \ldots, \tau_m$, so the result follows by 2.6.18*(ii)*.
- $A \equiv \lambda x.A'$ implies, by Lemma 2.5.8*(ii)*, $\tau_1 \wedge \ldots \wedge \tau_m \leq \sigma$ and $\Gamma \vdash A{:}\tau_j$ $(j \leq m)$ for some arrow types $\tau_1, \ldots, \tau_m$. Let $\tau_j^* = \bigwedge_{l \in L}(\mu_{j,l} \to \nu_{j,l})$ (where L depends on j). We have by Definition 2.5.3 and Proposition 2.5.5 that $\mu_{j,l} \to \nu_{j,l} \leq_1 \sigma$ for some j, l, since $\sigma \in T_1$. Being $\sigma \equiv \sigma_1 \vee \sigma_2$ we have by Definition 2.5.3 and by Proposition 2.5.5 $\mu_{j,l} \to \nu_{j,l} \leq_1 \sigma_1$ or $\mu_{j,l} \to \nu_{j,l} \leq_1 \sigma_2$. □

2.6.20. LEMMA. *Let $A \in \mathcal{A}$, Γ, Γ' be principal basis and τ be a principal type such that $\Gamma' \Subset \Gamma$ and $< \Gamma'; \tau > \in \Pi$. Then $\Gamma \vdash A : \tau$ implies $\Gamma' \vdash A : \tau$.*

PROOF. We prove a more general statement, i.e.:
Let $\Gamma, \Gamma', \Gamma''$ be principal bases and τ be a principal type such that $\Gamma'' \Subset \Gamma' \Subset \Gamma$ and $< \Gamma''; \tau > \in \Pi$. Then $\Gamma \vdash A : \tau$ implies $\Gamma' \vdash A : \tau$.
The proof is by a principal induction on A and a secondary induction on τ.
The case $A \equiv \Omega$ is immediate.
The case $\tau \in T_2 - T_1$ follows easily by the secondary induction. In fact if $\tau \equiv \tau_1 \wedge \tau_2$, then $\Gamma \vdash A : \tau$ implies both $\Gamma \vdash A : \tau_1$ and $\Gamma \vdash A : \tau_2$. Moreover by 2.6.17*(v)* there are $\Gamma_1, \Gamma_2 \Subset \Gamma'$ such that $< \Gamma_i, \tau_i > \in \Pi$ $(i = 1, 2)$.

$A \equiv xA_1 \ldots A_n$ and $n \geq 0$ implies by 2.5.8*(iv)* $\Gamma \vdash x{:}\sigma_1 \to \ldots \to \sigma_n \to \tau$ for some $\sigma_1, \ldots, \sigma_n$, such that $\Gamma \vdash A_i{:}\sigma_i$ for all $i \leq n$. By 2.6.18*(i)* $\Gamma(x) = \mu \wedge \nu$ for some μ, ν such that $\mu \in T_0$ and $\mu \leq_1 \sigma_1 \to \ldots \to \sigma_n \to \tau$. Let $\tau \equiv \sigma_{n+1} \to \ldots \to \sigma_{n+m} \to \tau'$ $(m \geq 0)$, where either τ' is a type variable, say $\tau' \equiv t$, or $\tau' \in T_1 - T_0$. Then $\mu \equiv \sigma_1' \to \ldots \to \sigma_{n+m}' \to t$ with $t \leq_1 \tau'$ and $\sigma_i \leq_2 \sigma_i'$ for $i \leq n + m$ by Definition 2.5.3. If $\tau' \not\equiv t$ by 2.5.3 we have that $\tau' = t \vee \tau''$ for some τ''. In both cases the hypothesis $< \Gamma''; \tau > \in \Pi$ assures us that t must occur in $\Gamma'' \Subset \Gamma'$. Therefore $\Gamma'(x) = \mu \wedge \nu'$ for some ν' and we have $\Gamma' \vdash x : \sigma_1' \to \ldots \to \sigma_{n+m}' \to \tau'$. Now $\Gamma \vdash A_i{:}\sigma_i$ implies $\Gamma \vdash A_i{:}\sigma_i'$ by rule $(\leq)$. By 2.6.17*(iii)* there are $\Gamma_i \Subset \Gamma'$ such that $< \Gamma_i; \sigma_i' > \in \Pi$. Therefore by the principal induction $\Gamma' \vdash A_i{:}\sigma_i'$ for all $i \leq n$. So we can conclude $\Gamma' \vdash A : \tau$.

$A \equiv \lambda x.A'$.

$\tau \in T_0$. Let $\tau \equiv \tau_1 \to \tau_2$. By 2.6.17*(iv)* $< \Gamma'', x : \tau_1; \tau_2 > \in \diamond$. By Lemma 2.5.8*(iii)* $\Gamma, x : \tau_1 \vdash A' : \tau_2$. Therefore by the principal or the secondary induction $\Gamma', x : \tau_1 \vdash A' : \tau_2$. By rule $(\to \mathrm{I})$ we conclude $\Gamma' \vdash A : \tau$.

$\tau \in T_1 - T_0$. Let $\tau \equiv \tau_1 \vee \tau_2$. By Lemma 2.6.19*(ii)* $\Gamma \vdash A : \tau_1$ or $\Gamma \vdash A : \tau_2$. By 2.6.17*(v)* there are $\Gamma_1, \Gamma_2 \Subset \Gamma'$ such that $< \Gamma_i; \tau_i > \in \sqcap \quad (i = 1, 2)$. Therefore the secondary induction applies.

$A \equiv A_1 + A_2$ implies by 2.5.8*(v)* $\Gamma \vdash A_1 \colon \tau$ and $\Gamma \vdash A_2 \colon \tau$. By the principal induction we have $\Gamma' \vdash A_1 \colon \tau$ and $\Gamma' \vdash A_2 \colon \tau$, so we can conclude $\Gamma' \vdash A \colon \tau$ by rule $(+\,\mathrm{I})$.

$A \equiv A_1 \| A_2$ implies by 2.5.8*(vi)* $\Gamma \vdash A_1 \colon \sigma_1$ and $\Gamma \vdash A_2 \colon \sigma_2$ for some σ_1, σ_2 such that $\sigma_1 \wedge \sigma_2 \leq \tau$. We need to consider only the case $\tau \in T_1$, therefore by 2.5.3 and 2.5.5 either $\sigma_1 \leq \tau$ or $\sigma_2 \leq \tau$. In the first case $\Gamma \vdash A_1 \colon \tau$, so by the principal induction $\Gamma' \vdash A_1 \colon \tau$. The second case is symmetric. □

2.6.21. LEMMA (Principal Pair Lemma).
If $A, A' \in \mathcal{A}$, $pp(A) = < \Gamma; \sigma >$ *and* $\Gamma \vdash A' : \sigma$, *then* $A \preceq A'$.

PROOF. By cases and then by induction on the structure of A. By hypothesis $A \in \mathcal{A}_2$.

Case $A \in \mathcal{A}_1$. In this case $\sigma \in T_1$, then by Lemma 2.6.19*(i)* there exists $B \in \mathcal{A}_1$ and some B' such that $A' \sim B \| B'$ and $\Gamma \vdash B \colon \sigma$. Let $B = B_1 + \ldots + B_n$ where $B_i \in \mathcal{A}_0$ $(1 \leq i \leq n)$; then $\Gamma \vdash B_i \colon \sigma$ $(i \leq n)$ by Lemma 2.5.8*(v)*. We distinguish three subcases after the shape of σ.

Subcase $\sigma \equiv t$. In this case we have $A \equiv x A_1 \ldots A_m$ for some $A_1, \ldots, A_m$ $(m \geq 0)$. Moreover by 2.6.17*(i)* there is only one type in Γ which contains the type variable t; let $x \colon \tau_1 \to \ldots \to \tau_m \to t \in \Gamma$. Therefore we have by Definition 2.6.15*(i)(b)*:

$$\Gamma = (\biguplus_{j \leq m} \Gamma_j) \uplus \{x \colon \tau_1 \to \ldots \to \tau_m \to t\} \text{ and } \Gamma_j \vdash A_j \colon \tau_j \ (j \leq m).$$

$\Gamma \vdash B_i \colon t$ $(i \leq n)$ and $B_i \in \mathcal{A}_0$ imply by Lemma 2.5.8*(ii)* $B_i \equiv x C_{i,1} \ldots C_{i,m}$. Moreover using Lemma 2.5.8*(i)* and *(iv)* $\Gamma \vdash C_{i,j} \colon \tau_j$ $(i \leq n, j \leq m)$. This implies by Lemma 2.6.20 $\Gamma_j \vdash C_{i,j} \colon \tau_j$ $(i \leq n, j \leq m)$. So we have by induction $A_j \preceq C_{i,j}$ $(j \leq m)$. Therefore

$$A_j \preceq C_{i,j} \ (i \leq n, j \leq m) \Rightarrow A \preceq B_i \ (i \leq n) \Rightarrow A \preceq B \Rightarrow A \preceq A'.$$

Subcase $\sigma \equiv \tau \to \mu$. In this case $A \equiv \lambda x.A''$. If $B_i \equiv \lambda x.B'_i$, it is easy by induction. If B_i is a λ-free term, then also $\lambda z.B_i z \in \mathcal{A}_0$, where z is fresh, and $\Gamma \vdash \lambda z.B_i z \colon \tau \to \mu$. We are in the previous case and we can prove $A \preceq_0 \lambda z.B_i z$, so we can conclude $A \preceq_0 B_i$.

Subcase $\sigma \equiv \tau_1 \vee \tau_2$. In this case we have $A \equiv A_1 + A_2$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $\Gamma_j \vdash A_j \colon \tau_j$ $(j = 1, 2)$ by 2.6.15*(i)(d)*. $\Gamma \vdash B_i \colon \sigma$ implies, by Lemma 2.6.19*(ii)*, $\exists l_i \leq 2$. Then we have $\Gamma \vdash B_i \colon \tau_{l_i}$, since $B_i \in \mathcal{A}_0$. This implies by Lemma 2.6.20 $\Gamma_{l_i} \vdash B_i \colon \tau_{l_i}$. By induction, $A_{l_i} \preceq B_i$, for all $i \leq n$, which implies $A \preceq B$, so we can conclude $A \preceq A'$.

Case $A \notin \mathcal{A}_1$. In this case $\sigma \equiv \tau_1 \wedge \tau_2$, $A \equiv A_1 \| A_2$ $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $\Gamma_j \vdash A_j : \tau_j$ $(j = 1, 2)$. By rule $(\leq)$ we have $\Gamma \vdash A' : \tau_j$ $(j = 1, 2)$ and this implies by Lemma 2.6.20 $\Gamma_j \vdash A' : \tau_j$ $(j = 1, 2)$. By induction $A_1 \preceq A'$ and $A_2 \preceq A'$, so we can conclude $A \preceq A'$. □

We are finally in place to prove:

2.6.22. Theorem (Second Adequacy Theorem).
The filter λ-lattice is adequate for the semantics based on capabilities, i.e.:

$$M \sqsubseteq^{\mathcal{C}} N \Rightarrow M \sqsubseteq^{\mathcal{A}} N.$$

Proof. We prove $M \not\sqsubseteq^{\mathcal{A}} N \Rightarrow M \not\sqsubseteq^{\mathcal{C}} N$. By Proposition 2.3.11*(iii)*,

$$M \not\sqsubseteq^{\mathcal{A}} N \Rightarrow \exists A \in \mathcal{A}(M). A \notin \mathcal{A}(N).$$

Let $pp(A^*) = < \Gamma; \sigma >$; by the Approximation Theorem, $\Gamma \vdash M : \sigma$. Assume now $\Gamma \vdash N : \sigma$. Then, by the Approximation Theorem again, there exists $A' \in \mathcal{A}(N)$ such that $\Gamma \vdash A' : \sigma$. Hence, by the Principal Pair Lemma, $A \preceq A'$ so that $A \in \mathcal{A}(N)$, which is absurd. It follows that $\Gamma \not\vdash N : \sigma$, so we can conclude $M \not\sqsubseteq^{\mathcal{L}} N$. □

We immediately have

2.6.23. Theorem (Full Abstraction Theorem). *The filter λ-lattice is fully abstract for the semantics based on capabilities, i.e.:*

$$M \sqsubseteq^{\mathcal{C}} N \Leftrightarrow M \sqsubseteq^{\mathcal{A}} N.$$

Proof. Immediate consequence of the Approximation Theorem 2.6.8 and of the Second Adequacy Theorem 2.6.22. □

From the Full Abstraction Theorem and the invariance of types under $=_a$ (Theorem 2.5.9), we have that the set of approximate normal forms is invariant under $=_a$ (see Remark 2.3.2).

By the Full Abstraction Theorem, in Proposition 2.5.19 we can replace $\sqsubseteq^{\mathcal{C}}$ by $\sqsubseteq^{\mathcal{A}}$.

Theorems 2.6.10 and 2.6.23 relate also the two operational semantics we considered: as expected $\sqsubseteq^{\mathcal{A}}$ turns out to be a refinement of $\sqsubseteq^{\mathcal{O}}$.

Chapter 3

Lazy Lambda-calculus

3.1. Introduction

Our study confronts various problems that had their origin in the theory of functional languages and λ-calculus. In [84] Plotkin showed that Scott continuous functions over domains are overaboundant to give meaning to the sequential functional language that has been called PCF (a simply typed λ-calculus with arithmetical constants, booleans, if-then-else and fixed-point operator). To be precise, he considered the following notion of operational equivalence. Two terms M and N of the same type are *operationally equivalent* if and only if, for all contexts $C[\ ]$ of ground type such that both $C[M]$ and $C[N]$ are well-typed closed terms, either the evaluations of $C[M]$ and $C[N]$ do not terminate (converge), or both terminate and give the same result. It comes out that, if two terms have the same denotation in the standard model (in which ground types are flat cpos and arrow types are interpreted as spaces of Scott continuous functions) then they are operationally equivalent (adequacy theorem); but the converse (full abstraction) does not hold.

In the same paper Plotkin proved that syntax can be reasonably enriched to get full abstraction, and that this can be achieved by using a suitable kind of parallel operators or combinators. Milner proved in [71] that this is also a necessary condition: any model of PCF is fully abstract if and only if all "finite" objects in the model are definable. Conversely, the standard model becomes fully abstract if we endow the calculus with operators that reinforce its expressive power such that it satisfies Milner definability requirement.

The same incompleteness phenomenon with respect to standard continuous semantics has been found for the lazy λ-calculus in [6]. This is a type free calculus, having the same syntax of pure λ-calculus and a reduction relation with just two rules:

$$(\lambda x.M)N \longrightarrow M[N/x] \quad \text{and} \quad \frac{M \longrightarrow M'}{MN \longrightarrow M'N}.$$

The full abstraction problem can be reformulated in this setting, even if we do not have the notion of ground type. Indeed Abramsky and Ong define the set Val of values as the set of abstractions. Then their notion of *may convergence* is: M may converge to V, written $M \Downarrow^{\mathrm{may}} V$, if V is a value and $M \longrightarrow^* V$. In [6] the operational semantics is given by axiomatizing $M \Downarrow^{\mathrm{may}} V$, instead of giving the reduction relation as a primitive notion: of course this is equivalent.

As a matter of fact, the problem of enriching the calculus so that the standard model is fully abstract can be solved by adding a combinator $\mathbf{P}$ testing convergence in parallel. More precisely

P satisfies

$$[\exists V.\ M \Downarrow^{\rm may} V \text{ or } N \Downarrow^{\rm may} V] \quad \Rightarrow \quad \mathbf{P}MN \Downarrow^{\rm may} \mathbf{I},$$

where $\mathbf{I} \equiv \lambda x.x$ is the identity combinator. This gives a combinator which tests convergence, i.e. a closed term $\mathbf{C}$ such that, for any term M, $\mathbf{C}M$ reduces to $\mathbf{I}$ if M reduces to a value and diverges otherwise: just take $\mathbf{C} \equiv \lambda x.\mathbf{P}xx$.

In [22] a further step is made by Boudol. The combinator $\mathbf{P}$ is split into its two components, namely parallelism and convergence test. The parallelism implicit in $\mathbf{P}$ is made explicit by adding a binary operator $\|$ such that

$$M \| N \Downarrow^{\rm may} \;\Leftrightarrow\; M \Downarrow^{\rm may} \;\text{ or }\; N \Downarrow^{\rm may},$$

where $M \Downarrow^{\rm may}$ abbreviates $\exists V.\ M \Downarrow^{\rm may} V$. To have this, with the above definition of convergence, the following rules suffice

$$\frac{M \longrightarrow M'}{M \| N \longrightarrow M' \| N} \quad \text{and} \quad \frac{N \longrightarrow N'}{M \| N \longrightarrow M \| N'}.$$

As the intended meaning of a parallel composition is a function, Boudol adds the following rule

$$(M \| N)L \longrightarrow (ML) \| (NL).$$

The internal convergence test is achieved using, besides standard call-by-name abstraction, call-by-value abstraction, originally considered by Landin [60] and Plotkin [81]. To see how this works, let us extend the set Val of values inductively so that it includes all terms of the shape $V \| N$ or $M \| V$, where V is a value. We use two sorts of variables to distinguish between call-by-value and call-by-name abstraction, namely $v, w, \ldots$ for call-by-value variables and $x, y, \ldots$ for call-by-name variables. Then we add to the lazy λ-calculus and to the rules for $\|$, the following rules

$$(\lambda v.M)V \longrightarrow M[V/v] \;\text{ if } V \in \mathsf{Val}, \qquad \frac{N \longrightarrow N'}{(\lambda v.M)N \longrightarrow (\lambda v.M)N'} \;\text{ if } N \notin \mathsf{Val}.$$

Now $\mathbf{P}$ becomes definable by $\lambda xy.\ (\lambda v.\mathbf{I})(x \| y)$.

We observe that the combination of parallelism (angelic non-determinism) and call-by-value is much more powerful than the use of combinators directly defining a parallel convergence test. First, the notion of being a value is no more equivalent to that of being irreducible. Moreover, as remarked in the early paragraphs of this introduction, $M \| N$ has to be interpreted as a multivalued function, since M and N are not necessarily interpreted by compatible functions. So the model of [22] is a solution of the domain equation $D = \mathcal{P}^\flat([D \to D]_\perp)$, where $[D \to D]_\perp$ is the lifted space of continuous functions, and $\mathcal{P}^\flat$ is the lower powerdomain functor (also called Hoare's powerdomain, see [92] for a definition). Since Boudol works in the category of prime algebraic lattices, he has this solution for free. In fact in that category $D \simeq \mathcal{P}^\flat(\mathsf{KP}(D))$, where $\mathsf{KP}(D)$ is the set of compact coprime elements of D. Let us recall that a *complete lattice* is a partial order $(D, \sqsubseteq)$ such that each subset X of D has a least upperbound $\bigsqcup X$. An element d of a complete lattice is *compact* if $d \sqsubseteq \bigsqcup X$ implies $d \sqsubseteq \bigsqcup Y$ for some finite subset Y of X. An element $d \in D$ is *coprime* if and only if $d \sqsubseteq x \sqcup y$ implies $d \sqsubseteq x$ or $d \sqsubseteq y$. A complete lattice is *prime algebraic* if any element is the join of the compact coprime elements it dominates. See also the discussion at the beginning of section 3.4.

Serious problems arise when we consider the full language, modeling also the demonic non-determinism (see [78, 79]), which is the central issue of the present chapter. Suppose that an internal choice operator $+$ is added, with the obvious reduction rules

$$M + N \longrightarrow M \quad \text{and} \quad M + N \longrightarrow N.$$

Then, following ideas explained above (see also [65]), we expect a convergence predicate $\Downarrow$ such that

$$M + N\Downarrow \quad \Leftrightarrow \quad M\Downarrow \;\text{ and }\; N\Downarrow .$$

But this is not true with the present definition of $\Downarrow^{\mathrm{may}}$.

The convergence predicate considered above (and in [22]) is a *may* convergence predicate, to be related to may testing equivalence if convergence is the only observable property (see [3, 47, 6]). A solution would be to consider a *must* convergence predicate as in [65] (see also [49]). An informal definition is the following: $M\Downarrow^{\mathrm{must}}$ if and only if there is an n such that every reduction out of M reaches a value within a number of steps bounded by n. Otherwise we write $M\Uparrow^{\mathrm{must}}$.

Of course, if we have to avoid the collapse of $\|$ and $+$ with respect to the predicate $\Downarrow^{\mathrm{must}}$, something has to be changed in the operational semantics of $\|$. In fact, with the old definition of $\|$-reduction rules, if we put for example $\Delta \equiv \lambda x.xx$ and we take the typical divergent combinator $\Omega \equiv \Delta\Delta$, then we have that $(\mathbf{II})\|\Omega\Uparrow^{\mathrm{must}}$. The problem is that nothing prevents the reduction of a parallel composition from being unfair: there exists a reduction out of $(\mathbf{II})\|\Omega$ that contracts Ω infinitely many times and never reaches the value $\mathbf{I}\|\Omega$. Really, we want to identify $\mathbf{I}\|\Omega$ with $\mathbf{I}$, since $\|$ is intended to take the best of its arguments; notice that the mentioned terms are not equivalent in a standard must semantics (see [30]), when the parallel operator is asynchronous.

There are many possibilities of changing the reduction rules for $\|$ in such a way that we cannot reduce infinitely many times on one side of a parallel composition, when the other one is reducible. We take the simplest way to get this kind of *fair* reduction and we introduce the rules

$$\frac{M \longrightarrow M' \quad N \longrightarrow N'}{M\|N \longrightarrow M'\|N'} \qquad\qquad \frac{M \longrightarrow M' \quad N \not\longrightarrow}{M\|N \longrightarrow M'\|N, \quad N\|M \longrightarrow N\|M'}$$

as our actual choice (see [31]), where $N \not\longrightarrow$ means that N is irreducible.

This implies that, as in [22], a value of the shape $V\|M$ is not necessarily a normal form, as M can be reduced. This fact, together with the presence of the choice operator, makes the β-rule for call-by-value sensible to the relative speed of parallel evaluations of its arguments.

To illustrate this, let us consider the context $C[\;] \equiv (\lambda v.vv)[\;]\Omega\mathbf{I}$ and the values $V \equiv \mathbf{I}\|(\mathbf{K}+\mathbf{O})$, $V' \equiv \mathbf{I}\|\mathbf{K}$ and $V'' \equiv \mathbf{I}\|\mathbf{O}$, where $\mathbf{K} \equiv \lambda xy.x$ and $\mathbf{O} \equiv \lambda xy.y$. Then $V \longrightarrow V'$ and $V \longrightarrow V''$. Now (writing $\xrightarrow{n}$ for $n > 1$ reduction steps)

$$\begin{array}{rcl}
C[V'] & \longrightarrow & (\mathbf{I}\|\mathbf{K})(\mathbf{I}\|\mathbf{K})\Omega\mathbf{I} \\
 & \xrightarrow{3} & (\mathbf{I}(\mathbf{I}\|\mathbf{K})\Omega\mathbf{I})\|(\mathbf{K}(\mathbf{I}\|\mathbf{K})\Omega\mathbf{I}) \\
 & \longrightarrow & ((\mathbf{I}\|\mathbf{K})\Omega\mathbf{I})\|((\lambda y.(\mathbf{I}\|\mathbf{K}))\Omega\mathbf{I}) \\
 & \longrightarrow & ((\mathbf{I}\Omega\|\mathbf{K}\Omega)\mathbf{I})\|((\mathbf{I}\|\mathbf{K})\mathbf{I}) \\
 & \longrightarrow & (\mathbf{I}\Omega\mathbf{I})\|(\mathbf{K}\Omega\mathbf{I})\|(\mathbf{II})\|(\mathbf{KI}) \\
 & \longrightarrow & (\Omega\mathbf{I})\|((\lambda y.\Omega)\mathbf{I})\|\mathbf{I}\|(\lambda y.\mathbf{I}) \\
 & \longrightarrow & (\Omega\mathbf{I})\|\Omega\|\mathbf{I}\|(\lambda y.\mathbf{I})
\end{array}$$

which is a value, and it is not hard to see that this is the only reduction out of $C[V']$ according to the rules defined in 3.2.2. Similarly,

$$\begin{array}{rcl}
C[V''] & \longrightarrow & (\mathbf{I}\|\mathbf{O})(\mathbf{I}\|\mathbf{O})\Omega\mathbf{I} \\
 & \longrightarrow^{*} & (\Omega\mathbf{I})\|\mathbf{I}\|(\Omega\mathbf{I})
\end{array}$$

and again this is the only reduction out of $C[V'']$. But now consider the following reduction of

$C[V]$

$$\begin{array}{rcll} C[V] & \longrightarrow & (\mathbf{I}\|(\mathbf{K}+\mathbf{O}))(\mathbf{I}\|(\mathbf{K}+\mathbf{O}))\Omega\mathbf{I} & \\ & \xrightarrow{3} & (\mathbf{I}(\mathbf{I}\|(\mathbf{K}+\mathbf{O}))\Omega\mathbf{I})\|((\mathbf{K}+\mathbf{O})(\mathbf{I}\|(\mathbf{K}+\mathbf{O}))\Omega\mathbf{I}) & \\ & \longrightarrow & ((\mathbf{I}\|(\mathbf{K}+\mathbf{O}))\Omega\mathbf{I})\|(\mathbf{O}(\mathbf{I}\|(\mathbf{K}+\mathbf{O}))\Omega\mathbf{I}) & \text{choosing } \mathbf{O} \ldots \\ & \xrightarrow{2} & (\mathbf{I}\Omega\mathbf{I})\|((\mathbf{K}+\mathbf{O})\Omega\mathbf{I})\|(\Omega\mathbf{I}) & \\ & \longrightarrow & (\Omega\mathbf{I})\|(\mathbf{K}\Omega\mathbf{I})\|(\Omega\mathbf{I}) & \ldots \text{ choosing } \mathbf{K} \\ & \longrightarrow & (\Omega\mathbf{I})\|\Omega\|(\Omega\mathbf{I}) & \end{array}$$

and from $(\Omega\mathbf{I})\|\Omega\|(\Omega\mathbf{I})$ we will never reach a value.

This example also shows that there are values V_0, V_1 and V_2 such that $V_0\|(V_1 + V_2)$ and $(V_0\|V_1) + (V_0\|V_2)$ would have different behaviors in some context, although this would be unexpected under any reasonable operational semantics. Indeed, $(\lambda v.vv)(V_0\|(V_1 + V_2))$ can reduce to $(V_0\|(V_1 + V_2))(V_0\|(V_1 + V_2))$, while $(\lambda v.vv)((V_0\|V_1) + (V_0\|V_2))$ can reduce either to $(V_0\|V_1)(V_0\|V_1)$ or to $(V_0\|V_2)(V_0\|V_2)$, but never to $(V_0\|(V_1+V_2))(V_0\|(V_1+V_2))$. Note that in the present context call-by-name and call-by-value implement run-time-choice and call-time-choice respectively (see [65]).

The problem of correcting the β-contraction rule for call-by-value is that, given a value V, we cannot decide whether it has been computed enough to perform the reduction step $(\lambda v.M)V \longrightarrow M[V/v]$, or if it is necessary to reduce V further, before contracting the outermost β-redex. We cannot reduce V as long as possible, since this could not terminate. In the meantime, $M[V/v]$ can diverge while $M[V'/v]$ can converge for all V' which are reducts of V, as shown by the previous example. On the other hand, any effective description of the operational semantics calls for a definition of a recursive one step reduction relation.

Now the solution we propose is to distinguish two cases: if V is an irreducible value (namely a λ-abstraction or the parallel composition of irreducible values), then the standard call-by-value β-contraction rule applies. If, instead, V can be reduced further, to compute $(\lambda v.M)V$ we want "take the best" between the terms $M[V'/v]$, for all V' such that $V \longrightarrow^* V'$. We realize this by evaluating in parallel $M[V/v]$ and $(\lambda v.M)V'$ for all V' such that $V \longrightarrow V'$. Using the operator $\|$, this can be formalized in our calculus as follows

$$\frac{V \not\longrightarrow \quad V \in \mathsf{Val}}{(\lambda v.M)V \longrightarrow M[V/v]} \qquad \frac{V \longrightarrow V' \quad V \in \mathsf{Val}}{(\lambda v.M)V \longrightarrow M[V/v]\|(\lambda v.M)V'}.$$

In other words, the solution we propose is to distinguish between *total* and *partial* values. A total value is an irreducible value, while a partial value is of the form $M\|N$ in which either M or N is not a total value. So we split the call-by-value β-contraction in two rules.

To conclude this part of our discussion, let us spend a few words to emphasize the effectiveness of the evaluation mechanism as a distinguishing feature of our calculus. As it is clear from the previous exposition, the papers closest to the present chapter are [22] and [78]. While our treatment improves on the former because of the presence in the same calculus of both angelic and demonic non-determinism, it improves on the second since the operational semantics on which we base our theory is effective. Indeed, the reduction relation is (as usual) presented by means of a formal system in the sense of Post, and the convergence predicate is (up to coding) recursively enumerable. This is mandatory when one expects to capture the intentional aspects of evaluation, and justifies our reduction relation as it will be defined in the technical development of the chapter.

Turning to the type assignment system, if M has type $\sigma \vee \tau$ then it can be that M evaluates both to some P and Q such that P has type σ, Q hat type τ, but neither P has type τ nor Q

has type σ. In this case, M has an essentially disjunctive type, which is possible even if M is a partial value. But all is determined in case of total values. So we expect the system to have the "disjunction property" for total values: if a total value has the type $\sigma \vee \tau$, then either σ or τ can be assigned to it (hence to all its reducts).

Consequently, we distinguish between call-by-name and call-by-value abstraction making a substantial use of disjunction. This intuitively explains why the following rule

$$\frac{\Gamma \vdash \lambda v.M : (\sigma \to \rho) \wedge (\tau \to \rho)}{\Gamma \vdash \lambda v.M : \sigma \vee \tau \to \rho}$$

is correct for call-by-value but not for call-by-name abstraction. Observe that this means that call-by-value abstraction yields a co-additive function (namely, meet preserving), which is the expected semantics of call-by-value in our setting.

As an example, if $M \equiv x\mathbf{I}\Omega \| x\Omega\mathbf{I}$, $\rho \equiv \omega \to \omega$, $\sigma \equiv \rho \to \omega \to \rho$, $\tau \equiv \omega \to \rho \to \rho$, then we have that $\vdash \lambda x.M \colon (\sigma \to \rho) \wedge (\tau \to \rho)$. Moreover, $\vdash \mathbf{K} \colon \sigma$ and $\vdash \mathbf{O} \colon \tau$, so that allowing the rule above for call-by-name abstraction one could deduce $(\lambda x.M)(\mathbf{K} + \mathbf{O}) \colon \rho$, using the rules for $+$ introduction and $\to$ elimination, too. But this would destroy the subject-reduction property, since $(\lambda x.M)(\mathbf{K} + \mathbf{O})$ reduces to $\Omega \| \Omega$, for which only types equivalent to ω can be deduced (see Corollary 3.5.6*(ii)*).

The domain that is determined by the type theory considered in this chapter is isomorphic to the initial solution of the domain equation $D = \mathcal{P}^\sharp([D \to D]_\perp)$ in the category of continuous lattices, where $\mathcal{P}^\sharp$ is the upper powerdomain functor (also called Smyth's powerdomain, see [92]). This is sound with respect to the operational semantics since this powerdomain constructor is needed to model demonic non-determinism, as angelic non-determinism is built in, by the fact that we work with prime algebraic lattices. This domain equation, and their relations to Abramsky and Boudol equations [5, 22], will be discussed further at the beginning of section 3.4.

We then get a notion of environment model for the present calculus, in the sense of [54]. The filter model induced by our type assignment turns out to fit into this notion, a fact that will be used to prove completeness of type inference.

Our study culminates in the full abstraction theorem, that we will prove by means of characteristic terms extending [22].

In Section 3.2 we formally define the concurrent λ-calculus and its reduction rules. We consider the reduction trees of terms to introduce convergence. Moreover, we consider another reduction relation, whose main feature is to characterize convergent terms as those which reduce to a sum of values.

Section 3.3 deals with types and the type assignment system. Crucial is the choice of the preorder on types, which will determine the topological structure of the filter model. The type assignment system turns out to enjoy structural properties which allow to prove preservation of type under subject reduction. The main result of this section is that all convergent terms can be typed by $\omega \to \omega$.

Section 3.4 presents the filter model as the initial solution of a suitable domain equation. Then we introduce the notion of environment model for concurrent λ-calculus and we prove that the filter model is in fact an environment model. This allows to have the completeness of type assignment.

Finally, we prove in section 3.5 the full abstraction of the filter model. First we define for each type a test term and a characteristic term. The application of the test term to an argument M converges only if M has the corresponding type. By means of a realizability interpretation of

types we show that all terms typed by $\omega \to \omega$ converge. This, together with the main result of section 3.3, implies that $\omega \to \omega$ completely characterizes convergence. Then the full abstraction of the model follows easily.
The content of this chapter is essentially [34].

3.2. The Calculus and its Operational Semantics

We extend the syntax of pure λ-calculus with a non-deterministic choice operator $+$ and a parallel operator $\|$. We use two sorts of variables, namely the set Vn of call-by-name variables, ranged over by x, y, z and the set Vv of call-by-value variables, ranged over by v, w. The symbol χ will range over the set $\mathsf{Vn} \cup \mathsf{Vv}$. The terms of the concurrent λ-calculus are defined by the following grammar

$$M ::= x \mid v \mid (\lambda x.M) \mid (\lambda v.M) \mid (MM) \mid (M+M) \mid (M\|M).$$

We call $\Lambda_{+\|}$ the set of terms. For any $M \in \Lambda_{+\|}$, $FV(M)$ denotes the set of free variables of M; $\Lambda^0_{+\|}$ is the set of terms M such that $FV(M) = \emptyset$. Moreover, we shall refer to the following set

$$\mathsf{Par} = \{(M\|N) \mid M, N \in \Lambda_{+\|}\}.$$

Notation. We use $\equiv$ for syntactical equality up to renaming of bound variables.
As usual for pure λ-calculus, we assume that application associates to the left and we write e.g. MNP instead of $((MN)P)$. If $\vec{L} \equiv L_1 \cdots L_n$ is any (possibly empty) vector of terms, then $M\vec{L} \equiv ML_1 \ldots L_n$. The expression $\lambda\chi_1 \ldots \chi_n.M$ is short for $(\lambda\chi_1.(\ldots(\lambda\chi_n.M)\ldots))$.
We will abbreviate some λ-terms as follows

$$\begin{array}{lll} \mathbf{I} \equiv \lambda x.x & \mathbf{K} \equiv \lambda xy.x & \mathbf{O} \equiv \lambda xy.y \\ \Delta \equiv \lambda x.xx & \Omega \equiv \Delta\Delta & \mathbf{Y} \equiv \lambda y.(\lambda x.y(xx))(\lambda x.y(xx)). \end{array}$$

Application and abstraction have precedence over $+$ and $\|$, e.g. $MN+P$ stands for $((MN)+P)$ and $\lambda x.M+N$ for $((\lambda x.M)+N)$. The operator $\|$ takes precedence over $+$: for example $M\|P+Q$ is short for $((M\|P)+Q)$. External parentheses are always omitted.
The operators $+$ and $\|$ will be written up to associativity: thjs will be enforced by our semantics.
We shall also make use of the following abbreviation

$$\sum_{i=1}^{n} M_i \equiv M_1 + \cdots + M_n.$$

Moreover, if $\mathcal{M} = \{M_1, \ldots, M_n\}$ is any finite multiset of terms then

$$\sum \mathcal{M} \equiv \sum_{i=1}^{n} M_i.$$

Observe that, being $\mathcal{M}$ a multiset, it can be the case that $M_i \equiv M_j$ for different i and j.

3.2.1. Partial versus Total Values

As discussed in the introduction, we need to distinguish between partial and total values; the main difference concerns the parallel operator. In fact we require both M and N to be total values to ensure that $M\|N$ is a total value, while in general it suffices that either M or N is a value to have that $M\|N$ is a value. As it is clear from the next definition, a value is either a total or a partial value.

3.2.1. Definition. We define the set Val of *values* according to the grammar

$$V ::= v \mid \lambda x.M \mid \lambda v.M \mid V \| M \mid M \| V$$

and the set TVal of *total values* as the subset of Val

$$W ::= v \mid \lambda x.M \mid \lambda v.M \mid W \| W \ .$$

A value V is *partial* iff $V \notin \mathsf{TVal}$.

We now introduce a reduction relation which is intended to formalize the expected behavior of a machine which evaluates in a synchronous way parallel compositions, until a value is produced. Partial values can be further evaluated, and this is essential in case of an application of a call-by-value abstraction. Therefore, in some cases an asynchronous evaluation of parallel composition is permitted.

It follows that the convergence predicate will not be any more coincident with the property of being (strongly) normalizable (see [22] for a similar proposal, even if in a may perspective) with respect to the given reduction relation. Observe that in the lazy λ-calculus of [6], as well as in the present calculus, $\lambda\chi.M$ is a normal form, no matter whether M is reducible or not.

3.2.2. Definition.

(i) The reduction relation $\longrightarrow$ is the least binary relation over $\Lambda^0_{+\|}$ such that

$$(\beta) \quad (\lambda x.M)N \longrightarrow M[N/x] \qquad\qquad (\beta_v) \quad \frac{W \in \mathsf{TVal}}{(\lambda v.M)W \longrightarrow M[W/v]}$$

$$(\beta_v\|) \quad \frac{V \longrightarrow V' \qquad V \in \mathsf{Val}}{(\lambda v.M)V \longrightarrow M[V/v] \| (\lambda v.M)V'} \qquad\qquad (\mu_v) \quad \frac{N \longrightarrow N' \quad N \notin \mathsf{Val}}{(\lambda v.M)N \longrightarrow (\lambda v.M)N'}$$

$$(\nu) \quad \frac{M \longrightarrow M' \qquad M \notin \mathsf{Val} \bigcup \mathsf{Par}}{MN \longrightarrow M'N} \qquad\qquad (\|_{app}) \quad (M\|N)L \longrightarrow ML\|NL$$

$$(\|_s) \quad \frac{M \longrightarrow M' \quad N \longrightarrow N'}{M\|N \longrightarrow M'\|N'} \qquad\qquad (\|_a) \quad \frac{M \longrightarrow M' \quad W \in \mathsf{TVal}}{M\|W \longrightarrow M'\|W,\ W\|M \longrightarrow W\|M'}$$

$$(+) \quad M + N \longrightarrow M \quad , \quad M + N \longrightarrow N.$$

(ii) We denote by $\longrightarrow^*$ the reflexive and transitive closure of $\longrightarrow$.

3.2.3. Lemma.

(i) $W \in \Lambda^0_{+\|}$ *is irreducible wrt* $\longrightarrow$ *iff* $W \in \mathsf{TVal}$*;*

(ii) If $V \in \Lambda^0_{+\|} \cap \mathsf{Val}$*, then either* $V \in \mathsf{TVal}$ *or* $V \longrightarrow V'$ *for some* $V' \in \mathsf{Val}$*;*

(iii) If $W, W_1, \ldots, W_n \in \mathsf{TVal}, N_1, \ldots, N_m \in \Lambda_{+\|}$*, then*

$$W[N_1/x_1, \ldots, N_m/x_m, W_1/v_1, \ldots, W_n/v_n] \in \mathsf{TVal}.$$

Proof. Easy by definitions. □

It is useful to consider reduction trees of terms and their bars.

3.2.4. Definition. Let $M \in \Lambda^0_{+\|}$.

(i) $tree(M)$ is the (unordered) reduction tree of M;

(ii) A *bar* of $tree(M)$ is a subset of the nodes of $tree(M)$ s.t. each maximal path intersects the bar at exactly one node;

(iii) $bar(M)$ is the set of bars of $tree(M)$;

(iv) For $b \in bar(M)$ the *height* of b (notation: $height(b)$) is the maximum of the heights of its nodes.

Inspecting the reduction rules, we see that $tree(M)$ is a finitely branching tree for all $M \in \Lambda^0_{+\|}$. This implies by König's Lemma that if we cut $tree(M)$ at a fixed height we obtain a finite tree. Since all nodes belonging to a bar b are in the subtree of $tree(M)$ obtained by cutting $tree(M)$ at $height(b)$, we have that $b \in bar(M)$ is always a finite set of nodes (see also [19]). This does not contradict the fact that a term may have infinite reduction paths. For example, let us consider the infinite reduction tree of $\mathbf{Y}M$, where $M \equiv \lambda x.(\mathbf{I} + x)$, which is shown in Figure 3.1. Admittedly, the set of nodes in $tree(\mathbf{Y}M)$ which are labeled by $\mathbf{I}$ is infinite, but it is not a bar. Indeed the infinite path in this tree does not have any node in such set and every $b \in bar(\mathbf{Y}M)$ must contain exactly one node of this path. Whichever node we choose on the infinite path we will exclude all nodes with greater height, so that b comes out to be finite.

$$
\begin{array}{llll}
 & \nearrow \mathbf{I} & & \\
\mathbf{Y}M \to N \to MN \to \mathbf{I}+N & & \nearrow \mathbf{I} & \\
 & \searrow N \to MN \to \mathbf{I}+N & & \\
 & & \searrow N \to \cdots &
\end{array}
$$

Figure 3.1: Reduction tree of $\mathbf{Y}M$, where $N \equiv (\lambda x.M(xx))(\lambda x.M(xx))$.

A bar is always relative to a tree and cannot be identified with the set of the labels of its nodes. For example $tree(M + \mathbf{I}M)$ has the shape shown in Figure 3.2. Now the indicated set of nodes b is a bar whose set of labels is the singleton $\{M\}$. But the set containing a single node labeled by M is not a bar of this tree. Moreover the height of the bar b is two, but if b would be identified with $\{M\}$, then $height(b)$ would be ambiguously one or two.

However, if $b \in bar(M)$, then two subtrees rooted in two nodes of b are equal if and only if their labels are the same. Hence we abuse notation and we write $b = \{M_1, \ldots, M_n\}$ (if $M_1, \ldots, M_n$ is the multiset of labels of nodes belonging to b). The abbreviations $M \in b$ and $b \subseteq \mathsf{Val}$ will have the obvious meanings.

3.2.2. Convergency

We now define the convergence predicate. A term is convergent if and only if all reduction paths will eventually reach a value. In other words, a term M converges if and only if there is a bar in $tree(M)$ which is a subset of Val. To formalize this, it is useful to introduce the bar $\mathcal{R}(M, k)$ whose labels are those terms which can be reached starting from a term M by performing k steps of reduction.

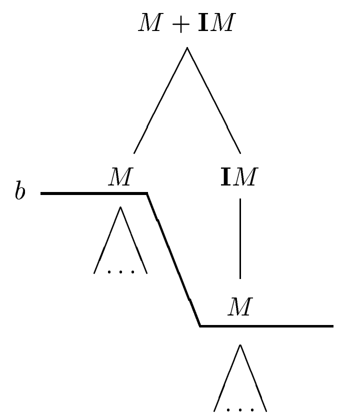

Figure 3.2: Reduction tree of $M + \mathbf{I}M$.

3.2.5. DEFINITION. Let $M \in \Lambda^0_{+\|}$, then

(i) $\mathcal{R}(M, k) \in bar(M)$ is the cut of $tree(M)$ at height k, namely it is the unique bar such that

(a) $height(\mathcal{R}(M, k)) \leq k$;

(b) $\forall M' \in \mathcal{R}(M, k).\ height(M') < k \;\Rightarrow\; M' \in \mathsf{TVal}$.

(ii) $M{\Downarrow}_k \Leftrightarrow \mathcal{R}(M, k) \subseteq \mathsf{Val}$;

(iii) $M{\Downarrow} \Leftrightarrow \exists k.\ M{\Downarrow}_k$.

Note that $(M + N){\Downarrow}$ if and only if both $M{\Downarrow}$ and $N{\Downarrow}$. On the other hand $(M\|N){\Downarrow}$ if and only if either $M{\Downarrow}$ or $N{\Downarrow}$ (or both). So $\Downarrow$ coincides with $\Downarrow^{\text{must}}$ as informally defined above. In general, if for some $b \in bar(M)$ we have $M'{\Downarrow}$ for all $M' \in b$, then $M{\Downarrow}$. The vice-versa is obviously true.

We depart from the standard way of defining must semantics using infinite paths (see [30]). This gives us a different theory of terms, for example we equate $\mathbf{I}$ and $\mathbf{I}\|\Omega$.

To study the operational semantics of our calculus it is useful to introduce a binary relation $\rhd$ whose main features are

- to satisfy the Church-Rosser property;
- to simulate the choices performed by rule $(+)$ without losing information about the discarded parts;
- to characterize the convergent terms as those which reduce to a sum of values.

Moreover we will consider the equivalence relation $\bowtie$ generated by $\rhd$.

3.2.6. DEFINITION.

(i) Define $\rhd$ as the least binary relation over $\Lambda^0_{+\|}$ such that

$(\beta)'$ $(\lambda x.M)N \rhd M[N/x]$,

$(\beta_v)'$ $(\lambda v.M)W \rhd M[W/v]$, if $W \in \mathsf{TVal}$,

$(\beta_v\|)'$ $(\lambda v.M)V \rhd \sum_{i=1}^{n}(M[V/v]\|(\lambda v.M)V_i)$ if $V \in \mathsf{Val}-\mathsf{TVal}$ and $\mathcal{R}(V,1) = \{V_1, \ldots, V_n\}$,

$(\mu_v)'$ $(\lambda v.M)N \rhd \sum_{i=1}^{n}(\lambda v.M)N_i$ if $N \notin \mathsf{Val}$ and $\mathcal{R}(N,1) = \{N_1, \ldots, N_n\}$,

$(+_{app})'$ $(M+N)L \rhd ML+NL$,

$(\|_{app})'$ $(M\|N)L \rhd ML\|NL$,

$(\|+)'$ $(M+N)\|L \rhd M\|L+N\|L$,

$(\nu)'$ $M \rhd M' \Rightarrow MN \rhd M'N$,

$(+)'$ $M \rhd M' \Rightarrow M+N \rhd M'+N$,

$(\|)'$ $M \rhd M' \Rightarrow M\|N \rhd M'\|N$,

$(+_c)'$ $M+N \rhd N+M$,

$(\|_c)'$ $M\|N \rhd N\|M$,

$(+_{ass})'$ $(M+N)+L \rhd M+(N+L)$ and $M+(N+L) \rhd (M+N)+L$,

$(\|_{ass})'$ $(M\|N)\|L \rhd M\|(N\|L)$ and $M\|(N\|L) \rhd (M\|N)\|L$.

(ii) $\rhd^*$ is the reflexive and transitive closure of $\rhd$.

(iii) $\bowtie$ is the symmetric closure of $\rhd^*$, up to associativity and commutativity of $+$ and $\|$.

3.2.7. PROPOSITION. *The relation $\rhd$ is Church-Rosser, namely*

$$\forall M, M_1, M_2 \in \Lambda^0_{+\|}.\ M \rhd^* M_1 \ \&\ M \rhd^* M_2 \Rightarrow \exists M_3.\ M_1 \rhd^* M_3 \ \&\ M_2 \rhd^* M_3.$$

PROOF. The proof is a variant of the Tait-Martin Löf proof for classical λ-calculus (see [17]). We define the following relation on closed terms

- $M \leadsto M$;
- if $M \rhd M'$ by any clause among $(\beta)'$, $(\beta_v)'$, $(\beta_v\|)'$ and $(\mu_v)'$ then $M \leadsto M'$;
- if $M \leadsto M'$, $N \leadsto N'$ and $L \leadsto L'$ then
  - $(M\|N)L \leadsto M'L\|N'L$,
  - $(M+N)L \leadsto M'L+N'L$,
  - $(M+N)\|L \leadsto M'\|L'+N'\|L'$,
  - $MN \leadsto M'N$,
  - $M+N \leadsto M'+N'$,
  - $M\|N \leadsto M'\|N'$,
  - $M+N \leadsto N'+M'$,
  - $M\|N \leadsto N'\|M'$,
  - $(M+N)+L \leadsto M'+(N'+L')$ and $M'+(N'+L') \leadsto (M'+N')+L'$,
  - $(M\|N)\|L \leadsto M'\|(N'\|L')$ and $M'\|(N'\|L') \leadsto (M'\|N')\|L'$.

By induction on the definition of $\rightsquigarrow$ it is routine to check that it satisfies the diamond property, namely

$$\forall M, M_1, M_2 \in \Lambda^0_{+\|}.\ M \rightsquigarrow M_1\ \&\ M \rightsquigarrow M_2 \Rightarrow \exists M_3.\ M_1 \rightsquigarrow M_3\ \&\ M_2 \rightsquigarrow M_3,$$

hence it is Church-Rosser. Now it is easy to see that $\rightsquigarrow^* = \ \triangleright^*$, from which the thesis follows. $\square$

The relation $\bowtie$ is weaker than the congruence generated by $\triangleright$ but stronger than its reflexive, symmetric and transitive closure. For example $\mathbf{II} \bowtie \mathbf{I}$, but $(\lambda v.v)(\mathbf{II}\|\mathbf{I}) \not\bowtie (\lambda v.v)(\mathbf{I}\|\mathbf{I})$.

3.2.8. Lemma.

(i) *If* $M \bowtie N$ *then for all* L, $ML \bowtie NL$ *and* $M\|L \bowtie N\|L$.

(ii) *If* $M + N \bowtie P + Q$ *then one of the following alternatives is true*

- $M \bowtie P\ \&\ N \bowtie Q$ *or*
- $M \bowtie Q\ \&\ N \bowtie P$ *or*
- $\exists M_0, M_1, N_0, N_1.\ M \bowtie M_0 + M_1\ \&\ N \bowtie N_0 + N_1\ \&\ P \bowtie M_0 + N_0\ \&\ Q \bowtie M_1 + N_1$.

Proof. Part *(i)* is straightforward by induction on $M \bowtie N$.
Part *(ii)* is a consequence of the Church-Rosser property. Indeed if $M + N \bowtie P + Q$ then there are L and L' such that $M + N \ \triangleright^* L$, $P + Q \ \triangleright^* L'$ and L and L' are equal up to commutativity and associativity of $+$ and $\|$. But any sum of the shape $M + N$ can be reduced only to a sum $M' + N'$ where $M \ \triangleright^* M'$ and $N \ \triangleright^* N'$, and similarly for $P + Q$. The thesis then follows. $\square$

The next Lemma connects the relation $\triangleright$ to the reduction trees of terms and hence to the reduction relation $\longrightarrow$.

Notation. From now on we abuse notation writing just $\triangleright$ instead of $\triangleright^*$ (unless otherwise stated).

3.2.9. Lemma. *Let* $M \in \Lambda^0_{+\|}$, *then*

(i) $\mathcal{R}(M, 1) = \{M_1, \ldots, M_n\} \Rightarrow M \ \triangleright \sum_{i=1}^n M_i$,

(ii) $\forall b \in \mathit{bar}(M).\ b = \{M_1, \ldots, M_n\} \Rightarrow M \ \triangleright \sum_{i=1}^n M_i$.

Proof. *(i)* The proof is by induction on $M \in \Lambda^0_{+\|}$.

- If $M \equiv \lambda\chi.M' \in \Lambda^0_{+\|}$ (that is $FV(M') \subseteq \{\chi\}$) then $M \in \mathsf{TVal}$ and $\mathcal{R}(M, 1) = \{M\}$.
- If $M \equiv PQ$ then $P, Q \in \Lambda^0_{+\|}$. We have some subcases. If $P \equiv \lambda\chi.P'$ and either $\chi \equiv x$ or both $\chi \equiv v$ and $Q \in \mathsf{TVal}$, then

$$\mathcal{R}(PQ, 1) = \{P'[Q/\chi]\} \ \text{ and } \ PQ \ \triangleright P'[Q/\chi]$$

by $(\beta)'$ or by $(\beta_v)'$.
Suppose that $\mathcal{R}(Q, 1) = \{Q_1, \ldots, Q_k\}$. If $P \equiv \lambda v.P'$ and $Q \in \mathsf{Val} - \mathsf{TVal}$ then

$$\mathcal{R}((\lambda v.P')Q, 1) = \{P'[Q/v]\|(\lambda v.P')Q_i \mid i \leq k\}$$

and
$$(\lambda v.P')Q \ \rhd \sum_{i=1}^{k} P'[Q/v] \| (\lambda v.P')Q_i$$
by $(\beta_v\|)'$. Otherwise if $Q \notin \mathsf{Val}$ then
$$\mathcal{R}((\lambda v.P')Q,1) = \{(\lambda v.P')Q_i \mid i \leq k\} \text{ and } (\lambda v.P')Q \ \rhd \sum_{i=1}^{k} (\lambda v.P')Q_i$$
by $(\mu_v)'$.
If $P \equiv P_0 \| P_1$ then
$$\mathcal{R}(PQ,1) = \{P_0Q \| P_1Q\} \text{ and } PQ \ \rhd P_0Q \| P_1Q$$
by $(\|_{app})'$.
In all other subcases $P \notin \mathsf{Val} \cup \mathsf{Par}$. Now let $\mathcal{R}(P,1) = \{P_1, \ldots, P_h\}$; hence we have $P \ \rhd \sum_{i=1}^{h} P_i$ by induction hypothesis and
$$PQ \ \rhd \left(\sum_{i=1}^{h} P_i \right) Q \ \rhd \sum_{i=1}^{h} (P_iQ).$$
From this the thesis follows since in these cases $\mathcal{R}(PQ,1) = \{P_iQ \mid i \leq h\}$.

- If $M \equiv P + Q$ then $P, Q \in \Lambda^0_{+\|}$ and the case is trivial since $\mathcal{R}(P+Q,1) = \{P, Q\}$.
- If $M \equiv P \| Q$ then $P, Q \in \Lambda^0_{+\|}$. Again suppose that $\mathcal{R}(P,1) = \{P_1, \ldots, P_h\}$, and $\mathcal{R}(Q,1) = \{Q_1, \ldots, Q_k\}$. Then we have
$$P \| Q \ \rhd \left(\sum_{i=1}^{h} P_i \right) \| \left(\sum_{j=1}^{k} Q_j \right) \ \rhd \sum_{i=1}^{h} \sum_{j=1}^{k} (P_i \| Q_j)$$
by induction hypothesis and clauses $(\|+)'$ and $(\|)'$. So we are done since $\mathcal{R}(PQ,1) = \{P_i \| Q_j \mid i \leq h, j \leq k\}$.

(ii) By induction on the height h of the bar b. If $h = 0$ then the thesis is trivial. Otherwise $tree(M)$ has the root node labeled by M and $tree(M_1), \ldots, tree(M_n)$ as its immediate subtrees, where $\{M_1, \ldots, M_n\} = \mathcal{R}(M,1)$. Because of $h \neq 0$ we have that the root is not in b (recall that a bar intersects each maximal path of $tree(M)$ in exactly one node). It follows that for all $i \leq n$ there exists $b_i \in bar(M_i)$ such that $b = b_1 \cup \cdots \cup b_n$. But the height of each b_i wrt $tree(M_i)$ has to be less than h, so that $M_i \ \rhd \sum \{M'_i \mid M'_i \in b_i\}$ by induction hypothesis. So the thesis follows by *(i)* of this lemma. $\square$

Part *(ii)* of Lemma 3.2.9 implies $M \ \rhd \sum \mathcal{R}(M,k)$ for all k. Moreover it implies that if $M \longrightarrow^* N$ then either $M \ \rhd N$ or $M \ \rhd N + L$ for some L.

Observe that the implications in Lemma 3.2.9 cannot be reversed. This is due to the more permissive clause $(\|)'$ of 3.2.6. Indeed if e.g. $M \equiv (\lambda x.xxx)(\lambda x.xxx)$ then there exists an infinite reduction $M \equiv M_0 \longrightarrow M_1 \longrightarrow \cdots$ where each M_i is an application and $M_i \not\equiv M_{i+1}$ for all i. Now the unique branch of $tree(M_0 \| M_1)$ is the infinite one: $M_0 \| M_1 \longrightarrow M_1 \| M_2 \longrightarrow \cdots$. But $M_0 \| M_1 \ \rhd M_i \| M_i$ for all $i \geq 1$, while for all $b \in bar(M)$, $b \neq \{M_i \| M_i\}$.

3.2.10. Corollary. *If $N \in \Lambda^0_{+\|}$ and $N \Downarrow$, then there exist $V_1, \ldots, V_n \in \mathsf{Val}$ such that* *(i)* $N \rhd \sum_{i=1}^{n} V_i$;

(ii) $\forall (\lambda v.M) \in \Lambda^0_{+\|}.\ (\lambda v.M)N \rhd \sum_{i=1}^{n} (\lambda v.M)V_i$.

PROOF. If $N\Downarrow$ then there exists a bar of values $\{V_1, \dots, V_n\} \in bar(N)$ such that each V_i is the first value that is met starting from the root through a maximal path in $tree(N)$. I.e., no value occurs in the path from N to V_i. By *(ii)* of Lemma 3.2.9 $N \rhd \sum_{i=1}^{n} V_i$. On the other hand $\{(\lambda v.M)V_i \mid i \leq n\} \in bar((\lambda v.M)N)$ because of rule (μ_v). Then the thesis follows by *(ii)* of Lemma 3.2.9. $\square$

3.2.11. LEMMA. *Let* $M, N, V \in \Lambda^0_{+\|}$.

(i) $(\lambda v.M)V\Downarrow\ \&\ V \in \mathsf{Val} \Rightarrow \exists V_1, \dots, V_n \in \mathsf{Val}.\ V \rhd \sum_{i=1}^{n} V_i\ \&\ \forall i \leq n.\ M[V_i/v]\Downarrow$.

(ii) $(\lambda v.M)N\Downarrow \Rightarrow \exists V_1, \dots, V_n \in \mathsf{Val}.\ N \rhd \sum_{i=1}^{n} V_i\ \&\ \forall i \leq n.\ M[V_i/v]\Downarrow$.

PROOF. *(i)* If $V \in \mathsf{TVal}$ then $\mathcal{R}((\lambda v.M)V, 1) = \{M[V/v]\}$, so that the hypothesis implies that $M[V/v]\Downarrow$. Otherwise $V \in \mathsf{Val} - \mathsf{TVal}$. By definition $(\lambda v.M)V \Downarrow_k$ for some $k > 0$ and we make induction on k. Suppose that $\mathcal{R}(V, 1) = \{V_1, \dots, V_n\}$, so that

$$\mathcal{R}((\lambda v.M)V, 1) = \{M[V/v] \| (\lambda v.M)V_i \mid i \leq n\}.$$

If $k = 1$ then, since for all i $(\lambda v.M)V_i$ is an application, that is it is not a value, we have $M[V/v] \in \mathsf{Val}$, so that $M[V/v]\Downarrow$. If $k > 1$ and $M[V/v]\Uparrow$ (otherwise the thesis is immediate), then for all $i \leq n$, $(\lambda v.M)V_i \Downarrow_{k-1}$. By induction there are $V_{i,1}, \dots, V_{i,n_i} \in \mathsf{Val}$ such that $V_i \rhd \sum_{j=1}^{n_i} V_{i,j}$ and $M[V_{i,j}/v]\Downarrow$ for all i and j. The thesis now follows since $V \rhd \sum_{i=1}^{n} V_i$ by *(i)* of Lemma 3.2.9.

(ii) If $N\Uparrow$ then for all $k \geq 0$ there exists $N' \in \mathcal{R}(N, k)$ s.t. $(\lambda v.M)N' \in \mathcal{R}((\lambda v.M)N, k)$ and rules (β_v) and $(\beta_v\|)$ cannot be applied to $(\lambda v.M)N'$. This implies that $(\lambda v.M)N\Uparrow$. By hypothesis and by contraposition we have that $N\Downarrow$. By Corollary 3.2.10 there exist values $V_1, \dots, V_n$ such that $N \rhd \sum_{i=1}^{n} V_i$ and $(\lambda v.M)N \rhd \sum_{i=1}^{n} (\lambda v.M)V_i$. Moreover, by the proof of the same corollary, $\{(\lambda v.M)V_i \mid i \leq n\} \in bar((\lambda v.M)N)$, so that by hypothesis $(\lambda v.M)V_i\Downarrow$ for all $i \leq n$. Now, by part *(i)* of this lemma, for each i there are $V_{i,1}, \dots, V_{i,n_i} \in \mathsf{Val}$ such that $V_i \rhd \sum_{j=1}^{n_i} V_{i,j}$ and $M[V_{i,j}/v]\Downarrow$, and the thesis follows. $\square$

3.2.12. THEOREM. *Let* $M, N \in \Lambda^0_{+\|}$, *then*

(i) $[M \rhd N\ \&\ N\Downarrow] \Rightarrow M\Downarrow$.

(ii) $M \bowtie N \Rightarrow [M\Downarrow \Leftrightarrow N\Downarrow]$.

PROOF. *(i)* In this proof we must distinguish between $\rhd$ and $\rhd^*$. Clearly, if we can prove the statement for $\rhd$, the same thesis holds for $\rhd^*$. As a matter of fact we prove, by induction on the definition of $\rhd$, the stronger statement

$$M \rhd N \Rightarrow \forall \vec{L}.\ [N\vec{L}\Downarrow \Rightarrow M\vec{L}\Downarrow]$$

from which the thesis follows taking the empty vector.

- If $M \rhd N$ thanks to $(\beta)'$, $(\beta_v)'$, $(\beta_v\|)'$, $(\mu_v)'$, $(+_{app})'$ or $(\|_{app})'$ (see Definition 3.2.6), then M is always an application and $N \equiv \sum_{i=1}^{n} M_i$ where $\mathcal{R}(M,1) = \{M_1, \ldots, M_n\}$ (where the multiset $\mathcal{R}(M,1)$ is ordered in such a way that it matches the shape of N). By this fact and rules (ν) and $(+)$ we have that

$$\{M_i\vec{L} \mid 1 \leq i \leq n\} \in bar(N\vec{L}) \cap bar(M\vec{L}).$$

Now $N\vec{L}\Downarrow$ implies that $M_i\vec{L}\Downarrow$ for all i $(1 \leq i \leq n)$, so that $M\vec{L}\Downarrow$ follows.

- Clause $(\|+)'$. Then $M \equiv (P+Q)\|R$ and $N \equiv P\|R + Q\|R$. Now

$$\begin{array}{rcl} (P\|R+Q\|R)\vec{L}\Downarrow & \Rightarrow & (P\vec{L}\Downarrow \ \& \ Q\vec{L}\Downarrow) \text{ or } R\vec{L}\Downarrow \\ & \Rightarrow & (P\vec{L}\Downarrow \ \text{ or } R\vec{L}\Downarrow) \ \& \ (Q\vec{L}\Downarrow \ \text{ or } R\vec{L}\Downarrow) \\ & \Rightarrow & ((P+Q)\|R)\vec{L}\Downarrow \end{array}$$

by rule (ν) and the remark after Definition 3.2.5.

- Clause $(\nu)'$. Then $M \equiv PQ$, $N \equiv P'Q$ and $P \rhd P'$. In this case $P'Q\vec{L}\Downarrow$ implies $PQ\vec{L}\Downarrow$ immediately by induction, taking the vector $Q\vec{L}$.

- Clause $(+)'$. Then $M \equiv P+Q$, $N \equiv P'+Q$ and $P \rhd P'$. Now

$$\begin{array}{rcll} (P'+Q)\vec{L}\Downarrow & \Rightarrow & P'\vec{L}\Downarrow \ \& \ Q\vec{L}\Downarrow & \\ & \Rightarrow & P\vec{L}\Downarrow \ \& \ Q\vec{L}\Downarrow & \text{by induction} \\ & \Rightarrow & (P+Q)\vec{L}\Downarrow\,. & \end{array}$$

- Clause $(\|)'$. Similar to the case of clause $(+)'$ where "or" replaces "&".

- For clauses $(+_c)'$, $(\|_c)'$, $(+_{ass})'$ and $(\|_{ass})'$ the proofs are similar to those of $(+)'$ and $(\|)'$.

(ii) M converges implies that there are values $V_1, \ldots, V_n$ such that $M \rhd \sum_{i=1}^{n} V_i$ by 3.2.10*(i)*. Therefore $N \bowtie \sum_{i=1}^{n} V_i$. By the Church Rosser property of $\rhd$ there is an L such that $N \rhd L$ and $\sum_{i=1}^{n} V_i \rhd L$. But $\sum_{i=1}^{n} V_i \rhd L$ implies that L is a sum of values and therefore L must converge. We conclude that N converges by *(i)*. □

Based on the convergence predicate the following definition adapts to the present setting the notion of contextual theories. This notion stems from [75] and it is widely used e.g. in [17] for the classical theory of solvability and in [6], [22] and [78], where it is shown to be equivalent to applicative bisimulation.

The idea is that two terms are operationally equivalent if and only if in all contexts they exhibit the same behavior with respect to some observable properties. Here convergence is the only observable, hence we can put

3.2.13. Definition. Let $M, N \in \Lambda_{+\|}$. Then

(i) $M \sqsubseteq^{\mathcal{O}} N \ \Leftrightarrow\ \forall C[\,].\ C[M]\Downarrow \ \Rightarrow\ C[N]\Downarrow$, where $C[M], C[N] \in \Lambda^0_{+\|}$.

(ii) $\simeq^{\mathcal{O}} \; = \; \sqsubseteq^{\mathcal{O}} \cap \sqsupseteq^{\mathcal{O}}$.

3.3. A Logical Presentation

To obtain a logical presentation of the semantics of the calculus we follow the paradigm of Leibniz which identifies objects with sets of their properties. This received an elegant mathematical treatment thanks to works like [89] and [5] and, especially in the case of type-free calculi, it is naturally formalized in suitable extensions of Curry type assignment system like the intersection type discipline considered in [18].

In the present case we use a more expressive system which allows for disjunctive types. We call them union types since they differ from coproducts in Church typed λ-calculi much in the same way as intersection differs from cartesian product. See [16] for a study of this discipline in case of classical λ-calculus.

3.3.1. The Set of Types and its Preorder

The type syntax is as follows

$$\sigma ::= \omega \mid \sigma \to \sigma \mid \sigma \wedge \sigma \mid \sigma \vee \sigma$$

and we call *Type* the resulting set. In writing types, we assume that $\wedge$ and $\vee$ take precedence over $\to$.

The choice of the preorder on types is crucial, since it will be used in a subtyping rule in subsection 3.3.2 and it will determine the structure of the set of filters in section 3.4.

3.3.1. DEFINITION. Let $\sigma \leq \tau$ be the smallest preorder over types such that

(i) $\langle Type, \leq \rangle$ is a distributive lattice, in which $\wedge$ is the meet, $\vee$ is the join and ω is the top;

(ii) the arrow satisfies

  (a) $\sigma \to \omega \leq \omega \to \omega$;

  (b) $(\sigma \to \rho) \wedge (\sigma \to \tau) \leq \sigma \to \rho \wedge \tau$;

  (c) $\sigma \geq \sigma', \tau \leq \tau' \Rightarrow \sigma \to \tau \leq \sigma' \to \tau'$.

Following [39] by lattice we mean a poset in which every finite non empty subset has a meet and a join. According to this definition, there are lattices without bottom (like the present one).

We write $\sigma = \tau$ for "$\sigma \leq \tau$ and $\tau \leq \sigma$". Note that, if $\sigma \neq \omega$ then $\sigma \leq \omega \to \omega$.

NOTATION. Let $\omega^0 \to \omega = \omega$, $\omega^{n+1} \to \omega = \omega \to \omega^n \to \omega$.

The types $\omega^n \to \omega$ for suitable n are "better than" all other types, as shown in the following Proposition.

3.3.2. PROPOSITION. *For all σ, there exists n such that $\omega^n \to \omega \leq \sigma$.*

PROOF. By induction on the structure of σ.

$\sigma \equiv \omega$**.** Trivial.

$\sigma \equiv \sigma_1 \to \sigma_2$**.** By induction hypothesis $\exists n. \omega^n \to \omega \leq \sigma_2$, hence $\omega^{n+1} \to \omega \leq \sigma$, by Definition 3.3.1.*(ii.c)*, using $\sigma_1 \leq \omega$.

$\sigma \equiv \sigma_1 \wedge \sigma_2$**.** By induction hypothesis $\exists n_i. \omega^{n_i} \to \omega \leq \sigma_i$ $(i = 1, 2)$. Let $n = \max(n_1, n_2)$. Then $\omega^n \to \omega \leq \sigma$, since $\omega^n \to \omega \leq \omega^{n_i} \to \omega$ $(i = 1, 2)$ and $\sigma_1 \wedge \sigma_2$ is the meet of σ_1 and σ_2.

$\sigma \equiv \sigma_1 \vee \sigma_2$. Recall that $\sigma_1 \wedge \sigma_2 \leq \sigma_1 \vee \sigma_2$, and then proceed as in the previous case. $\square$

We need some properties of the $\leq$ relation, whose proof requires a stratification of $Type$, to be compared with the stratification of Definition 2.5.2.

3.3.3. DEFINITION. (Stratification of $Type$)
Let us define three subsets T_0, T_1, T_2 of $Type$ recursively
$\omega \to \omega \in T_0$;
$\omega \in T_2$;
$\sigma \in T_2, \tau \in T_1 \Rightarrow \sigma \to \tau \in T_0$;
$n \geq 1, \sigma_1, \ldots, \sigma_n \in T_0 \Rightarrow \sigma_1 \vee \ldots \vee \sigma_n \in T_1$;
$n \geq 1, \sigma_1, \ldots, \sigma_n \in T_1 \Rightarrow \sigma_1 \wedge \ldots \wedge \sigma_n \in T_2$.

To rephrase the previous definition, we consider types in conjunctive normal form, that is conjunctions of disjunctions of arrows, ω being the empty conjunction.
Taking $n = 1$ in the clauses above, one sees that $T_0 \subseteq T_1 \subseteq T_2$, and such inclusions are clearly proper.
Over each of these sets we introduce a preorder.

3.3.4. DEFINITION. $\leq_i \subseteq T_i \times T_i$ is the least preorder such that

$(\leq_0)$: $\sigma \leq_0 \tau \;\Leftrightarrow\; \tau \equiv \omega \to \omega$ or $\sigma \equiv \sigma' \to \sigma'', \tau \equiv \tau' \to \tau''$ and $\tau' \leq_2 \sigma'$ and $\sigma'' \leq_1 \tau''$;

$(\leq_1)$: $\sigma_1 \vee \ldots \vee \sigma_n \leq_1 \tau_1 \vee \ldots \vee \tau_m \;\Leftrightarrow\; \forall i \leq n\, \exists j \leq m.\, \sigma_i \leq_0 \tau_j$;

$(\leq_2)$: $\sigma \leq_2 \tau \;\Leftrightarrow\; \tau \equiv \omega$ or $\sigma \equiv \sigma_1 \wedge \ldots \wedge \sigma_n$, $\tau \equiv \tau_1 \wedge \ldots \wedge \tau_m$ and $\forall j \leq m\, \exists i \leq n.\, \sigma_i \leq_1 \tau_j$.

Really, for each type in $Type$, we can find an equivalent type in T_2; therefore we introduce a map which associates to each type its equivalent type in T_2.

3.3.5. DEFINITION. Let $* : Type \to T_2$ be defined by

$$\omega^* = \omega$$

$$(\sigma \to \tau)^* = \begin{cases} \bigwedge_{i \in I}(\sigma^* \to \tau_i) & \text{if } \tau^* \equiv \bigwedge_{i \in I} \tau_i \text{ and } \tau^* \not\equiv \omega \\ \omega \to \omega & \text{otherwise} \end{cases}$$

$$(\sigma \vee \tau)^* = \begin{cases} \bigwedge_{i \in I} \bigwedge_{j \in J}(\sigma_i \vee \tau_j) & \text{if } \sigma^* \equiv \bigwedge_{i \in I} \sigma_i,\ \sigma^* \not\equiv \omega \text{ and } \tau^* \equiv \bigwedge_{j \in J} \tau_j,\ \tau^* \not\equiv \omega \\ \omega & \text{otherwise} \end{cases}$$

$$(\sigma \wedge \tau)^* = \begin{cases} \sigma^* & \text{if } \tau^* \equiv \omega \\ \tau^* & \text{if } \sigma^* \equiv \omega \\ \sigma^* \wedge \tau^* & \text{otherwise.} \end{cases}$$

As in previous chapter, the stratification and the mapping $*$ can be used to prove the following properties of the pre-order on types.

3.3.6. PROPOSITION. *For all* $\sigma, \tau \in Type$

(i) $\sigma = \sigma^*$;

(ii) $\sigma, \tau \in T_i, \sigma \leq_i \tau \Rightarrow \sigma \leq \tau$ *for* $i = 0, 1, 2$;

(iii) $\sigma \leq \tau \Rightarrow \sigma^* \leq_2 \tau^*$.

3.3.7. LEMMA.

(i) $\mu \wedge \nu \leq \sigma \to \tau$ & $\mu \neq \omega$ & $\nu \neq \omega \Rightarrow \exists \tau_1, \tau_2.\, \tau = \tau_1 \wedge \tau_2$ & $\mu \leq \sigma \to \tau_1$ & $\nu \leq \sigma \to \tau_2$;

(ii) $\bigwedge_{i \in I}(\mu_i \to \nu_i) \leq \sigma \to \tau$ & $\tau \neq \omega \Rightarrow \exists J \subseteq I.\, \sigma \leq \bigwedge_{j \in J} \mu_j$ & $\bigwedge_{j \in J} \nu_j \leq \tau$.

3.3.1. Remark. Notice that Lemma 3.3.7 cannot be trivially satisfied by choosing $\tau_1 = \tau_2 = \tau$. In fact in general $\mu \wedge \nu \leq \sigma \to \tau$ does not imply $\mu \leq \sigma \to \tau$. For a counter-example take $\mu = \sigma = \tau = \omega \to \omega$ and $\nu = \sigma \to \tau$.

A type σ is *join irreducible* or *coprime* if and only if

$$\sigma \leq \tau \vee \rho \Rightarrow \sigma \leq \tau \text{ or } \sigma \leq \rho$$

for any τ, ρ. Let $CType$ be the set of coprime types different from ω. Observe that, because of distributivity, coprime types are closed under $\wedge$. Being $\langle Type, \leq \rangle$ the free distributive lattice satisfying the arrow axioms, each type is the join of a finite number of coprime types. To see this, it suffices to define the following mapping $\Theta : Type \to \mathcal{P}(CType)$

$$\begin{array}{lcl} \Theta(\omega) & = & \{\omega\} \\ \Theta(\sigma \to \tau) & = & \{\sigma \to \tau\} \\ \Theta(\sigma \wedge \tau) & = & \{\sigma' \wedge \tau' \mid \sigma' \in \Theta(\sigma) \;\&\; \tau' \in \Theta(\tau)\} \\ \Theta(\sigma \vee \tau) & = & \Theta(\sigma) \cup \Theta(\tau). \end{array}$$

If $\Theta(\sigma) = \{\sigma_1, \ldots, \sigma_n\}$, it is easy to verify that σ_i is join irreducible for each i and $\sigma = \sigma_1 \vee \cdots \vee \sigma_n$.

3.3.2. The Type Assignment System

In this subsection we introduce our type assignment system $\mathcal{L}$. We start with the notion of basis. We state that only coprime types different from ω can be assumed for call-by-value variables. This restriction is justified by the correspondence between total values and coprime types (see Theorem 3.3.12*(ii)*).

3.3.8. Definition. A *basis* $\Gamma : (\mathsf{Vn} \to Type) \cap (\mathsf{Vv} \to CType)$ is a mapping such that $\Gamma(x) = \omega$ for all x but a finite subset of Vn and $\Gamma(v) = \omega \to \omega$ for all v but a finite subset of Vv.

To each basis Γ we associate the finite set

$$Dom(\Gamma) = \{x \in \mathsf{Vn} \mid \Gamma(x) \neq \omega\} \cup \{v \in \mathsf{Vv} \mid \Gamma(v) \neq \omega \to \omega\}.$$

The notation $\Gamma, \chi : \sigma$ is a shorthand for the function $\Gamma'(\chi') = \sigma$ if $\chi' \equiv \chi$, $\Gamma(\chi')$ otherwise. To meet a common practice we shall sometime identify Γ with the (finite) set of judgments $\{\chi{:}\sigma \mid \chi \in Dom(\Gamma) \;\&\; \Gamma(\chi) = \sigma\}$ and write $\chi{:}\sigma \in \Gamma$.

3.3.9. Definition. The axioms and rules of the assignment system $\mathcal{L}$ are the following

$$(\mathrm{Ax})\quad \Gamma \vdash \chi : \Gamma(\chi) \qquad\qquad (\omega)\quad \Gamma \vdash M : \omega$$

$$(\to \mathrm{I}_n)\ \frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad (\to \mathrm{I}_v)\ \frac{\Gamma, v:\sigma' \vdash M:\tau \quad \forall \sigma' \in \Theta(\sigma)}{\Gamma \vdash \lambda v.M : \sigma \to \tau}$$

$$(\to \mathrm{E})\ \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

$$(\wedge \mathrm{I})\ \frac{\Gamma \vdash M:\sigma \qquad \Gamma \vdash M:\tau}{\Gamma \vdash M : \sigma \wedge \tau} \qquad (\leq)\ \frac{\Gamma \vdash M:\sigma \quad \sigma \leq \tau}{\Gamma \vdash M:\tau}$$

$$(+\mathrm{I})\ \frac{\Gamma \vdash M:\sigma \qquad \Gamma \vdash N:\tau}{\Gamma \vdash M+N : \sigma \vee \tau} \qquad (\| \mathrm{I})\ \frac{\Gamma \vdash M:\sigma \qquad \Gamma \vdash N:\tau}{\Gamma \vdash M\|N : \sigma \wedge \tau}.$$

$\Gamma \vdash M:\sigma$ abbreviates "$\Gamma \vdash M:\sigma$ is derivable in $\mathcal{L}$".

Rule $(\to \mathrm{I}_v)$ models call-by-value abstractions. To give a hint for understanding it we consider the following example. Let W_1, W_2 be total values such that $\vdash W_i{:}\sigma_i$ $(i = 1, 2)$ for some coprime types σ_1, σ_2. Clearly this implies $\vdash W_1 + W_2{:}\sigma_1 \vee \sigma_2$ by rule $(+\mathrm{I})$. Consider $(\lambda v.M)(W_1 + W_2)$: it reduces to $M[W_1/v]$ and $M[W_2/v]$. Therefore $v{:}\sigma_i \vdash M{:}\tau$ for $i = 1, 2$ suffices to assure that $(\lambda v.M)$ has type $\sigma_1 \vee \sigma_2 \to \tau$. The real justification of this rule is that it implies the completeness of the type assignment (Theorem 3.4.11) and the full abstraction of the filter model (Theorem 3.5.11).

We shall write $\Gamma \leq \Gamma'$ if $\forall \chi.\ \Gamma(\chi) \leq \Gamma'(\chi)$: in this case it is easy to verify that, if $\Gamma' \vdash M:\sigma$, then $\Gamma \vdash M:\sigma$ for any M and σ.

The system $\mathcal{L}$ enjoys structural properties which can be shown by simple inductions on derivations.

3.3.10. Theorem (Derivability properties of system $\mathcal{L}$).

(i) $\Gamma \vdash \chi:\tau \;\Leftrightarrow\; \Gamma(\chi) \leq \tau$;

(ii) $\Gamma \vdash \lambda\chi.M:\rho \;\Leftrightarrow$
$\exists n,\ \sigma_1,\ldots,\sigma_n,\tau_1,\ldots\tau_n.(\forall i \leq n.\ \Gamma \vdash \lambda\chi.M:\sigma_i \to \tau_i)\ \&\ \bigwedge_{i=1}^{n}(\sigma_i \to \tau_i) \leq \rho$;

(iii) $\Gamma \vdash \lambda x.M:\sigma \to \tau \;\Leftrightarrow\; \Gamma, x:\sigma \vdash M:\tau$;

(iv) $\Gamma \vdash \lambda v.M:\sigma \to \tau\ \&\ \sigma \neq \omega \;\Leftrightarrow\; \forall \sigma' \in \Theta(\sigma).\ \Gamma, v:\sigma' \vdash M:\tau$;

(v) $\Gamma \vdash \lambda v.M:\sigma \to \tau\ \&\ \sigma = \omega \Rightarrow \tau = \omega$;

(vi) $\Gamma \vdash MN:\tau\ \&\ \tau \neq \omega \;\Leftrightarrow\; \exists \sigma.\ \Gamma \vdash M:\sigma \to \tau\ \&\ \Gamma \vdash N:\sigma$;

(vii) $\Gamma \vdash M+N:\sigma \;\Leftrightarrow\; \Gamma \vdash M:\sigma\ \&\ \Gamma \vdash N:\sigma$;

(viii) $\Gamma \vdash M\|N:\tau \;\Leftrightarrow\; \exists \sigma,\sigma'.\ \Gamma \vdash M:\sigma\ \&\ \Gamma \vdash N:\sigma'\ \&\ \sigma \wedge \sigma' \leq \tau$.

Proof. We consider only the interesting cases.

(ii) Given a derivation of $\Gamma \vdash \lambda\chi.M{:}\rho$, let

$$\Gamma \vdash \lambda\chi.M{:}\sigma_1 \to \tau_1, \ldots, \Gamma \vdash \lambda\chi.M{:}\sigma_n \to \tau_n$$

be all the statements in this deduction on which $\Gamma \vdash \lambda\chi.M{:}\rho$ depends and which are conclusions of rule $(\to \mathrm{I}_n)$ or of rule $(\to \mathrm{I}_v)$. Then

$$(\sigma_1 \to \tau_1) \wedge \cdots \wedge (\sigma_n \to \tau_n) \leq \rho.$$

(iii) If $\tau = \omega$ it is trivial. Otherwise let $\sigma_1, \ldots, \sigma_n, \tau_1, \ldots, \tau_n$ be as in the proof of *(ii)* where ρ has been replaced by $\sigma \to \tau$. Then

$$(\sigma_1 \to \tau_1) \wedge \cdots \wedge (\sigma_n \to \tau_n) \leq \sigma \to \tau$$

which implies, by Lemma 3.3.7*(ii)*,

$$\exists J \subseteq \{1, \cdots, n\}.\ \sigma \leq \bigwedge_{j \in J} \sigma_j \ \&\ \bigwedge_{j \in J} \tau_j \leq \tau.$$

Moreover $\Gamma, x{:}\sigma_i \vdash M : \tau_i$ for $1 \leq i \leq n$, so that one can conclude $\Gamma, x{:}\sigma \vdash M{:}\tau$.

(iv) Let $\sigma_1, \ldots, \sigma_n, \tau_1, \ldots, \tau_n$ be as in the proof of *(ii)* where ρ has been replaced by $\sigma \to \tau$. Similarly to case *(iii)* we have

$$\exists J \subseteq \{1, \cdots, n\}.\ \sigma \leq \bigwedge_{j \in J} \sigma_j \ \&\ \bigwedge_{j \in J} \tau_j \leq \tau.$$

Moreover $\Gamma, v{:}\sigma'_i \vdash M : \tau_i$ for all $\sigma'_i \in \Theta(\sigma_i)$ and for $1 \leq i \leq n$. Now $\sigma \leq \sigma_j$ implies $\forall \sigma' \in \Theta(\sigma)\ \exists \sigma'_j \in \Theta(\sigma_j)$ such that $\sigma' \leq \sigma'_j$ by definition of coprimality. So we can conclude $\forall \sigma' \in \Theta(\sigma).\ \Gamma, x{:}\sigma' \vdash M{:}\tau$.

(v) We assume ad absurdum that $\tau \neq \omega$. Then, if $\sigma_1, \ldots, \sigma_n, \tau_1, \ldots, \tau_n$ and J are as in *(iv)*, we would have $\sigma_j = \omega$ for all $j \in J$, and this is impossible according to our definition of basis.

(vii) Again, given a deduction of $\Gamma \vdash M + N{:}\sigma$, let

$$\Gamma \vdash M + N{:}\sigma_1, \ldots, \Gamma \vdash M + N{:}\sigma_n$$

be all the statements in this deduction on which $\Gamma \vdash M + N{:}\sigma$ depends and which are conclusions of rule (+I). Then $\sigma_1 \wedge \cdots \wedge \sigma_n \leq \sigma$ and there are μ_i, ν_i such that $\sigma_i = \mu_i \vee \nu_i$, $\Gamma \vdash M{:}\mu_i$, $\Gamma \vdash N{:}\nu_i$, for $1 \leq i \leq n$. So we can deduce $\Gamma \vdash M{:}\sigma$ and $\Gamma \vdash N{:}\sigma$ using ($\wedge$ I) and ($\leq$).

(viii) Finally, given a deduction of $\Gamma \vdash M \| N{:}\tau$, let

$$\Gamma \vdash M \| N{:}\sigma_1, \ldots, \Gamma \vdash M \| N{:}\sigma_n$$

be all the statements in this deduction on which $\Gamma \vdash M \| N{:}\tau$ depends and which are conclusions of rule ($\|$I). Then $\sigma_1 \wedge \cdots \wedge \sigma_n \leq \tau$ and there are μ_i, ν_i such that $\sigma_i = \mu_i \wedge \nu_i$, $\Gamma \vdash M{:}\mu_i$, $\Gamma \vdash N{:}\nu_i$, for $1 \leq i \leq n$. Then we can choose $\sigma = \bigwedge_{i \leq n} \mu_i$, and $\sigma' = \bigwedge_{i \leq n} \nu_i$. In fact $\sigma \wedge \sigma' \leq \tau$ and we can derive $\Gamma \vdash M{:}\sigma$ and $\Gamma \vdash N{:}\sigma'$ using ($\wedge$ I). $\Box$

As immediate consequence of 3.3.10*(iv)* we have the co-additivity of call-by-value abstraction (i.e. finite meets are preserved).

3.3.11. Corollary. $\Gamma \vdash \lambda v.M : (\sigma \to \rho) \wedge (\tau \to \rho) \ \Rightarrow\ \Gamma \vdash \lambda v.M : \sigma \vee \tau \to \rho.$

We show how types characterize partial values and total values.

3.3.12. Theorem (Characterization of values).

(i) $V \in \mathsf{Val} \Rightarrow \vdash V : \omega \to \omega$;

(ii) $W \in \mathsf{TVal}$ & $\Gamma \vdash W : \sigma \Rightarrow \exists \sigma' \in \Theta(\sigma).\ \Gamma \vdash W : \sigma'$.

PROOF. *(i)* By induction on the definition of values. If $V \equiv v \in \mathsf{Vv}$ then $\Gamma(v) = \omega \to \omega$, since $Dom(\Gamma) = \emptyset$, and the thesis follows by (Ax). If $V \equiv \lambda x.M$ then $\Gamma, x : \omega \vdash M : \omega$ is derivable, by rule (ω); hence the thesis using $(\to \mathrm{I}_n)$. If $V \equiv \lambda v.M$ we do the same as before but assuming $v : \omega \to \omega$. The thesis follows using $(\to \mathrm{I}_v)$ and $(\leq)$. Finally if $V \equiv V' \| M$ or $M \| V'$ the thesis follows by induction using $(\|)$ and $(\leq)$.
(ii) By induction on the definition of total values. If $W \equiv v$ then the thesis follows by 3.3.10*(i)* and the definition of basis. If $W \equiv \lambda \chi.M$ then, noting that each arrow type is coprime, the thesis follows from 3.3.10*(ii)* and the closure of coprime types under $\wedge$. Finally if $W \equiv W' \| W''$, by 3.3.10*(viii)* there exist μ and ν such that $\Gamma \vdash W' : \mu$, $\Gamma \vdash W'' : \nu$ and $\mu \wedge \nu \leq \sigma$; now by induction there are $\mu' \in \Theta(\mu), \nu' \in \Theta(\nu)$ such that $\Gamma \vdash W' : \mu'$ and $\Gamma \vdash W'' : \nu'$. Since coprime types are closed under $\wedge$, $\mu' \wedge \nu' \leq \sigma$ implies that there is $\sigma' \in \Theta(\sigma)$ such that $\mu' \wedge \nu' \leq \sigma'$. □

The following Lemma states the substitution properties of terms.

3.3.13. LEMMA.

(i) $\Gamma \vdash M[N/x] : \tau \Leftrightarrow \exists \sigma.\ \Gamma, x : \sigma \vdash M : \tau$ & $\Gamma \vdash N : \sigma$;

(ii) $\Gamma \vdash M[V/v] : \tau$ & $V \in \mathsf{Val} \Rightarrow \exists \sigma\ \forall \sigma' \in \Theta(\sigma).\ \Gamma, v : \sigma' \vdash M : \tau$ & $\Gamma \vdash V : \sigma$;

(iii) $\Gamma \vdash M[W/\chi] : \tau$ & $W \in \mathsf{TVal} \Leftrightarrow \exists \sigma \in CType.\ \Gamma, \chi : \sigma \vdash M : \tau$ & $\Gamma \vdash W : \sigma$.

PROOF. *(i)* $(\Rightarrow)$. If x does not occur in M we can choose $\sigma \equiv \omega$. Otherwise let σ be the intersection of all predicates of statements with subject N which occur in a given deduction of $\Gamma \vdash M[N/x] : \tau$. The proof of $\Leftarrow$ is standard.

(ii). If v does not occur in M we can choose $\sigma \equiv \omega \to \omega$. Otherwise let ρ be the intersection of all predicates of statements with subject V which occur in a given deduction of $\Gamma \vdash M[V/v] : \tau$. If $\rho \neq \omega$ we can choose $\sigma \equiv \rho$, otherwise $\sigma \equiv \omega \to \omega$.

(iii)$(\Rightarrow)$. If χ does not occur in M we can choose $\sigma \equiv \omega \to \omega$. Otherwise let ρ be the intersection of all predicates of statements with subject W which occur in a given deduction of $\Gamma \vdash M[W/\chi] : \tau$. By 3.3.12*(ii)* there is $\rho' \in \Theta(\rho)$ such that $\Gamma \vdash W : \rho'$. If $\rho' \neq \omega$ we can choose $\sigma \equiv \rho'$, otherwise $\sigma \equiv \omega \to \omega$. The proof of $\Leftarrow$ is standard. □

Notice that in 3.3.13*(ii)* the "$\Rightarrow$" cannot be replaced by "$\Leftrightarrow$". An easy proof of this uses the characterization of divergent terms by types which will be given in 3.5.6*(ii)*. So we will prove it in Corollary 3.5.7*(i)*.

As an immediate consequence of 3.3.12*(ii)* and 3.3.13*(iii)* the following rule $(\vee \mathrm{E})$ is admissible

$$(\vee \mathrm{E}) \quad \frac{\forall \sigma' \in \Theta(\sigma) \quad \Gamma, \chi : \sigma' \vdash M : \rho \qquad \Gamma \vdash W : \sigma \qquad W \in \mathsf{TVal}}{\Gamma \vdash M[W/\chi] : \rho}$$

Therefore, the restriction over the basis can be relaxed, allowing $\Gamma(v)$ to be any type different from ω. This would have the advantage of having a unique rule for abstraction, i.e. the standard one, avoiding $(\to \mathrm{I}_v)$ which is a rule schema. Of course rule $(\vee \mathrm{E})$ should be added in this case. The reason why we choose the present less elegant version is that it greatly simplifies proofs.

3.3.3. The Logic Congruence Relation

We introduce now the logical equivalence $\simeq^{\mathcal{L}}$; thereafter we shall use the properties stated in Theorem 3.3.10 to establish the basic (in)-equalities holding under this notion of equivalence. The invariance of types with respect to $\bowtie$ and to the reduction relation studied in section 3.2 will follow.

3.3.14. Definition. Let $M, N \in \Lambda_{+\|}$, then

(i) $M \sqsubseteq^{\mathcal{L}} N \;\Leftrightarrow\; \forall\Gamma,\sigma.\, \Gamma \vdash M : \sigma \Rightarrow \Gamma \vdash N : \sigma$;

(ii) $\simeq^{\mathcal{L}} = \sqsubseteq^{\mathcal{L}} \cap \sqsupseteq^{\mathcal{L}}$.

As a first step in the study of the relation $\simeq^{\mathcal{L}}$ we fix some basic properties of it with respect to the various kinds of β-contraction present in our calculus. These can be easily proved using 3.3.13.

3.3.15. Lemma.

(i) $(\lambda x.M)N \simeq_{\mathcal{L}} M[N/x]$;

(ii) $M[V/v] \sqsubseteq^{\mathcal{L}} (\lambda v.M)V$ *if* $V \in \mathsf{Val}$;

(iii) $(\lambda v.M)W \simeq^{\mathcal{L}} M[W/v]$ *if* $W \in \mathsf{TVal}$.

Proof. The most interesting case is the inclusion from left to right of *(iii)* when $\tau \neq \omega$.

$$\begin{array}{lll} & \Gamma \vdash (\lambda v.M)W : \tau & \\ \Rightarrow & \exists\sigma.\, \Gamma \vdash \lambda v.M : \sigma \to \tau \;\&\; \Gamma \vdash W : \sigma & \text{by 3.3.10}(vi) \\ \Rightarrow & \exists\sigma.\, \forall\sigma' \in \Theta(\sigma).\, \Gamma, v : \sigma' \vdash M : \tau \;\&\; \Gamma \vdash W : \sigma & \text{by 3.3.10}(iv) \\ \Rightarrow & \exists\sigma' \in CType.\, \Gamma, v : \sigma' \vdash M : \tau \;\&\; \Gamma \vdash W : \sigma' & \text{by 3.3.12}(ii) \text{ since } W \in \mathsf{TVal} \\ \Rightarrow & \Gamma \vdash M[W/v] : \tau & \text{by 3.3.13}(iii)(\Leftarrow). \qquad \square \end{array}$$

Notice that the opposite of Lemma 3.3.15*(ii)* does not hold. This will be proved in Corollary 3.5.7*(ii)*, since it follows immediately from point *(i)* of the same Corollary.

The following three lemmas are easy consequences of 3.3.10. The second and third lemmas state that non-deterministic choice and parallel composition are the meet and the join respectively. Moreover they illustrate the behaviors of these operators with respect to application and abstraction.

3.3.16. Lemma. *The relation* $\simeq^{\mathcal{L}}$ *is a congruence over* $\Lambda_{+\|}$.

3.3.17. Lemma.

(i) $M + N \sqsubseteq^{\mathcal{L}} M, N$;

(ii) $L \sqsubseteq^{\mathcal{L}} M, N \Rightarrow L \sqsubseteq^{\mathcal{L}} M + N$;

(iii) $(M + N)L \simeq^{\mathcal{L}} ML + NL$;

(iv) $L(M + N) \sqsubseteq^{\mathcal{L}} LM + LN$;

(v) $(\lambda v.M)(N + L) \simeq^{\mathcal{L}} (\lambda v.M)N + (\lambda v.M)L$;

(vi) $\lambda\chi.(M + N) \sqsubseteq^{\mathcal{L}} \lambda\chi.M + \lambda\chi.N$.

PROOF. All inclusions are immediate. The converse of *(vi)* does not hold. Indeed, let $\sigma \equiv (\rho \to \rho) \vee (\tau \to \omega^2 \to \omega)$ where $\rho \equiv \omega^3 \to \omega$ and $\tau \equiv (\omega \to \omega) \to \omega^2 \to \omega$. Then we have $\vdash \mathbf{I} : \rho \to \rho$ and $\vdash \Delta : \tau \to \omega^2 \to \omega$, which imply $\vdash \mathbf{I} + \Delta : \sigma$, but $\not\vdash \lambda x.(x + xx) : \sigma$. In fact, by Theorem 3.3.10*(iii)* and *(vii)*, if we could derive $\lambda x.(x + xx) : \sigma$, then we would also have $x : \mu \vdash x : \nu$ and $x : \mu \vdash xx : \nu$ for some μ, ν such that $\mu \to \nu \leq \sigma$. This implies either $\mu \to \nu \leq \rho \to \rho$ or $\mu \to \nu \leq \tau \to \omega^2 \to \omega$ by Definition 3.3.4 and Proposition 3.3.6. But it is easy to verify, using Theorem 3.3.10*(i)* and *(vi)*, that $x : \rho \not\vdash xx : \rho$ and $x : \tau \not\vdash x : \omega^2 \to \omega$. □

3.3.18. LEMMA.

(i) $M, N \sqsubseteq^{\mathcal{L}} M \| N$;

(ii) $M, N \sqsubseteq^{\mathcal{L}} L \Rightarrow M \| N \sqsubseteq^{\mathcal{L}} L$;

(iii) $(M \| N)L \simeq^{\mathcal{L}} ML \| NL$;

(iv) $LM \| LN \sqsubseteq^{\mathcal{L}} L(M \| N)$;

(v) $\lambda\chi.(M \| N) \simeq^{\mathcal{L}} \lambda\chi.M \| \lambda\chi.N$;

(vi) $(M + N) \| L \simeq^{\mathcal{L}} M \| L + N \| L$.

The inequalities 3.3.17*(iv)* and 3.3.18*(iv)* are proper, and this can be proved using the structural properties of deductions (Theorem 3.3.10). But an easier proof will be done in Corollary 3.5.7*(iii)* and *(iv)* using 3.5.6*(ii)*.

The following Theorem provides a first evidence of the matching between operational and logic semantics.

3.3.19. THEOREM (Type invariance).

(i) $\Gamma \vdash M : \sigma \ \&\ M \bowtie N \Rightarrow \Gamma \vdash N : \sigma$;

(ii) $\Gamma \vdash M : \sigma \ \&\ M \longrightarrow^* N \Rightarrow \Gamma \vdash N : \sigma$.

PROOF. *(i)* is an easy consequence of Lemmas 3.3.15, 3.3.16, 3.3.17 and 3.3.18.
(ii) If $M \longrightarrow^* N$ then for some $b \in bar(M)$ it is the case that $N \in b$. By 3.2.9 *(ii)* $M \rhd \sum\{M' \mid M' \in b\}$. Being $\rhd \subseteq \bowtie$, by part *(i)* of the present theorem and 3.3.10*(vii)* we have $\Gamma \vdash M' : \sigma$ for all $M' \in b$, from which the thesis follows. □

The subject expansion property fails for $\xrightarrow{*}$. For example $\vdash \mathbf{I} : \omega \to \omega$ but, as we shall be able to derive from Corollary 3.5.6*(ii)*, $\not\vdash \mathbf{I} + \Omega : \omega \to \omega$.

The main result of the present section is that convergence implies typability by $\omega \to \omega$. We will see in section 3.5 that also the converse is true. Therefore this type will completely characterize terms whose meaning is to be eventually a function, even if not a unique one.

3.3.20. THEOREM. *Let M be a closed term.*

$$M\Downarrow \ \Rightarrow\ \vdash M : \omega \to \omega.$$

PROOF.

$$\begin{array}{lll} M\Downarrow & \Rightarrow \ \exists V_1, \ldots, V_n \in \mathsf{Val}.\ M \rhd \sum_{i=1}^n V_i & \text{by } 3.2.10(i) \\ & \Rightarrow \ \vdash \sum_{i=1}^n V_i : \omega \to \omega & \text{by } 3.3.12(i) \text{ and rule } (+\mathrm{I}) \\ & \Rightarrow \ \vdash M : \omega \to \omega & \text{by } 3.3.19(i). \end{array}$$ □

3.4. Models and Completeness

If we want to devise a domain equation for our concurrent λ-calculus, it is natural to start from the equations in the literature for similar languages.

Abramsky in [4] interprets the lazy λ-calculus by means of a Scott domain D solving the equation

$$D = [D \to D]_\perp$$

where $[D \to D]$ is the space of continuous functions and $(\cdot)_\perp$ is the lifting operator.

Boudol in [22] gives the semantics of the lazy, call-by-name and call-by-value λ-calculus enriched with a parallel operator using the same equation, but in a different category. It is easy to see from the asynchronous reduction rules of Boudol's parallel operator (shown at page 48) that in a "may" perspective $\|$ can be interpreted using the lower powerdomain. Boudol recalls that each prime algebraic lattice D is isomorphic to the lower powerdomain of the posets of the compact coprime elements of D. Therefore it suffices to find a solution of Abramsky's equation in this category to have a domain suitable for Boudol's language. Notice that Boudol interprets $M\|N$ as the join of the interpretations of M and N.

The reduction rules of the present parallel operator differ from those given in [22]. Really, our $\|$ is synchronous. But we are in a different perspective: we consider "must" convergence instead of "may" convergence. Therefore our parallel operator behaves exactly like Boudol's one from the viewpoint of convergence. In fact both operators converge whenever one of the two arguments does. This is clear when we think to the correspondence between asynchronicity in a "may" perspective and synchronicity in a "must" perspective. So we could have used Abramsky's domain equation again, if we would not have to interpret also the non-deterministic choice.

The reduction rules of $+$ in a "must" perspective clearly suggests the upper powerdomain for its interpretation. The whole discussion leads us to the following domain equation

$$D = \mathcal{P}^\sharp([D \to D]_\perp),$$

where $\mathcal{P}^\sharp$ is the upper powerdomain functor, in the category of prime algebraic lattices.

It is well known that each prime algebraic lattice can be described by an information system ([61]) and also by means of intersection types ([25]). Really we have developed in previous sections a system of intersection and union types; we will use this system now to build a model, which actually is the initial solution of our domain equation.

Because of rules $(\omega), (\leq)$ and $(\wedge\,\mathrm{I})$, the set of types assigned in $\mathcal{L}$ to any term is a filter over $Type$: let $\mathcal{F}$ be the set of all filters. We have that $\mathcal{F}$ is a distributive lattice under subset ordering (distributivity comes as a consequence of the distributivity of $Type$ itself), with intersection as meet and

$$F \bar{\cup} F' = \uparrow\{\sigma \wedge \tau \mid \sigma \in F, \tau \in F'\}$$

as join ($\uparrow$ stands as usual for upper closure). The bottom and the top of this lattice are respectively $\uparrow\omega$ and $Type$, where in general $\uparrow\sigma$ is the principal filter generated by σ. The compact elements are the principal filters. Moreover this lattice is prime algebraic, since each filter is the join of the compact coprime filters it dominates. Notice that a filter $F \in \mathcal{F}$ is compact *coprime* if and only if it is a principal filter generated by a *meet irreducible* type [1]. We refer to [7] for

[1] A type σ is *meet irreducible* or *prime* if and only if

$$\tau \wedge \rho \leq \sigma \Rightarrow \tau \leq \sigma \text{ or } \rho \leq \sigma$$

for any τ, ρ.

the whole proof that $\mathcal{F}$ is the initial solution of our domain equation.

Among filters assigned as meanings of terms, Theorem 3.3.12*(ii)* indicates that prime filters are the interpretations of terms that are total values. We recall that a filter $F \in \mathcal{F}$ is *prime* if and only if for all σ and τ

$$\sigma \vee \tau \in F \Rightarrow \sigma \in F \text{ or } \tau \in F.$$

We write $\mathcal{F}_P$ to denote the set of prime filters.

In any distributive lattice D the set $\mathrm{Pr}(D)$ of *prime elements* is defined as follows

$$d \in \mathrm{Pr}(D) \;\Leftrightarrow\; \forall x, y \in D.\, x \sqcap y \sqsubseteq d \Rightarrow x \sqsubseteq d \text{ or } y \sqsubseteq d.$$

We write $\mathrm{Pr}(x) = \uparrow x \cap \mathrm{Pr}(D)$ for $x \in D$. Let us define, for any filter F the set

$$\mathrm{Pr}(F) = \{P \in \mathcal{F}_P \mid F \subseteq P\}$$

which is called the *prime decomposition* of F. It is straightforward to see that $\mathrm{Pr}(\mathcal{F}) = \mathcal{F}_P$ and consequently that the previous definition of $\mathrm{Pr}(F)$ is consistent with the notation $\mathrm{Pr}(x)$.

From Priestley's Theorem we know that the structure of a distributive lattice is recoverable from its prime filters (or dually from its prime ideals). The following fact is at the basis of this result (see e.g. [29] Theorem 10.3)

(DPI) Let D be a distributive lattice, F a filter and I an ideal in D, such that $F \cap I = \emptyset$. Then there exists a prime filter P and a prime ideal J (actually J is the complement of P in D) such that $F \subseteq P$, $I \subseteq J$ and $P \cap J = \emptyset$.

The principle (DPI) implies that each filter is completely determined by its prime decomposition.

3.4.1. LEMMA. $\quad \forall F \in \mathcal{F}.\, F = \bigcap_{P \in \mathrm{Pr}(F)} P$.

PROOF. The left to right inclusion is immediate. To see the inverse inclusion let us suppose to a contradiction that there exists some $\sigma \in \bigcap_{P \in \mathrm{Pr}(F)} P$ such that $\sigma \notin F$. This implies that $\downarrow\sigma \cap F = \emptyset$, where $\downarrow\sigma$ is the principal ideal generated by σ; it follows by (DPI) that for some $P \in \mathcal{F}_P$ we have $F \subseteq P$ and $\downarrow\sigma \cap P = \emptyset$. $\square$

The last Lemma is an instance of a more general fact: let D be a lattice, then $X \subseteq D$ is *order generating* if and only if for all $x \in D$, $x = \sqcap(\uparrow x \cap X)$ (see [39], Ch.1, Definition 3.8). If D is continuous (i.e. complete and each element is the sup of its way below elements) then it is distributive if and only if $\mathrm{Pr}(D)$ is order generating (see [39], Ch.1, Theorem 3.14). But $\mathcal{F}$ is a distributive lattice which is prime algebraic, so it is a fortiori continuous. Therefore $\mathcal{F}_P$ is order generating.

To interprete functional application we turn $\mathcal{F}$ into an applicative structure as follows

$$F \cdot F' = \{\tau \mid \exists \sigma \in F'.\, \sigma \to \tau \in F\} \;\cup \{\uparrow\omega\}.$$

Observe that the definition of application is slightly different from that one given in [18]. Indeed we have to add explicitly the principal filter of ω since in our setting $\omega \neq \omega \to \omega$; otherwise $\uparrow\omega \cdot \uparrow\omega$ would be the empty set, which is not a filter.

3.4.2. Lemma. *The operation of application over $\mathcal{F}$ is monotonic in both its arguments; moreover*

$$(F \cap F') \cdot G \supseteq (F \cdot G) \cap (F' \cdot G) \quad \textit{and} \quad (F \bar{\cup} F') \cdot G \subseteq (F \cdot G) \bar{\cup} (F' \cdot G)$$

for all $F, F', G \in \mathcal{F}$.

The proof is straightforward. Just note that these inclusions are actually equalities, since the opposite inclusions follow from the monotonicity of the application.

The properties of $\mathcal{F}$ which have been seen so far suggest the following definition

3.4.3. Definition. A *pre-model* of $\Lambda_{+\|}$ is a structure $\mathcal{D} = \langle D, \sqsubseteq, \cdot, \sqcap, \sqcup \rangle$ where $\langle D, \sqsubseteq \rangle$ is a distributive continuous lattice and $\cdot$ is a monotonic binary operation on D such that, for all $d, d', e \in D$

(a) $(d \sqcap d') \cdot e \sqsupseteq (d \cdot e) \sqcap (d' \cdot e)$;

(b) $(d \sqcup d') \cdot e \sqsubseteq (d \cdot e) \sqcup (d' \cdot e)$.

Total values are associated by system $\mathcal{L}$ to prime filters different from $\uparrow \omega$. A call-by-value variable is a total value, hence a correct notion of environment for $\mathcal{F}$ is a mapping $\eta : \mathsf{Vn} \cup \mathsf{Vv} \to \mathcal{F}$ such that $\eta(\mathsf{Vv}) \subseteq \mathcal{F}_P - \{\uparrow \omega\}$. In general, given a pre-model $\mathcal{D}$, if $\mathcal{P} = \mathrm{Pr}(D) - \{\bot\}$, we define Env_D as the set of mappings $\eta : \mathsf{Vn} \cup \mathsf{Vv} \to D$ such that $\eta(\mathsf{Vv}) \subseteq \mathcal{P}$.

Now, for any environment $\eta \in Env_{\mathcal{F}}$ and for any basis Γ, we define

$$\Gamma \models \eta \;\Leftrightarrow\; \forall \chi \in \mathsf{Vn} \cup \mathsf{Vv}.\, \Gamma(\chi) \in \eta(\chi).$$

We are now in place of defining the map $[\![\cdot]\!]^{\mathcal{F}} : \Lambda_{+\|} \to Env_{\mathcal{F}} \to \mathcal{F}$ as follows

$$[\![M]\!]^{\mathcal{F}}_{\eta} = \{\sigma \mid \exists \Gamma.\, \Gamma \models \eta \;\&\; \Gamma \vdash M : \sigma\}.$$

This definition is consistent with the logical inclusion, which is equivalent to subset inclusion of interpretations.

3.4.4. Proposition. *For all $M, N \in \Lambda_{+\|}$*

$$M \sqsubseteq^{\mathcal{L}} N \;\Leftrightarrow\; \forall \eta.\, [\![M]\!]^{\mathcal{F}}_{\eta} \subseteq [\![N]\!]^{\mathcal{F}}_{\eta}\ .$$

Proof. ($\Rightarrow$) Immediate. ($\Leftarrow$) Let us define, for any basis Γ, $\eta_\Gamma(\chi) = \uparrow \Gamma(\chi)$ for all variable χ; then $\eta_\Gamma \in Env_{\mathcal{F}}$ since $\Gamma(v)$ is coprime for all call-by-value variable v, hence $\uparrow \Gamma(v)$ is a prime filter. Now $\Gamma \models \eta_\Gamma$ so that $\Gamma \vdash M : \sigma$ implies $\sigma \in [\![M]\!]^{\mathcal{F}}_{\eta_\Gamma}$. By hypothesis $\sigma \in [\![N]\!]^{\mathcal{F}}_{\eta_\Gamma}$, hence $\Gamma' \vdash N : \sigma$ for some Γ' such that $\Gamma' \models \eta_\Gamma$. We conclude that $\Gamma \vdash N : \sigma$ since $\Gamma' \models \eta_\Gamma$ implies $\Gamma \leq \Gamma'$. □

3.4.5. Corollary. *For all $M, N \in \Lambda_{+\|}$ and $\eta \in Env_{\mathcal{F}}$*

$$[\![M + N]\!]^{\mathcal{F}}_{\eta} = [\![M]\!]^{\mathcal{F}}_{\eta} \cap [\![N]\!]^{\mathcal{F}}_{\eta} \quad \textit{and} \quad [\![M \| N]\!]^{\mathcal{F}}_{\eta} = [\![M]\!]^{\mathcal{F}}_{\eta} \bar{\cup} [\![N]\!]^{\mathcal{F}}_{\eta}\ .$$

Proof. Immediate from 3.4.4 and from Lemmas 3.3.17*(i)*, *(ii)* and 3.3.18*(i)*, *(ii)*. □

Elaborating on the definition of λ-model, and also on the notion of λ-lattice proposed in [31], we fix the following.

3.4.6. Definition. The structure $\langle \mathcal{D}, [\![\cdot]\!]^{\mathcal{D}} \rangle$ is a *model* if $\mathcal{D} = \langle D, \sqsubseteq, \cdot, \sqcap, \sqcup \rangle$ is a pre-model and $[\![\cdot]\!]^{\mathcal{D}} : \Lambda_{+\|} \to Env_D \to D$ satisfies the following conditions

(i) $[\![\chi]\!]^{\mathcal{D}}_{\eta} = \eta(\chi)$;

(ii) $[\![MN]\!]^{\mathcal{D}}_{\eta} = [\![M]\!]^{\mathcal{D}}_{\eta} \cdot [\![N]\!]^{\mathcal{D}}_{\eta}$;

(iii) $[\![\lambda x.M]\!]^{\mathcal{D}}_{\eta} \cdot d = [\![M]\!]^{\mathcal{D}}_{\eta[x \mapsto d]}$;

(iv) $[\![\lambda v.M]\!]^{\mathcal{D}}_{\eta} \cdot d = \begin{cases} \bot & \text{if } d = \bot \\ \sqcap_{e \in \mathrm{Pr}(d)} [\![M]\!]^{\mathcal{D}}_{\eta[v \mapsto e]} & \text{otherwise;} \end{cases}$

(v) $(\forall \chi \in \mathrm{FV}(M).\ \eta(\chi) = \eta'(\chi)) \Rightarrow [\![M]\!]^{\mathcal{D}}_{\eta} = [\![M]\!]^{\mathcal{D}}_{\eta'}$;

(vi) $[\![\lambda\chi.M]\!]^{\mathcal{D}}_{\eta} = [\![\lambda\chi'.M[\chi'/\chi]]\!]^{\mathcal{D}}_{\eta}$ if $\chi' \notin \mathrm{FV}(M)$ and either $\chi, \chi' \in \mathsf{Vn}$ or $\chi, \chi' \in \mathsf{Vv}$;

(vii) $(\forall d \in D.\ [\![M]\!]^{\mathcal{D}}_{\eta[x \mapsto d]} = [\![N]\!]^{\mathcal{D}}_{\eta[x \mapsto d]}) \Rightarrow [\![\lambda x.M]\!]^{\mathcal{D}}_{\eta} = [\![\lambda x.N]\!]^{\mathcal{D}}_{\eta}$;

(viii) $(\forall e \in \mathcal{P}.\ [\![M]\!]^{\mathcal{D}}_{\eta[v \mapsto e]} = [\![N]\!]^{\mathcal{D}}_{\eta[v \mapsto e]}) \Rightarrow [\![\lambda v.M]\!]^{\mathcal{D}}_{\eta} = [\![\lambda v.N]\!]^{\mathcal{D}}_{\eta}$;

(ix) $[\![M + N]\!]^{\mathcal{D}}_{\eta} = [\![M]\!]^{\mathcal{D}}_{\eta} \sqcap [\![N]\!]^{\mathcal{D}}_{\eta}$;

(x) $[\![M\|N]\!]^{\mathcal{D}}_{\eta} = [\![M]\!]^{\mathcal{D}}_{\eta} \sqcup [\![N]\!]^{\mathcal{D}}_{\eta}$,

(xi) $W \in \mathsf{TVal} \Rightarrow [\![W]\!]^{\mathcal{D}}_{\eta} \in \mathcal{P}$

where $\mathcal{P} = \mathrm{Pr}(D) - \{\bot\}$.

With respect to the classical definition of (syntactical) λ-models, the novelties are in clauses *(iv)* and *(viii)*-*(x)*. Clause *(xi)* reflects the intended meaning of total values, which essentially are not sums. Clause *(viii)* takes into account that by definition $\eta(\mathsf{Vv}) \subseteq \mathcal{P}$. The last two clauses are suggested by Corollary 3.4.5. Clause *(iv)* is more demanding: indeed from Corollary 3.3.11 and Proposition 3.4.4 we can argue that a call-by-value abstraction defines a co-additive function, but this does not suffice to show that it is completely co-additive (i.e. preserving arbitrary meets). This is however true in the filter model, and finely fits into the fact that prime elements are order generating in continuous distributive lattices. To show this, or equivalently that the pre-model $\mathcal{F}$ can be turned into a model using $[\![\cdot]\!]^{\mathcal{F}}$, we need a couple of Lemmas.

3.4.7. Lemma. *For all $M, N \in \Lambda_{+\|}$ and $\eta \in Env_{\mathcal{F}}$*

$$[\![MN]\!]^{\mathcal{F}}_{\eta} = [\![M]\!]^{\mathcal{F}}_{\eta} \cdot [\![N]\!]^{\mathcal{F}}_{\eta}.$$

Proof. To prove $[\![MN]\!]^{\mathcal{F}}_{\eta} \subseteq [\![M]\!]^{\mathcal{F}}_{\eta} \cdot [\![N]\!]^{\mathcal{F}}_{\eta}$, let $\sigma \in [\![MN]\!]^{\mathcal{F}}_{\eta}$ and $\sigma \neq \omega$; then for some basis Γ we have $\Gamma \models \eta$ and $\Gamma \vdash MN : \sigma$. By Theorem 3.3.10*(vi)*, there exists some τ such that

$$\Gamma \vdash M : \tau \to \sigma \ \&\ \Gamma \vdash N : \tau.$$

It follows that $\tau \to \sigma \in [\![M]\!]^{\mathcal{F}}_{\eta}$ and $\tau \in [\![N]\!]^{\mathcal{F}}_{\eta}$, so that the thesis follows.

To see that $[\![M]\!]^{\mathcal{F}}_{\eta} \cdot [\![N]\!]^{\mathcal{F}}_{\eta} \subseteq [\![MN]\!]^{\mathcal{F}}_{\eta}$ we reason as in [18], namely if for some $\tau \in [\![N]\!]^{\mathcal{F}}_{\eta}$ it is the case that $\tau \to \sigma \in [\![M]\!]^{\mathcal{F}}_{\eta}$, then there are two bases, Γ_0, Γ_1, such that $\Gamma_i \models \eta$ for $i = 0, 1$, and

$$\Gamma_0 \vdash N : \tau \ \&\ \Gamma_1 \vdash M : \tau \to \sigma.$$

Now taking Γ_2 such that $\Gamma_2(\chi) = \Gamma_0(\chi) \wedge \Gamma_1(\chi)$ for all χ, it is easy to see that

$$\Gamma_2 \models \eta \ \&\ \Gamma_2 \vdash N : \tau \ \&\ \Gamma_2 \vdash M : \tau \to \sigma,$$

from which we get $\Gamma_2 \vdash MN : \sigma$, that is $\sigma \in [\![MN]\!]^{\mathcal{F}}_{\eta}$. □

3.4.8. LEMMA. *Let $\Sigma = \{\sigma_i\}_{i\in\mathbb{N}}$ be a chain such that for all i, $\sigma_i \leq \sigma_{i+1}$, and let $F \in \mathcal{F}$, then*
$$\forall P \in \Pr(F).\ P \cap \Sigma \neq \emptyset \Rightarrow F \cap \Sigma \neq \emptyset.$$

PROOF. Let I be the downward closure of Σ; then

$$\begin{array}{rcl} \rho_0, \rho_1 \in I & \Rightarrow & \exists \sigma_i, \sigma_j \in \Sigma.\ \rho_0 \leq \sigma_i \ \&\ \rho_1 \leq \sigma_j \\ & \Rightarrow & \exists \sigma_i, \sigma_j \in \Sigma.\ \rho_0 \vee \rho_1 \leq \sigma_i \vee \sigma_j = \sigma_{\max\{i,j\}} \in \Sigma \\ & \Rightarrow & \rho_0 \vee \rho_1 \in I \end{array}$$

so that I is an ideal. If $F \cap \Sigma = \emptyset$, then $F \cap I = \emptyset$, being F an upward closed set. By (DPI) there exists $P \in \Pr(F)$ such that $P \cap I = \emptyset$ and consequently $P \cap \Sigma = \emptyset$, so that the thesis follows by contraposition. □

3.4.9. THEOREM. *The structure $\langle\langle \mathcal{F}, \subseteq, \cdot, \cap, \bar{\cup}\rangle, [\![\cdot]\!]^{\mathcal{F}}\rangle$ is a model.*

PROOF. Because of Proposition 3.4.4 and Lemmas 3.3.15, 3.3.17, 3.3.18 and 3.4.7 the only relevant remaining point is to show that $\mathcal{F}$ satisfies clause *(iv)* of Definition 3.4.6. Recall that the bottom in $\mathcal{F}$ is $\uparrow\omega$; then this amounts to show that

$$[\![\lambda v.M]\!]^{\mathcal{F}}_\eta \cdot F = \begin{cases} \uparrow\omega & \text{if } F \equiv\uparrow\omega \\ \bigcap_{P\in\Pr(F)} [\![M]\!]^{\mathcal{F}}_{\eta[v\mapsto P]} & \text{otherwise.} \end{cases}$$

Now if $F \equiv\uparrow\omega$ then

$$\begin{array}{rcll} \sigma \in [\![\lambda v.M]\!]^{\mathcal{F}}_\eta \cdot \uparrow\omega & \Leftrightarrow & \omega \to \sigma \in [\![\lambda v.M]\!]^{\mathcal{F}}_\eta & \\ & \Leftrightarrow & \exists \Gamma \models \eta.\ \Gamma \vdash \lambda v.M : \omega \to \sigma & \\ & \Rightarrow & \sigma = \omega & \text{by Lemma 3.3.10}(v). \end{array}$$

If $F \not\equiv\uparrow\omega$, let us suppose that $\sigma \neq \omega$ (otherwise the thesis is trivial), then

$$\begin{array}{rcl} \sigma \in [\![\lambda v.M]\!]^{\mathcal{F}}_\eta \cdot F & \Leftrightarrow & \exists \tau \in F.\ \tau \to \sigma \in [\![\lambda v.M]\!]^{\mathcal{F}}_\eta \\ & \Leftrightarrow & \exists \tau \in F,\ \exists \Gamma \models \eta.\ \Gamma \vdash \lambda v.M : \tau \to \sigma \\ & \Leftrightarrow & \exists \tau \in F,\ \exists \Gamma \models \eta.\ \forall \tau' \in \Theta(\tau).\ \Gamma, v : \tau' \vdash M : \sigma \\ & & \text{by 3.3.10}(iv)\text{, since } \sigma \neq \omega \Rightarrow \tau \neq \omega \text{ by 3.3.10}(v). \end{array}$$

Now $F = \bigcap_{P\in\Pr(F)} P$ implies that, for all $P \in \Pr(F)$, $\tau \in P$ and hence $\tau' \in P$ for some $\tau' \in \Theta(\tau)$ by definition of prime filter (notice that τ' depends on P). This implies $\Gamma, v : \tau' \models \eta[v \mapsto P]$ for some $\tau' \in \Theta(\tau)$, so that $\sigma \in [\![M]\!]^{\mathcal{F}}_{\eta[v\mapsto P]}$: it follows that

$$[\![\lambda v.M]\!]^{\mathcal{F}}_\eta \cdot F \subseteq \bigcap_{P\in\Pr(F)} [\![M]\!]^{\mathcal{F}}_{\eta[v\mapsto P]}.$$

To see the opposite inclusion let $G = [\![\lambda v.M]\!]^{\mathcal{F}}_\eta$; we first show that, if $P \in \mathcal{F}_P$ and $P \neq\uparrow\omega$, then $G \cdot P = [\![M]\!]_{\eta[v\mapsto P]}$. Indeed, let $w \in \mathsf{Vv} - FV(M)$ and $\eta \in Env_{\mathcal{F}}$ be such that $\eta(w) = P$; then by Lemmas 3.4.7 and 3.3.15*(iii)*, we have

$$G \cdot P = [\![\lambda v.M]\!]^{\mathcal{F}}_\eta \cdot [\![w]\!]^{\mathcal{F}}_\eta = [\![(\lambda v.M)w]\!]^{\mathcal{F}}_\eta = [\![M[w/v]]\!]^{\mathcal{F}}_\eta = [\![M]\!]^{\mathcal{F}}_{\eta[v\mapsto P]}.$$

From $F \neq\uparrow\omega$ it follows $\uparrow\omega \notin \Pr(F)$, so that the equality above implies

$$\bigcap_{P\in\Pr(F)} G \cdot P \ = \bigcap_{P\in\Pr(F)} [\![M]\!]^{\mathcal{F}}_{\eta[v\mapsto P]}.$$

Suppose that $\sigma \in \bigcap_{P \in \mathrm{Pr}(F)} G \cdot P$. Let us define

$$\Pi = \{\pi \in \mathit{Type} \mid \exists P \in \mathrm{Pr}(F).\ \pi \in P \ \&\ \pi \to \sigma \in G\}.$$

This set is non-empty by hypothesis and it is countable being a subset of the denumerable set Type. Let us suppose that an enumeration of Π has been fixed; then we put

$$\Sigma = \{\tau_n\}_{n \in \mathbb{N}} \quad \text{where} \quad \tau_n = \bigvee_{i=0}^{n} \pi_i.$$

By construction Σ is a chain such that $\forall P \in \mathrm{Pr}(F).\ P \cap \Sigma \neq \emptyset$, so that by Lemma 3.4.8, there exists an m such that $\tau_m \in F$. Since G is the interpretation of a call-by-value abstraction, a simple induction using 3.3.11 shows that $\tau_n \to \sigma \in G$ for all n: we conclude that $\sigma \in G \cdot F$. $\square$

So far terms have been interpreted as collections of their properties, namely of their types. We now provide an interpretation of types as subsets of the domains of models. The domains have an order whose meaning is "to be better behaved" and more defined. We do not want that more defined objects have less properties than their minors, hence the subsets interpreting types have to be upward closed; more precisely they will be filters of the domain itself.

3.4.10. Definition. Let $\mathcal{D} = \langle D, \sqsubseteq, \cdot, \sqcap, \sqcup \rangle$ be a pre-model, and define for $X, Y \subseteq D$

$$X \Rightarrow Y = \{d \in D \mid d \neq \bot \ \&\ \forall e \in X.\ d \cdot e \in Y\},$$
$$X \bar{\cup} Y = \uparrow \{d \sqcap d' \mid d \in X, d' \in Y\},$$

where we overload $\bar{\cup}$. Then a *type structure* over $\mathcal{D}$ is a sublattice $\mathcal{T}$ of the lattice of filters over $\mathcal{D}$, such that $D \in \mathcal{T}$ and $\mathcal{T}$ is closed under $\Rightarrow, \cap$, and $\bar{\cup}$.
The map $[\![\cdot]\!]^{\mathcal{T}} : \mathit{Type} \to \mathcal{T}$ is inductively defined as follows

(i) $[\![\omega]\!]^{\mathcal{T}} = D$;

(ii) $[\![\sigma \to \tau]\!]^{\mathcal{T}} = [\![\sigma]\!]^{\mathcal{T}} \Rightarrow [\![\tau]\!]^{\mathcal{T}}$;

(iii) $[\![\sigma \wedge \tau]\!]^{\mathcal{T}} = [\![\sigma]\!]^{\mathcal{T}} \cap [\![\tau]\!]^{\mathcal{T}}$;

(iv) $[\![\sigma \vee \tau]\!]^{\mathcal{T}} = [\![\sigma]\!]^{\mathcal{T}} \bar{\cup} [\![\tau]\!]^{\mathcal{T}}$.

Finally, given a model $\langle \mathcal{D}, [\![\cdot]\!]^{\mathcal{D}} \rangle$ we define

(v) $\Gamma \models_{\mathcal{D}} M : \sigma \;\Leftrightarrow\; (\forall \eta \in \mathit{Env}_D.\ \Gamma \models \eta \Rightarrow [\![M]\!]^{\mathcal{D}}_{\eta} \in [\![\sigma]\!]^{\mathcal{T}})$;

(vi) $\Gamma \models M : \sigma \;\Leftrightarrow\; \forall \mathcal{D}.\ \Gamma \models_{\mathcal{D}} M : \sigma$.

In the definition of $X \Rightarrow Y$ we put the condition $d \neq \bot$ since we are modeling a lazy calculus: this means that the bottom cannot be interpreted as a function. Consequently we exclude $\bot$ from $X \Rightarrow Y$, whose intended meaning is the set of representatives of functions which, when restricted to X, have ranges included in Y. Observe that $X \Rightarrow Y$ is a filter in D if both X and Y are.

We end this section by stating and proving a completeness theorem.

3.4.11. Theorem (Completeness).

$$\Gamma \vdash M : \sigma \;\Leftrightarrow\; \Gamma \models M : \sigma.$$

Proof. ($\Rightarrow$) By induction on derivations in $\mathcal{L}$.
($\Leftarrow$) First define $X_\sigma = \{F \in \mathcal{F} \mid \sigma \in F\}$ as usual, so that it is easily checked that $\mathcal{T} = \{X_\sigma\}_{\sigma \in Type}$ is a type structure over $\mathcal{F}$. More precisely $X_\omega = \mathcal{F}$, $X_{\sigma\to\tau} = X_\sigma \Rightarrow X_\tau$, $X_{\sigma\wedge\tau} = X_\sigma \cap X_\tau$ and $X_{\sigma\vee\tau} = X_\sigma \bar{\cup} X_\tau$. This implies, by a simple induction on types, that $[\![\sigma]\!]^{\mathcal{T}} = X_\sigma$. Suppose $\Gamma \models M : \sigma$, hence in particular $\Gamma \models_{\mathcal{F}} M : \sigma$, since by Theorem 3.4.9, $\mathcal{F}$ is a model. Put $\eta_\Gamma(\chi) = \uparrow\Gamma(\chi)$. Now $\Gamma \models \eta_\Gamma$ implies $[\![M]\!]^{\mathcal{F}}_{\eta_\Gamma} \in X_\sigma$, that is $\sigma \in [\![M]\!]^{\mathcal{F}}_{\eta_\Gamma}$. It follows that, for some Γ' such that $\Gamma' \models \eta_\Gamma$ we have $\Gamma' \vdash M : \sigma$. We conclude that $\Gamma \vdash M : \sigma$ since $\Gamma' \models \eta_\Gamma$ implies $\Gamma \leq \Gamma'$. □

3.5. Full Abstraction

In this section we will prove that the filter model exactly mirrors the operational semantics, i.e. that it is fully abstract. This means that

- the filter model is adequate, that is it does not equate operationally distinct programs

$$M \sqsubseteq^{\mathcal{L}} N \Rightarrow M \sqsubseteq^{\mathcal{O}} N\;;$$

- the filter model reflects the operational distinctions

$$M \sqsubseteq^{\mathcal{O}} N \Rightarrow M \sqsubseteq^{\mathcal{L}} N.$$

The key property on which the proof of full abstraction relies is that any compact element of $\mathcal{F}$, which is of the shape $\uparrow\sigma$, is λ-definable, since for all types σ there exists a characteristic (closed) term R_σ such that

$$\vdash \mathsf{R}_\sigma : \tau \quad\Leftrightarrow\quad \sigma \leq \tau$$

that is $[\![\mathsf{R}_\sigma]\!]^{\mathcal{F}} = \uparrow\sigma$.

Such terms are constructed inductively together with test terms. To each type σ we associate a test term T_σ such that for all closed terms M:

$$\mathsf{T}_\sigma M \Downarrow \;\Leftrightarrow\; \vdash M : \sigma.$$

The definition of characteristic and test terms finely reflects the duality between $\|$ and $+$, as well as their correspondence with $\wedge$ and $\vee$, respectively.

A further step in the full abstraction proof consists in giving a "realizability interpretation" of types as sets of closed terms. This is sound since each type is inhabited by some closed term (at least its characteristic term).

The main result we obtain is the converse of 3.3.20, namely

$$(*) \qquad \vdash M : \omega \to \omega \Rightarrow M\Downarrow$$

for all closed terms M.

The full abstraction theorem then follows. Indeed adequacy is a consequence of $(*)$ and of the fact that $\simeq^{\mathcal{L}}$ is a congruence. For the converse it suffices to observe that test terms discriminate internally, that is with respect to the convergence predicate, terms having different interpretations in the filter model.

3.5.1. Characteristic Terms

We define two families of terms $\{\mathsf{R}_\sigma\}_{\sigma\in Type}$ and $\{\mathsf{T}_\sigma\}_{\sigma\in Type}$ starting from Ω . We can replace safely Ω by any unsolvable of degree 0.

3.5.1. Definition. The *characteristic terms* R_σ and the *test terms* T_σ are defined by simultaneous induction on σ:

$$\begin{array}{ll} \mathsf{R}_\omega \equiv \Omega; & \mathsf{T}_\omega \equiv \lambda x\mathbf{I}; \\ \mathsf{R}_{\sigma\to\tau} \equiv \lambda x.\mathsf{T}_\sigma x\,\mathsf{R}_\tau; & \mathsf{T}_{\sigma\to\tau} \equiv \lambda v.\mathsf{T}_\tau(v\,\mathsf{R}_\sigma); \\ \mathsf{R}_{\sigma\wedge\tau} \equiv \mathsf{R}_\sigma \| \mathsf{R}_\tau; & \mathsf{T}_{\sigma\wedge\tau} \equiv \lambda x.(\mathsf{T}_\sigma\,x + \mathsf{T}_\tau\,x); \\ \mathsf{R}_{\sigma\vee\tau} \equiv \mathsf{R}_\sigma + \mathsf{R}_\tau. & \mathsf{T}_{\sigma\vee\tau} \equiv \lambda v.(\mathsf{T}_\sigma\,v \,\|\, \mathsf{T}_\tau\,v) \text{ where } \sigma\vee\tau \neq \omega. \end{array}$$

Notice the different use of call-by-value variables in the definition of $\mathsf{T}_{\sigma\to\tau}$ and $\mathsf{T}_{\sigma\vee\tau}$: $\mathsf{T}_{\sigma\to\tau}$ must check that its argument has type $\sigma\to\tau$ which, by $(*)$ above, implies that it has to be convergent. On the other hand the argument of $\mathsf{T}_{\sigma\vee\tau}$ may reduce to a sum $P+Q$ having type $\sigma\vee\tau$ because P has type σ and Q has type τ but neither σ nor τ can be deduced for $P+Q$. Therefore it is essential that it is evaluated before the application in parallel of T_σ and T_τ.

The types which can be deduced for R_σ and T_σ are meaningful for their operational behavior. In fact:

- R_σ has exactly the types greater than or equal to σ;
- T_σ has type $\tau\to\rho\to\rho$ only if $\tau\le\sigma$.

This means that R_σ is "the worst" term of type σ and that $\mathsf{T}_\sigma M$ reduces to a value if and only if we can deduce the type σ for M (and this value behaves like the identity combinator).

3.5.2. Lemma.

(i) $\vdash \mathsf{R}_\sigma : \tau \;\Leftrightarrow\; \sigma\le\tau$;

(ii) $\vdash \mathsf{T}_\sigma : \tau\to\gamma\to\delta \;\Leftrightarrow\; \tau\le\sigma \;\&\; \gamma\le\delta$.

Proof. We prove *(i)* and *(ii)* simultaneously by induction on σ. We consider only the interesting cases.

$\sigma\equiv\mu\to\nu$

(i) $\mathsf{R}_{\mu\to\nu} \equiv \lambda x.(\mathsf{T}_\mu x)\,\mathsf{R}_\nu$ and assume $\vdash \mathsf{R}_{\mu\to\nu} : \tau$. Then we proceed by a subordinate induction on the structure of τ, the base case being $\tau\equiv\alpha\to\beta$ since $\mathsf{R}_{\mu\to\nu}$ is an abstraction (see 3.3.10*(ii)*). Now by 3.3.10*(iii)* and *(vi)* we have $x:\alpha\vdash \mathsf{T}_\mu:\gamma\to\delta\to\beta$, $x:\alpha\vdash x:\gamma$, and $x:\alpha\vdash\mathsf{R}_\nu:\delta$, for some γ and δ. By induction and 3.3.10*(i)* this implies $\gamma\le\mu$, $\delta\le\beta$, $\alpha\le\gamma$ and $\nu\le\delta$, that is $\alpha\le\mu$ and $\nu\le\beta$. We conclude that $\mu\to\nu\le\alpha\to\beta$.
When $\tau\equiv\alpha\vee\beta$ the thesis follows from the subordinate induction hypothesis. In fact 3.3.12*(ii)* implies that $\vdash\mathsf{R}_{\mu\to\nu}:\alpha$ or $\vdash\mathsf{R}_{\mu\to\nu}:\beta$, being $\mathsf{R}_{\mu\to\nu}$ a total value.
The case $\tau\equiv\alpha\wedge\beta$ follows immediately from the subordinate induction hypothesis.

(ii) $\mathsf{T}_{\mu\to\nu}\equiv\lambda v.\mathsf{T}_\nu(v\mathsf{R}_\mu)$ and suppose that $\vdash\mathsf{T}_{\mu\to\nu}:\tau\to\gamma\to\delta$. Then we have

$$\begin{array}{ll} & \forall\tau'\in\Theta(\tau).\ v:\tau'\vdash\mathsf{T}_\nu(v\mathsf{R}_\mu):\gamma\to\delta \qquad \text{by 3.3.10}\textit{(iv)} \\ \Rightarrow & \forall\tau'\in\Theta(\tau)\ \exists\alpha.\ v:\tau'\vdash\mathsf{T}_\nu:\alpha\to\gamma\to\delta \;\&\; v:\tau'\vdash v\mathsf{R}_\mu:\alpha \\ & \qquad\qquad \text{by 3.3.10}\textit{(vi)} \\ \Rightarrow & \forall\tau'\in\Theta(\tau)\ \exists\alpha.\ \alpha\le\nu \;\&\; \gamma\le\delta \;\&\; \tau'\le\mu\to\alpha \\ & \qquad\qquad \text{by induction and 3.3.10}\textit{(i)}, \textit{(vi)} \\ \Rightarrow & \forall\tau'\in\Theta(\tau).\ \tau'\le\mu\to\nu \;\&\; \gamma\le\delta \\ \Rightarrow & \tau\le\mu\to\nu \;\&\; \gamma\le\delta. \end{array}$$

$\sigma \equiv \mu \vee \nu$

(ii) $\mathsf{T}_{\mu\vee\nu} \equiv \lambda v.(\mathsf{T}_\mu v \| \mathsf{T}_\nu v)$ and suppose that $\vdash \mathsf{T}_{\mu\vee\nu} : \tau \to \gamma \to \delta$. Then we have:

$$\begin{array}{lll} & \forall \tau' \in \Theta(\tau).\ v : \tau' \vdash (\mathsf{T}_\mu v \| \mathsf{T}_\nu v) : \gamma \to \delta & \text{by 3.3.10}\textit{(iv)} \\ \Rightarrow & \forall \tau' \in \Theta(\tau)\ \exists \rho_1, \rho_2.\ v : \tau' \vdash \mathsf{T}_\mu v : \rho_1\ \&\ v : \tau' \vdash \mathsf{T}_\nu : \rho_2\ \& & \\ & \rho_1 \wedge \rho_2 \leq \gamma \to \delta & \text{by 3.3.10}\textit{(viii)}. \end{array}$$

We assume $\rho_1 \neq \omega$ and $\rho_2 \neq \omega$. $\rho_1 \wedge \rho_2 \leq \gamma \to \delta$ implies by 3.3.7*(i)* that there are δ_1, δ_2 such that

$$\delta_1 \wedge \delta_2 = \delta\ \&\ \rho_1 \leq \gamma \to \delta_1\ \&\ \rho_2 \leq \gamma \to \delta_2.$$

$$\begin{array}{lll} & v : \tau' \vdash \mathsf{T}_\mu v : \rho_1 & \\ \Rightarrow & \exists \alpha.\ v : \tau' \vdash v : \alpha\ \&\ \vdash \mathsf{T}_\mu : \alpha \to \rho_1 & \text{by 3.3.10}\textit{(vi)} \\ \Rightarrow & \vdash \mathsf{T}_\mu : \tau' \to \rho_1 & \text{by 3.3.10}\textit{(i)} \text{ and rule } (\leq) \\ \Rightarrow & \vdash \mathsf{T}_\mu : \tau' \to \gamma \to \delta_1 & \text{by above and rule } (\leq) \\ \Rightarrow & \tau' \leq \mu\ \&\ \gamma \leq \delta_1 & \text{by induction.} \end{array}$$

Analogously from $v : \tau' \vdash \mathsf{T}_\nu v : \rho_2$ we deduce $\tau' \leq \nu$ and $\gamma \leq \delta_2$. So we can conclude $\tau \leq \mu \vee \nu$ and $\gamma \leq \delta_1 \wedge \delta_2$.
The case in which one of ρ_i $(i = 1, 2)$ is equal to ω is similar and simpler. In fact if $\rho_2 = \omega$ we have $\rho_1 \leq \gamma \to \delta$. This allows us to prove $\tau' \leq \mu$ and $\gamma \leq \delta$ from $v : \tau' \vdash \mathsf{T}_\mu v : \rho_1$. So we can conclude once more $\tau \leq \mu \vee \nu$ and $\gamma \leq \delta$.
□

3.5.2. Realizability

Aim of this subsection is to prove that

$$\vdash M : \omega \to \omega\ \Rightarrow\ M \Downarrow$$

for all closed terms M. As an immediate consequence we have that only types equivalent to ω can be derived for divergent terms.

The proof of this fact requires a double induction, on types and deductions. Following a standard methodology, we split this induction by introducing a "realizability interpretation" of types as sets of closed terms.

3.5.3. Definition. We define the mapping $[\![\cdot]\!] : \mathit{Type} \to \mathcal{P}(\Lambda^0_{+\|})$ by induction:

(i) *(a)* $[\![\omega]\!] = \Lambda^0_{+\|}$;

(b) $[\![\sigma \to \tau]\!] = \{M \in \Lambda^0_{+\|} \mid M\Downarrow\ \&\ \forall N \in [\![\sigma]\!] \Rightarrow MN \in [\![\tau]\!]\}$;

(c) $[\![\sigma \wedge \tau]\!] = \{M \in \Lambda^0_{+\|} \mid M \in [\![\sigma]\!] \text{ and } M \in [\![\tau]\!]\}$;

(d) $[\![\sigma \vee \tau]\!] = \{M \in \Lambda^0_{+\|} \mid M \in [\![\sigma]\!] \text{ or } M \in [\![\tau]\!] \text{ or } \exists N \in [\![\sigma]\!], L \in [\![\tau]\!].\ M \bowtie N + L\}$.

(ii) If M is open, let $FV(M) = \{x_1, \ldots, x_m, v_1, \ldots, v_n\}$, and $\Gamma(x_i) = \mu_i$ $(1 \leq i \leq m)$, $\Gamma(v_j) = \nu_j$ $(1 \leq j \leq n)$, then

$$\Gamma \models^r M : \sigma \Leftrightarrow M[N_1/x_1, \ldots, N_m/x_m, L_1/v_1, \ldots, L_n/v_n] \in [\![\sigma]\!]$$

for all $N_i \in [\![\mu_i]\!]$ and $L_j \in [\![\nu_j]\!]$ $(1 \leq i \leq m$ and $1 \leq j \leq n)$.

The correctness of this definition is due to the fact that all types are inhabited by some closed term; in fact 3.5.5 will imply that $\models^r \mathsf{R}_\sigma : \sigma$ for all types σ.

The following Lemma states some key properties of our realizability interpretation.

3.5.4. LEMMA. *Let* $M, N, W \in \Lambda^0_{+\|}$*, then:*

(i) $M \in [\![\sigma]\!]$ & $M \bowtie N \Rightarrow N \in [\![\sigma]\!]$;

(ii) $M \in [\![\sigma]\!] \Rightarrow M\|N \in [\![\sigma]\!]$;

(iii) $M \in [\![\sigma]\!]$ & $N \in [\![\sigma]\!] \Leftrightarrow M + N \in [\![\sigma]\!]$;

(iv) $W \in [\![\sigma \vee \tau]\!]$ & $W \in \mathsf{TVal} \Rightarrow W \in [\![\sigma]\!]$ *or* $W \in [\![\tau]\!]$;

(v) $M \in [\![\sigma]\!]$ & $\sigma \leq \tau \Rightarrow M \in [\![\tau]\!]$;

(vi) $M \in [\![\sigma]\!]$ & $\sigma \neq \omega \Rightarrow M\Downarrow$.

PROOF. We prove points *(i)-(iii)* of this lemma by induction on σ. The case $\sigma \equiv \omega$ is always trivial. The cases $\sigma \equiv \tau \wedge \rho$ and $\sigma \equiv \tau \vee \rho$ with $M \in [\![\tau]\!] \cup [\![\rho]\!]$ ($N \in [\![\tau]\!] \cup [\![\rho]\!]$) immediately follow from the induction hypothesis. Therefore the proofs of these cases are omitted. *(i)*

Case $\sigma \equiv \tau \to \rho$. We have that $M\Downarrow$ implies $N\Downarrow$ by 3.2.12*(ii)*; moreover

$$\begin{array}{lll} & M \in [\![\sigma]\!] & \\ \Rightarrow & \forall L \in [\![\tau]\!].\ ML \in [\![\rho]\!] & \text{by definition} \\ \Rightarrow & \forall L \in [\![\tau]\!].\ NL \in [\![\rho]\!] & \text{by induction since } ML \bowtie NL \text{ by 3.2.8}\mathit{(i)} \\ \Rightarrow & N \in [\![\sigma]\!] & \text{by definition.} \end{array}$$

Case $\sigma \equiv \tau \vee \rho$ and $M \bowtie P + Q$, for some $P \in [\![\tau]\!]$ and $Q \in [\![\rho]\!]$. Therefore $N \bowtie P + Q$ and we are done.

(ii)

Case $\sigma \equiv \tau \to \rho$. We have that $M\Downarrow$ implies $M\|N\Downarrow$; moreover

$$\begin{array}{lll} & M \in [\![\sigma]\!] & \\ \Rightarrow & \forall L \in [\![\tau]\!].\ ML \in [\![\rho]\!] & \text{by definition} \\ \Rightarrow & \forall L \in [\![\tau]\!], N \in \Lambda^0_{+\|}.\ ML\|NL \in [\![\rho]\!] & \text{by induction} \\ \Rightarrow & \forall L \in [\![\tau]\!], N \in \Lambda^0_{+\|}.\ (M\|N)L \in [\![\rho]\!] & \text{by } \mathit{(i)} \text{ since } (M\|N)L \rhd ML\|NL \\ & & \text{by rule } (\|_{app})' \\ \Rightarrow & \forall N \in \Lambda^0_{+\|}. M\|N \in [\![\sigma]\!] & \text{by definition.} \end{array}$$

Case $\sigma \equiv \tau \vee \rho$ and $M \bowtie P + Q$, for some $P \in [\![\tau]\!]$ and $Q \in [\![\rho]\!]$. Now $M \bowtie P + Q$ implies $M\|N \bowtie (P + Q)\|N$ by 3.2.8*(i)* and $(P + Q)\|N \rhd P\|N + Q\|N$ by rule $(+\|)'$ for all $N \in \Lambda^0_{+\|}$. By induction $P\|N \in [\![\tau]\!]$ and $Q\|N \in [\![\rho]\!]$, so we conclude that $M\|N \in [\![\sigma]\!]$ for all $N \in \Lambda^0_{+\|}$.

(iii) ($\Rightarrow$).

Case $\sigma \equiv \tau \to \rho$.

$$\begin{array}{lll} & M, N \in [\![\sigma]\!] & \\ \Rightarrow & \forall L \in [\![\tau]\!].\ ML, NL \in [\![\rho]\!] & \text{by definition} \\ \Rightarrow & \forall L \in [\![\tau]\!].\ ML + NL \in [\![\rho]\!] & \text{by induction} \\ \Rightarrow & \forall L \in [\![\tau]\!].\ (M + N)L \in [\![\rho]\!] & \text{by } \mathit{(i)} \text{ since } (M + N)L \rhd ML + NL \\ & & \text{by rule } (+_{app})' \\ \Rightarrow & M + N \in [\![\sigma]\!] & \text{by definition.} \end{array}$$

Case $\sigma \equiv \tau \vee \rho$ and $M \bowtie M_0 + M_1$, $N \bowtie N_0 + N_1$, for some $M_0, N_0 \in [\![\tau]\!]$ and $M_1, N_1 \in [\![\rho]\!]$.
From the induction hypothesis $M_0 + N_0 \in [\![\tau]\!]$ and $M_1 + N_1 \in [\![\rho]\!]$; therefore

$$M + N \bowtie (M_0 + N_0) + (M_1 + N_1) \in [\![\tau \vee \rho]\!].$$

($\Leftarrow$).

Case $\sigma \equiv \tau \to \rho$.

$$\begin{array}{lll} & M + N \in [\![\sigma]\!] & \\ \Rightarrow & \forall L \in [\![\tau]\!].\, (M+N)L \in [\![\rho]\!] & \text{by definition} \\ \Rightarrow & \forall L \in [\![\tau]\!].\, ML + NL \in [\![\rho]\!] & \text{by } (i) \text{ since } (M+N)L \rhd ML + NL \\ & & \text{by rule } (+_{app})' \\ \Rightarrow & \forall L \in [\![\tau]\!].\, ML, NL \in [\![\rho]\!] & \text{by induction} \\ \Rightarrow & M, N \in [\![\sigma]\!] & \text{by definition.} \end{array}$$

Case $\sigma \equiv \tau \vee \rho$ and $M + N \bowtie P + Q$, for some $P \in [\![\tau]\!]$ and $Q \in [\![\rho]\!]$. If $M \bowtie P$ and $N \bowtie Q$ (or vice-versa) the thesis follows from *(i)*. Otherwise, by Lemma 3.2.8*(ii)* there exist M_0, M_1, N_0, N_1 such that $M \bowtie M_0 + M_1$, $N \bowtie N_0 + N_1$ and $P \bowtie M_0 + N_0$, $Q \bowtie M_1 + N_1$. By induction $M_0, N_0 \in [\![\tau]\!]$ and $M_1, N_1 \in [\![\rho]\!]$. So we conclude that $M \in [\![\tau \vee \rho]\!]$ and $N \in [\![\tau \vee \rho]\!]$ by definition.

(iv) It suffices to observe that $W \in \mathsf{TVal}$ implies that $W \bowtie M + N$ is impossible for any M, N.
(v) By induction on the definition of $\leq$. In the case $\sigma \vee \sigma \leq \sigma$ use *(iii)*($\Rightarrow$).
We consider the case $(\sigma \vee \tau) \wedge \rho \leq (\sigma \wedge \rho) \vee (\tau \wedge \rho)$ when $M \bowtie P + Q$, $P \in [\![\sigma]\!]$, $Q \in [\![\tau]\!]$ and $M \in [\![\rho]\!]$. Now $M \in [\![\rho]\!]$ implies $P \in [\![\rho]\!]$ and $Q \in [\![\rho]\!]$ by *(i)* and *(iii)*($\Leftarrow$). Therefore we have $P \in [\![\sigma \wedge \rho]\!]$ and $Q \in [\![\tau \wedge \rho]\!]$, so we can conclude $M \in [\![(\sigma \wedge \rho) \vee (\tau \wedge \rho)]\!]$.
(vi) By induction on σ, taking into account that $\sigma \equiv \tau \vee \rho$ and $\sigma \neq \omega$ imply $\tau \neq \omega$ and $\rho \neq \omega$.
□

As expected, realizability coincides with derivability in $\vdash$ and this implies in turn that we can assure convergence for all closed terms typable by $\omega \to \omega$.

3.5.5. Theorem (Realizability Theorem).

$$\forall \Gamma, \sigma, M.\, [\Gamma \models^r M : \sigma \;\Leftrightarrow\; \Gamma \vdash M : \sigma].$$

Proof. Let $FV(M) = \{x_1, \ldots, x_m, v_1, \ldots, v_n\}$ and $\Gamma = \{x_1{:}\mu_1, \ldots, x_m{:}\mu_m, v_1{:}\nu_1, \ldots, v_n{:}\nu_n\}$. Let us choose $N_1, \ldots, N_m$ and $L_1, \ldots, L_n$ as in Definition 3.5.3*(ii)* and put

$$U^* \equiv U[N_1/x_1, \ldots, N_m/x_m, L_1/v_1, \ldots, L_n/v_n].$$

It suffices to show that $M^* \in [\![\sigma]\!]$ iff $\Gamma \vdash M : \sigma$.
($\Leftarrow$) The proof is by induction on deductions. We consider only the interesting cases.

- Case ($\to$ I$_v$). Then $M \equiv \lambda v.P$ for some P, $\sigma \equiv \tau \to \rho$ and $\Gamma, v : \tau' \vdash P : \rho$ has been derived for all $\tau' \in \Theta(\tau)$. Let $L \in [\![\tau']\!]$ for some $\tau' \in \Theta(\tau)$. Now $L \in [\![\tau']\!]$ implies $L \Downarrow$ by 3.5.4*(vi)* since $\tau' \neq \omega$. Therefore, by Corollary 3.2.10, there are $V_1, \ldots, V_k \in \mathsf{Val}$ such that $L \rhd \sum_{i=1}^k V_i$ and $(\lambda v.P)L \rhd \sum_{i=1}^k (\lambda v.P)V_i$. Notice that by 3.5.4*(i)* and *(iii)* $\forall i \leq k.\, V_i \in [\![\tau']\!]$.
  From the induction hypothesis for all $i \leq n$ $P[V_i/v]^* \in [\![\rho]\!]$; there is no loss of generality in supposing that v does not occur in $N_1, \ldots, N_m, L_1, \ldots, L_n$ so that $(P[V_i/v])^* \equiv P^*[V_i/v]$. We have $(\lambda v.P)^* V_i \equiv (\lambda v.P^*)V_i$. Now $(\lambda v.P^*)V_i \rhd P^*[V_i/v]$ if $V_i \in \mathsf{TVal}$ and $(\lambda v.P^*)V_i \rhd$

$\sum_{j=1}^{h}(P^*[V_i/v]\|(\lambda v.P^*)V_j')$, where $\mathcal{R}(V_i,1) = \{V_1',\dots,V_h'\}$, otherwise. In both cases it follows $(\lambda v.P)^*V_i \in [\![\rho]\!]$ by 3.5.4*(i)*, *(ii)*, *(iii)*.

$$\begin{array}{rcll} \forall i \leq k.\ M^*V_i \in [\![\rho]\!] & \Rightarrow & \sum_{i=1}^{k} M^*V_i \in [\![\rho]\!] & \text{by } 3.5.4\textit{(iii)} \\ & \Rightarrow & M^*L \in [\![\rho]\!] & \text{by } 3.5.4\textit{(i)} \\ & & & \text{since } M^*L \rhd \sum_{i=1}^{k} M^*V_i \\ & & & \text{by construction.} \end{array}$$

So we conclude $M^* \in [\![\sigma]\!]$ by the arbitrariness of the computable term L.

- Case ($\|$ I). Then $M \equiv P\|Q$ for some P, Q, $\sigma \equiv \tau \wedge \rho$ and , say, $\Gamma \vdash P : \tau$ and $\Gamma \vdash Q : \rho$ have been derived. From the induction hypothesis $P^* \in [\![\tau]\!]$ and $Q^* \in [\![\rho]\!]$, so that by Lemma 3.5.4*(ii)*, $P^*\|Q^* \in [\![\tau]\!]$ and $P^*\|Q^* \in [\![\rho]\!]$, which imply by definition $(P\|Q)^* \in [\![\sigma]\!]$.

($\Rightarrow$) By induction on σ. The only interesting case is when $\sigma \equiv \tau \to \rho$.

$$\begin{array}{rcll} M^* \in [\![\sigma]\!] & \Rightarrow & \forall N \in [\![\tau]\!].\ M^*N \in [\![\rho]\!] & \text{by definition} \\ & \Rightarrow & M^*\mathsf{R}_\tau \equiv (M\mathsf{R}_\tau)^* \in [\![\rho]\!] & \text{since } \mathsf{R}_\tau \in [\![\tau]\!] \text{ by } (\Leftarrow) \\ & \Rightarrow & \Gamma \vdash M\mathsf{R}_\tau : \rho & \text{by induction} \\ & \Rightarrow & \Gamma \vdash M : \tau \to \rho & \text{by } 3.3.10\textit{(vi)} \text{ and } 3.5.2\textit{(i)}. \end{array}$$

□

The main result of this subsection is the characterization of convergent terms by the type $\omega \to \omega$ (see also Theorem 3.3.20).

3.5.6. Corollary.

(i) $\forall M \in \Lambda^0_{+\|}.\ \vdash M : \omega \to \omega \Rightarrow M\Downarrow$.

(ii) $\forall M \in \Lambda^0_{+\|}.\ M\Uparrow\ \&\ \vdash M : \sigma \Rightarrow \sigma = \omega$.

As already stated, the characterization of types which can be deduced for divergent terms given in Corollary 3.5.6*(ii)* allows us to prove that some inclusions in the model are proper.

3.5.7. Corollary.

(i) $\exists M \in \Lambda_{+\|}, V \in \mathsf{Val}$, σ *such that* $[\forall \sigma' \in \Theta(\sigma).\ v : \sigma' \vdash M : \tau]\ \&\ \vdash V : \sigma$ *but* $\not\vdash M[V/v] : \tau$;

(ii) $\exists M \in \Lambda_{+\|}, V \in \mathsf{Val}$, τ *such that* $\vdash (\lambda v.M)V : \tau$ *but* $\not\vdash M[V/v] : \tau$;

(iii) $\exists L, M, N \in \Lambda_{+\|}$ *such that* $L(M+N) \sqsubset^{\mathcal{L}} LM + LN$;

(iv) $\exists L, M, N \in \Lambda_{+\|}$ *such that* $LM\|LN \sqsubset^{\mathcal{L}} L(M\|N)$.

Proof.

(i) An example is $M \equiv v\mathbf{I}\Delta\Delta\|v\Delta\mathbf{I}\Delta$, $V \equiv \Delta\|(\mathbf{K}+\mathbf{O})$ and $\sigma = \sigma_1 \vee \sigma_2$, where $\sigma_1 \equiv \rho \to \omega \to \rho$, $\sigma_2 \equiv \omega \to \rho \to \rho$, $\rho \equiv \tau \to \tau$ and $\tau \equiv (\omega^2 \to \omega) \to \omega \to \omega$. We can easily check that $\vdash \mathbf{I} : \rho$ (1), $\vdash \Delta : \tau$ (2), $\vdash \mathbf{K} : \sigma_1$ (3), and $\vdash \mathbf{O} : \sigma_2$ (4). From (1) and (2) we obtain $v : \sigma_1 \vdash v\mathbf{I}\Delta\Delta : \tau$ (5) and $v : \sigma_2 \vdash v\Delta\mathbf{I}\Delta : \tau$ (6), which imply respectively $v : \sigma_1 \vdash M : \tau$ and $v : \sigma_2 \vdash M : \tau$. Using (3) and (4) we derive $\vdash \mathbf{K} + \mathbf{O} : \sigma_1 \vee \sigma_2$, which implies $\vdash V : \sigma_1 \vee \sigma_2$. But type τ cannot be deduced for $M[V/v]$. In fact, it is easy to verify that $M[V/v]$ diverges, since it reduces to $\Omega\|\Omega\|\Omega\mathbf{I}\Delta\|\Omega$. Therefore by 3.5.6*(ii)* $M[V/v]$ has only types equivalent to ω.

(ii) Immediate from *(i)*.

(iii) An example is $L \equiv \lambda x.(x\mathbf{I}\Delta\Delta \| x\Delta\mathbf{I}\Delta)$, $M \equiv \mathbf{K}$ and $N \equiv \mathbf{O}$. Analogously to (5) and (6) (in the proof of *(i)*) we have $x : \sigma_1 \vdash x\mathbf{I}\Delta\Delta : \tau$ (7) and $x : \sigma_2 \vdash x\Delta\mathbf{I}\Delta : \tau$ (8), which imply $\vdash L : \sigma_1 \to \tau$ (9) and $\vdash L : \sigma_2 \to \tau$ (10). Then from (3), (4), (9), and (10) we conclude $\vdash LM + LN : \tau$, but this type cannot be deduced for $L(M+N)$. Really, $L(\mathbf{K}+\mathbf{O})$ reduces to $\Omega\|\Omega$, which being divergent has only types equivalent to ω.

(iv) We choose $L \equiv \lambda x.(x\mathbf{I}\Delta\Delta + x\Delta\mathbf{I}\Delta)$, $M \equiv \mathbf{K}$ and $N \equiv \mathbf{O}$. Using (7) and (8) (from the proof of *(iii)*) we have $\vdash L : \sigma_1 \wedge \sigma_2 \to \tau$. (3) and (4) (from the proof of *(i)*) imply $\vdash \mathbf{K}\|\mathbf{O} : \sigma_1 \wedge \sigma_2$. Then $\vdash L(M\|N) : \tau$, but this type cannot be deduced for $LM\|LN$. The reason is again that this term diverges, since it reduces to $\Omega\|\Omega$. □

Notice that 3.5.7*(i)* says that we cannot relax the condition on the premises of rule ($\vee$E) (discussed at page 66), allowing W to be a partial value. This does not mean that our calculus distinguishes internally between partial and total values. In fact we do not have a type (and therefore a test term by the following Full Abstraction Theorem) that characterizes all total values, as type $\omega \to \omega$ characterizes all convergent terms. The best we can do is the statement of Theorem 3.3.12*(ii)*.

As suggested by one of the referees, some further remarks are in order about the theory of the model. In fact equivalence classes of terms wrt $\simeq^{\mathcal{L}}$ build a distributive lattice in which $+$ is the meet and $\|$ is the join. Moreover Ω and all divergent terms are the bottom, since only types equivalent to ω can be deduced for them. Finally, the lattice has a top, namely the equivalence class of the term $(\lambda xy.xx)(\lambda xy.xx)$ (usually called the "ogre"). We recall that $(\lambda xy.xx)(\lambda xy.xx)$ is convertible to the fixed-point of $\mathbf{K}$, i.e. to $\mathbf{YK}$, using the standard β-rule. Moreover it is unsolvable of infinite order, since it reduces to $\lambda z_1 \ldots z_n.(\lambda xy.xx)(\lambda xy.xx)$ for any $n \geq 0$ using the standard β-rule. To prove that *Type* is the interpretation of $\mathbf{YK}$, in the remaining part of this section we will show that $\vdash (\lambda xy.xx)(\lambda xy.xx) : \sigma$ for all types σ.

Define for every $n \geq 1$, the following types

$$\sigma_1 \equiv \omega \qquad\qquad \sigma_{n+1} \equiv \tau_n \wedge \sigma_n \qquad\qquad \tau_n \equiv \sigma_n \to \omega^n \to \omega.$$

3.5.8. LEMMA. *(i) For every $n \geq 1$, $\vdash \lambda xy.xx \colon \tau_n$.*

(ii) For every $n \geq 1$, $\vdash \lambda xy.xx \colon \sigma_n$.

PROOF. *(i)* We distinguish two cases.

Case $n = 1$.

$$\cfrac{\cfrac{\vdash xx \colon \omega}{\vdash \lambda y.xx \colon \omega \to \omega}\ (\to \mathrm{I}_n)}{\vdash \lambda xy.xx \colon \omega \to \omega \to \omega}\ (\to \mathrm{I}_n)$$

Case $n > 1$.

$$\cfrac{\cfrac{\cfrac{\cfrac{x{:}\sigma_n \vdash x \colon \tau_{n-1} \wedge \sigma_{n-1}}{x{:}\sigma_n \vdash x \colon \tau_{n-1}}\ (\leq) \qquad \cfrac{x{:}\sigma_n \vdash x \colon \tau_{n-1} \wedge \sigma_{n-1}}{x{:}\sigma_n \vdash x \colon \sigma_{n-1}}\ (\leq)}{x{:}\sigma_n \vdash xx \colon \omega^{n-1} \to \omega}\ (\to \mathrm{E})}{x{:}\sigma_n \vdash \lambda y.xx \colon \omega^n \to \omega}\ (\to \mathrm{I}_n)}{\vdash \lambda xy.xx \colon \sigma_n \to \omega^n \to \omega}\ (\to \mathrm{I}_n)$$

(ii) The case $n = 1$ is trivial. If $n > 1$, then $\sigma_n = \bigwedge_{i=1}^{n-1} \tau_i$, so that the thesis follows immediately from *(i)*. □

3.5.9. THEOREM. *(i) For every* $n \geq 1$, $\vdash (\lambda xy.xx)(\lambda xy.xx)\colon \omega^n \to \omega$.

(ii) For all types σ, $\vdash (\lambda xy.xx)(\lambda xy.xx)\colon \sigma$.

PROOF. *(i)* Just consider the derivations, for $n \geq 1$

$$\dfrac{\dfrac{\text{(by Lemma 3.5.8}\textit{(i)}\text{)}}{\vdash \lambda xy.xx\colon \tau_n} \quad \dfrac{\text{(by Lemma 3.5.8}\textit{(ii)}\text{)}}{\vdash \lambda xy.xx\colon \sigma_n}}{\vdash (\lambda xy.xx)(\lambda xy.xx)\colon \omega^n \to \omega}\ (\to \mathrm{E})$$

since $\tau_n \equiv \sigma_n \to \omega^n \to \omega$.

(ii) Immediate from *(i)* and 3.3.2. □

3.5.3. Full Abstraction Theorem

To establish the main result of this chapter we use the discriminability power of test terms.

3.5.10. THEOREM. *Let* M *be a closed term. Then* $\mathsf{T}_\sigma M\Downarrow \;\Leftrightarrow\; \vdash M : \sigma$.

PROOF. If $\mathsf{T}_\sigma M\Downarrow$ then, by 3.3.20, $\vdash \mathsf{T}_\sigma M : \omega \to \omega$. By 3.3.10*(vi)* it follows that $\mathsf{T}_\sigma : \rho \to \omega \to \omega$ for some ρ such that $\vdash M : \rho$. By 3.5.2*(ii)* it has to be true that $\rho \leq \sigma$, so that $\vdash M : \sigma$ using rule $(\leq)$. vice-versa $\vdash M : \sigma$ implies $\vdash \mathsf{T}_\sigma M : \omega \to \omega$ by 3.5.2*(ii)*, so we conclude $\mathsf{T}_\sigma M\Downarrow$ by 3.5.6*(i)*. □

3.5.11. THEOREM (Full Abstraction).

$$M \sqsubseteq^{\mathcal{L}} N \Leftrightarrow M \sqsubseteq^{\mathcal{O}} N.$$

PROOF. $(\Rightarrow)$ (Adequacy) Since $\sqsubseteq^{\mathcal{L}}$ is a precongruence, $M \sqsubseteq^{\mathcal{L}} N$ implies that, for any context $C[\,]$ closing both M and N, if $\vdash C[M] : \omega \to \omega$ then $\vdash C[N] : \omega \to \omega$. It follows that

$$C[M]\Downarrow \;\Rightarrow \vdash C[M] : \omega \to \omega \text{ by 3.3.20 } \Rightarrow \vdash C[N] : \omega \to \omega \Rightarrow C[N]\Downarrow \text{ by 3.5.6}(i).$$

$(\Leftarrow)$ (Completeness) $M \not\sqsubseteq^{\mathcal{L}} N \Rightarrow \exists\Gamma, \sigma.\ \Gamma \vdash M : \sigma \;\&\; \Gamma \not\vdash N : \sigma$. Let $FV(MN) = \{\chi_i \mid 1 \leq i \leq n\}$, $\Gamma = \{\chi_i : \tau_i \mid 1 \leq i \leq n\}$ and $\tau = \tau_1 \to \cdots \to \tau_n \to \sigma$, then $\vdash \lambda\chi_1 \ldots \chi_n.M : \tau$ and $\not\vdash \lambda\chi_1 \ldots \chi_n.N : \tau$ by 3.3.10*(iii)*, *(iv)*. Therefore, choosing $C[\,] = \mathsf{T}_\tau(\lambda\chi_1 \ldots \chi_n.[\,])$, we have by 3.5.10 that $C[M]\Downarrow$ and $C[N]\Uparrow$, which imply $M \not\sqsubseteq^{\mathcal{O}} N$. □

A natural question, posed by one of the referees, is whether $\simeq^{\mathcal{L}}$ on the sub-calculus without $+$ coincides with Boudol's equivalence [22]. The answer is negative, since our calculus is more discriminating also for pure λ-terms. For example, Boudol equates $\lambda x.xx$ and its η-expansion $\lambda x.x(\lambda y.xy)$: this is proved in [87]. Instead, in our calculus with $+$, $\lambda x.xx$ is strictly better than $\lambda x.x(\lambda y.xy)$. This is shown by the fact that $\vdash \lambda x.xx : \sigma \wedge (\sigma \to \tau) \to \tau$, while $\lambda x.x(\lambda y.xy)$ does not have this type, where $\sigma \equiv (\mu \to \omega \to \mu) \vee (\omega \to \mu \to \mu)$, $\tau \equiv \omega \to \mu$ and $\mu \equiv \omega \to \omega$. This can be checked by verifying that the test term corresponding to $\sigma \wedge (\sigma \to \tau) \to \tau$ converges when applied to $\lambda x.xx$ and diverges when applied to $\lambda x.x(\lambda y.xy)$. Another way of showing this is to consider the application of these terms to $P + Q$, where $P \equiv \lambda v.v(v\mathbf{O})$ and $Q \equiv \lambda yx.x\Omega$. In fact we have that $(\lambda x.xx)(P + Q)$ converges, while $(\lambda x.x(\lambda y.xy))(P + Q)$ diverges. In figure 3.3 we show the whole reduction tree of $(\lambda x.xx)(P + Q)$ and an infinite reduction path out of $(\lambda x.x(\lambda y.xy))(P + Q)$.

$$\begin{tikzcd}[column sep=small, row sep=small]
 & & (\lambda x.xx)(P+Q) \arrow[d] & \\
 & & (P+Q)(P+Q) \arrow[dl] \arrow[dr] & \\
 & P(P+Q) \arrow[dl] \arrow[dr] & & Q(P+Q) \arrow[d] \\
PP \arrow[d] & & PQ \arrow[d] & \lambda x.x\Omega \\
P(P\mathbf{O}) \arrow[d] & & Q(Q\mathbf{O}) \arrow[d] & \\
P(\mathbf{O}(\mathbf{O}\mathbf{O})) \arrow[d] & & \lambda x.x\Omega & \\
P\mathbf{I} \arrow[d] & & & \\
\mathbf{I}(\mathbf{I}\mathbf{O}) \arrow[d] & & & \\
\mathbf{I}\mathbf{O} \arrow[d] & & & \\
\mathbf{O} & & &
\end{tikzcd}$$

(a)

$$\begin{array}{c}
(\lambda x.x(\lambda y.xy))(P+Q) \\
\downarrow \\
(P+Q)(\lambda y.(P+Q)y) \\
\downarrow \\
P(\lambda y.(P+Q)y) \\
\downarrow \\
(\lambda y.(P+Q)y)((\lambda y.(P+Q)y)\mathbf{O}) \\
\downarrow \\
(P+Q)((\lambda y.(P+Q)y)\mathbf{O}) \\
\downarrow \\
P((\lambda y.(P+Q)y)\mathbf{O}) \\
\downarrow \\
P((P+Q)\mathbf{O}) \\
\downarrow \\
P(Q\mathbf{O}) \\
\downarrow \\
P(\lambda x.x\Omega) \\
\downarrow \\
(\lambda x.x\Omega)((\lambda x.x\Omega)\mathbf{O}) \\
\downarrow \\
(\lambda x.x\Omega)\mathbf{O}\Omega \\
\downarrow \\
\mathbf{O}\Omega\Omega \\
\downarrow \\
\Omega \\
\downarrow \\
\vdots
\end{array}$$

(b)

Figure 3.3: *(a)* The reduction tree of $(\lambda x.xx)(P + Q)$. *(b)* An infinite reduction path out of $(\lambda x.x(\lambda y.xy))(P + Q)$.

Really, we strongly conjecture that the restriction of $\simeq^{\mathcal{L}}$ to the pure λ-calculus coincides with the equality of Lévy-Longo trees associated with λ-terms. This equivalence relation between λ-terms, which we denote by $\simeq^{\mathcal{T}}$, was defined in [66].

From one side, Sangiorgi in [87] proves that using a set of well-formed operators and comparing terms through bisimulation, one always obtains an equivalence relation (we call $\simeq^{\mathcal{OP}}$) which includes $\simeq^{\mathcal{T}}$. Indeed, we have that $M \simeq^{\mathcal{T}} N$ implies $M \simeq^{\mathcal{OP}} N$ for all λ-terms M, N.

Now our concurrent λ-calculus respects the conditions on well-formed operators of [87]. Moreover we compare terms through contexts, which equate in our case more than bisimulation. For example, we equate $\mathbf{I} + \Omega$ and Ω, which are not bisimilar. Then we conclude that $M \simeq^{\mathcal{T}} N$ implies $M \simeq^{\mathcal{L}} N$ for all λ-terms M, N.

From the other side, given two different Lévy-Longo trees, we believe that there is always a type which can be deduced only for one of the corresponding terms. Note that we cannot use Sangiorgi result that the non-deterministic choice is sufficient to obtain the discriminating power of Lévy-Longo trees, since we compare terms by means of contexts instead of using bisimulation.

The coincidence of $\mathcal{L}$ and $\mathcal{T}$ would be another argument showing the robustness of the theory induced by $\simeq^{\mathcal{T}}$. In fact the same theory is also induced by:

- the encoding of λ-calculus into π-calculus [87];
- Plotkin-Scott-Engeler models [77];
- contexts with multiplicities [23].

Lastly we want to discuss the negative results of [90]. In that paper, Sieber considers a call-by-value version of PCF enriched with a non-deterministic choice operator (he calls this language PCF_{nv}). He proves that for this language the full abstraction of the Smyth powerdomain semantics fails in an irreparable way. Indeed, there is no extension of PCF_{nv} by computable operators, for which this semantics is fully abstract.

Notice that PCF_{nv} is a typed language, while our concurrent λ-calculus is type free and we use types only to describe its semantics. Indeed it is not surprising that an untyped calculus has a stronger power in discriminating internally between terms than a typed one. In particular, Böhm's theorem [17] (10.4) does not hold for the simply typed λ-calculus.

The same phenomenon happens here, since one can easily discriminate the terms which are at the basis of Sieber's proof. Omitting types, these terms are

$$\begin{array}{rcl} M_1 & = & \lambda f.\,\mathbf{if}\ f(\lambda x.0)\ \mathbf{then}\ f(\lambda x.\Omega)\ \mathbf{else}\ \Omega\ \mathbf{fi} \\ M_2 & = & \lambda f.\,\mathbf{if}\ f(\lambda x.0)\ \mathbf{and}\ f(\lambda x.\Omega)\ \mathbf{then}\ 0\ \mathbf{else}\ \Omega\ \mathbf{fi}. \end{array}$$

where 0 is interpreted as "true" and all other integers as "false". We can encode this example in the pure λ-calculus with the η-reduction rule using Church's numerals. Recall that $\mathbf{O}$ represents Church's zero and that $\mathbf{I}$ is η-convertible to Church's one. If we denote by $(\ )^\dagger$ this translation, it is easy to check that a correct choice is the following

$$(\ \mathbf{if}\ A\ \mathbf{then}\ B\ \mathbf{else}\ C\ \mathbf{fi})^\dagger\ = A^\dagger\mathbf{O}(\mathbf{K}B^\dagger)C^\dagger \qquad\qquad (A\ \mathbf{and} B)^\dagger\ = A^\dagger\mathbf{O}B^\dagger,$$

since for Church's numerals it holds $(\mathbf{n}+\mathbf{1})\mathbf{O} \xrightarrow{*} \mathbf{O}$. Therefore we have

$$\begin{array}{rcl} M_1^\dagger & = & \lambda f.f(\lambda x.\mathbf{O})\mathbf{O}(\mathbf{K}(f(\lambda x.\Omega)))\Omega \\ M_2^\dagger & = & \lambda f.f(\lambda x.\mathbf{O})\mathbf{O}(f(\lambda x.\Omega))\mathbf{O}(\mathbf{KO})\Omega. \end{array}$$

Now the pure λ-calculus semi-separates easily these terms. It suffices to choose $F \equiv \lambda x_1 \ldots x_6.x_6$, since $M_1^\dagger F \xrightarrow{*} \mathbf{O}$ and $M_2^\dagger F \xrightarrow{*} \Omega$. It is clear that PCF_{nv} does not allow to define a well-typed operator which behaves like F. Really, F applied to any term returns $\lambda x_1 \ldots x_5.x_5$, which does not have the type of integers.

Chapter 4

A Convex Powerdomain over Lattices

4.1. Introduction

Set theoretically a multifunction (or many-valued function) is a function with values in some powerset. If one takes domains and continuous functions as abstract counterparts of data types and computable functions, the problem of a theory of multifunctions reduces to the problem of a theory of powerdomains. Different constructions are possible, essentially because of different treatments of the undefined object. In the theory of powerdomains (see [82, 92, 50] and [43] for an elementary presentation) three main constructions have been devised (to know about other constructions, see e.g. [41, 42, 44]): the lower, upper and convex powerdomains. The first two could be grouped together as they are, though different, domains of total sets. Roughly speaking, the lower powerdomain identifies those sets which have the same defined objects, while the upper powerdomains identifies with the totally undefined set any set having the undefined object among its elements. The convex powerdomain, on the other hand, allows "partial" sets, which may contain the undefined object without either ignoring it or collapsing to the undefined set.

It could be asked, however, what are the right powerdomains to model parallelism and non-determinism respectively. There is a general agreement to consider the parallel composition of two functions as the best behaved among them with respect to the partial ordering of their ranges. In case of functions this leads to the interpretation of the parallel operator as a join: in case of multifunctions the lower powerdomain is the right choice. Indeed any upper semilattice which has all directed joins is a complete lattice. On the other hand, any prime algebraic lattice is isomorphic to the lower powerdomain of its compact coprime elements; in this case the continuous union operation of the powerdomain coincides with the join. This reinforces the intuition that lower powerdomains and lattices are natural models for such "parallel functions" (see e.g. [22] for an application to parallel extensions of both the lazy, call-by-name and call-by-value lambda calculus). This is the choice done in previous chapters.

Things are less clear when modeling non-determinism. In this case, if we want to keep it distinct from parallelism (see [65] for some reasons to do this), we are left either with a theory of total sets based on the upper powerdomain (as in chapters 2 and 3), or with some theory of partial sets. Unfortunately the category **ALG** of algebraic lattices is not closed under the convex powerdomain, which is a serious drawback if we wish to model a calculus in which both non-determinism and parallelism are present (like in [78, 34]), by partial sets.

To solve this problem we introduce a new powerdomain functor $\mathcal{P}$ such that the category of algebraic lattices is closed under $\mathcal{P}$. Let D be an algebraic lattice; following a general pattern, we consider the set $M(D)$ of finite non-empty sets of compacts in D, and define a preorder over this set by

$$u \preceq v \Leftrightarrow [\forall e \in v\, \exists d \in u.\, d \sqsubseteq e] \;\&\; \bigsqcup u \sqsubseteq \bigsqcup v, \tag{4.1}$$

The quotient of $M(D)$ under $\preceq$ is a sup-semilattice. It follows that, if we take $\mathcal{P}(D)$ as its ideal completion, then we obtain a complete lattice, which is of course algebraic. Continuous singleton $\{\!|\_|\!\}$ and union $\uplus$ operations are defined in the standard way, and the action of $\mathcal{P}$ over a continuous map $f : D \to E$ of algebraic lattices is then given by the unique continuous extension of the map

$$f^{\dagger}(\{\!|d_1|\!\} \uplus \cdots \uplus \{\!|d_n|\!\}) = \{\!|f(d_1)|\!\} \uplus \cdots \uplus \{\!|f(d_n)|\!\} \uplus \{\!|f(\bigsqcup_{i \leq n} d_i)|\!\}. \tag{4.2}$$

It follows that $\mathcal{P}$ is a functor and that $\mathcal{P}(D)$ is actually a free construction, and indeed it is initial in the category of the $\mathcal{P}$-algebras. The latter is the subcategory of **ALG** whose objects are endowed with a continuous binary operation $*$ which is idempotent, commutative and associative, and which satisfies the law

$$(d_1 * d_2) \sqcup d_3 = (d_1 \sqcup d_3) * (d_2 \sqcup d_3).$$

Essentially $*$ is the algebraic counter-part of the union operator.

Morphisms of $\mathcal{P}$-algebras are continuous maps satisfying the law

$$f(d_1 * d_2) = f(d_1) * f(d_2) * f(d_1 \sqcup d_2).$$

$\mathcal{P}(D)$ is a powerdomain of partial sets, and it is similar to the convex powerdomain. More precisely the closure operator associated to $\mathcal{P}$ has as fixed points exactly those subsets of D which are convex (X is convex if $X \ni x \sqsubseteq y \sqsubseteq z \in X$ implies $y \in X$) and closed under arbitrary join.

Let us concentrate on the set of compact elements $K(\mathcal{P}(D))$ of a lattice $\mathcal{P}(D)$. There are three kinds of objects: the bottom element, that is $\bot_{\mathcal{P}(D)} = \{\!|\bot_D|\!\}$; total objects of the shape $\{\!|d_1, \ldots, d_n|\!\}$ (an abbreviation for $\{\!|d_1|\!\} \uplus \cdots \uplus \{\!|d_n|\!\}$), where $d_i \neq \bot$ for all i; finally objects of the shape $\{\!|d_1, \ldots, d_n, \bot|\!\}$, the partial objects. Two choices for the set of values seem to propose themselves. Either a value is any object which is different from $\bot_{\mathcal{P}(D)}$, call this set $\mathcal{V}^{\flat}$; or the set of values coincides with the set of total objects, call this set $\mathcal{V}^{\sharp}$ (strictly speaking, these definitions concern just compact values, but they uniquely extend to the whole $\mathcal{P}(D)$).

To compare these two possibilities, let us consider the spaces of call-by-value functions that they determine. First of all these have to be strict functions, but in presence of partial objects one has to be careful.

Let $g : \mathcal{P}(D) \to E$ be any continuous function; then its strict version is $g_{\bot} = \lambda s \in \mathcal{P}(D).$ if $s \neq \bot$ then $g(s)$ else $\bot$. This has the counter-intuitive effect, however, of mapping a partial object $\{\!|d, \bot|\!\}$, say, to a possibly total object $g(\{\!|d, \bot|\!\}) \not\supseteq \{\!|\bot|\!\}$. A way out seems to consider, at least when E is a $\mathcal{P}$-algebra, strict morphisms of $\mathcal{P}$-algebras. To be concrete, let $E = \mathcal{P}(E')$ and suppose that $g' = \mathcal{P}(f_{\bot})$, where $f : D \to E'$; then $g'(\{\!|d, \bot|\!\}) = \{\!|f(d), \bot|\!\}$.

We now relativize the above definition to a set $\mathcal{V} \subseteq \mathcal{P}(D)$ of values. If $f : D \to E$ then it determines the call-by-value function $f^{\mathcal{V}} : \mathcal{P}(D) \to \mathcal{P}(E)$, relative to $\mathcal{V}$:

$$f^{\mathcal{V}} = \lambda s \in \mathcal{P}(D).\ \text{if } s \in \mathcal{V} \text{ then } \mathcal{P}(f_{\bot})(s) \text{ else } \bot. \tag{4.3}$$

It is easy to see that $f^\flat(\{\!|d,\perp|\!\}) = \{\!|f(d),\perp|\!\}$, while $f^\sharp(\{\!|d,\perp|\!\}) = \perp$ (where $f^\flat = f^{\mathcal{V}^\flat}$ and $f^\sharp = f^{\mathcal{V}^\sharp}$). Moreover, if both d_1 and d_2 are different from $\perp$, then $f^\flat(\{\!|d_1,d_2|\!\}) = \{\!|f(d_1), f(d_2), f(d_1 \sqcup d_2)|\!\}$. In general this says that, for any $s,t \in \mathcal{P}(D)$,

$$f^\flat(s \uplus t) \in \mathcal{V}^\flat \;\Leftrightarrow\; f^\flat(s) \in \mathcal{V}^\flat \text{ or } f^\flat(t) \in \mathcal{V}^\flat \;\Leftrightarrow\; f^\flat(s \sqcup t) \in \mathcal{V}^\flat$$

which implies that, at least with respect this notion of value, the union and the join collapse in $\mathcal{P}(D)$. On the other hand

$$\begin{array}{lcl} f^\sharp(s \uplus t) \in \mathcal{V}^\sharp & \Leftrightarrow & f^\sharp(s) \in \mathcal{V}^\sharp \text{ and } f^\sharp(t) \in \mathcal{V}^\sharp \\ f^\sharp(s \sqcup t) \in \mathcal{V}^\sharp & \Leftrightarrow & f^\sharp(s) \in \mathcal{V}^\sharp \text{ or } f^\sharp(t) \in \mathcal{V}^\sharp \end{array}$$

Note that, if e.g. $s = \{\!|d_1,\perp|\!\}$ and $t = \{\!|d_2,\perp|\!\}$ then $s \sqcup t = \{\!|d_1,d_2,d_1 \sqcup d_2,\perp|\!\}$. Since we want to preserve the distinction between multifunctions arising from non-determinism and those which model parallel computations, our choice has to be $\mathcal{V} = \mathcal{V}^\sharp$.

We further investigate how our powerdomain can be endowed with an applicative structure, suitable to model some kind of type free λ-calculus. In case of the classical, type free λ-calculus, terms denote objects which are at the same time functions and elements in the domain (and the range) of these functions. In our case they are also sets of functions. Moreover the λ-calculus we think of includes both call-by-name and call-by-value abstractions. Therefore, as all functions are basic values (that is the non-bottom elements of the domain D of which we shall take the powerdomain $\mathcal{P}(D)$), the space of functions has to be lifted, so that even the everywhere undefined function will be different from the bottom (and hence from the singleton of the bottom). To make this precise, we consider the following system of domain equations:

$$\left\{ \begin{array}{lcl} D & = & \mathcal{P}(N_\perp) \\ N & = & [D \to D]. \end{array} \right. \tag{4.4}$$

If we want to underline that the objects we are dealing with are "sets", then we have to solve the equation

$$D = \mathcal{P}([D \to D]_\perp). \tag{4.5}$$

Otherwise, if the first class objects are functions, we get

$$N = [\mathcal{P}(N_\perp) \to \mathcal{P}(N_\perp)]. \tag{4.6}$$

Both solutions exist, since $\mathcal{P}$ turns out to be locally continuous, so that they can be obtained as direct limits.

Considering the equation (4.5), application is naturally defined (restricting again to the compacts) as follows:

$$\{\!|f_1,\ldots,f_m|\!\} \cdot d = \left\{ \begin{array}{ll} f_1(d) \uplus \cdots \uplus f_m(d) & \text{if } \{f_1,\ldots,f_m\} \subseteq [D \to D] \\ f_1(d) \uplus \cdots \uplus f_{m-1}(d) \uplus \perp & \text{if } \{f_1,\ldots,f_{m-1}\} \subseteq [D \to D] \text{ and } f_m = \perp. \end{array} \right. \tag{4.7}$$

On the other hand, if we consider (4.6), we have a particular case of the construction in [74]. Indeed it turns out that the functor $T = \mathcal{P} \circ (\_)_\perp$ is a strong monad, and that the solution N is a call-by-name T-reflexive object.

Working inside the category of algebraic lattices, with Scott-continuous maps as morphisms, has also the technical advantage that the domain can be described using the simpler theory of Extended Abstract Type Structures (see [24]) instead of the theory of Domain Prelocales (see [5]). EATS are isomorphic to the join-semilattice of compact elements of ω-algebraic lattices (there is no loss of generality as the initial solution of the equation (4.5) is the colimit of an ω-chain starting with the trivial one element domain, and therefore its compacts are denumerable). A minimal EATS is generated from a countable set of type constants, closing under a binary operator $\wedge$. Over types it is defined a preorder $\leq$ such that, taking the quotient, $\sigma \wedge \tau$ is the meet of σ and τ. Given an EATS, the whole domain is recovered from filters of types, ordered by subset inclusion.

EATS's are instances of Information Systems (see [89]), and represent several domain constructors. As usual, intersection represents join: no compatibility restriction is required since we work with lattices (for more details see [27]). The constructors involved in equation (4.5) are lifting, exponentiation and the powerdomain functor $\mathcal{P}$. The lifting is easily represented by adding a constant ω for the top, so that the filter $\uparrow\omega$ will be the newly added bottom of the domain of filters. The space of continuous functions is represented by arrow types $\sigma \to \tau$, which are intrinsic to EATS's. By the way we recall that the arrow type constructor is contravariant in its first argument and covariant in the second, and moreover that the following equality hold:

$$\sigma \to (\tau \wedge \rho) = (\sigma \to \tau) \wedge (\sigma \to \rho).$$

It follows that the principal filters of arrow types represent step functions. The only exception to the contravariance-covariance of the arrow is the inequality

$$\sigma \to \omega \leq \omega \to \omega.$$

This has the consequence that the set of arrow types has a maximum, namely $\omega \to \omega$, so that, the function space has a minimum: $\lambda x.\,\bot$. It is then clear why we have to rule out the inequation $\omega \leq \omega \to \omega$ (which is on the contrary included in the preorder of Chapter 2), since this would collapse the bottom of the space $[D \to D]$ with the bottom of $[D \to D]_\bot$.

We are left with the functor $\mathcal{P}$. Therefore we need a type connective which is the EATS counter-part of the union. To this aim we introduce a binary operator $\oplus$, and the axioms involving $\oplus$ derive directly from the characterization of the algebras of the powerdomain functor (that is, ultimately, from the definition of $\preceq$). $\oplus$ is idempotent, commutative and associative and it satisfies the law:

$$(\sigma \oplus \tau) \wedge \rho = (\sigma \wedge \rho) \oplus (\tau \wedge \rho).$$

This, together with the covariance of $\oplus$ in both its arguments, implies that the following equation holds:

$$\sigma \oplus \tau = \sigma \oplus \tau \oplus (\sigma \wedge \tau),$$

which represents the property that "sets" in our powerdomain are closed under joins of their elements. We just remark that no special constructor is needed for the singleton operation, since all "singletons" will be represented exactly by those types that cannot be non-trivially equated to any type whose leftmost operator is $\oplus$.

We can now face the problem of the representation of the initial solution of equation (4.5). Indeed, applying to the case of EATS the technique to solve domain equations using Information Systems (see [61]), we know that it suffices, for each domain D_n in the direct limit $\lim_{\to} D_n$ (that gives the solution of (4.5)), to put into a single bag the types that represent its compacts, and then to take the space of filters. Now D_0 is the one element domain, hence no basic type constant but the constant ω is needed, so that the representation of the domain $D = \mathcal{P}([D \to D]_\bot)$ will simply be the set $\mathcal{F}$ of all filters of types generated from ω closing under $\wedge$, $\to$ and $\oplus$, which are (pre)-ordered as described above.

Usually one begins with a syntax, that is a language and a notion of evaluation, and looks for a model and an interpretation equating equivalent terms. In our exposition we go in the opposite direction. First an abstract structure of objects has been devised, and then we look for a syntax which is expressive with respect to that structure. The criterion of expressibility, as we shall see in the next sections, is adequacy and completeness (full abstraction).

To fix ideas, we start with the syntax of type free λ-calculus. We have to define both call-by-name and call-by-value abstractions, and we choose to introduce two sorts of variables, namely $x, y, \ldots$ and $v, w, \ldots$ so that $\lambda x.M$ and $\lambda v.M$ are call-by-name and call-by-value abstractions, respectively.

Then we enrich this syntax (see e.g. [78, 34]) by adding operators for set construction (implicitly representing the non-deterministic internal choice) and parallel evaluation: "+" and "$\|$" respectively. The intended interpretation of these operators are the powerdomain union and the join of the lattice.

Coming to the reduction relation, usually written $\longrightarrow$, we consider rules (β), (μ) and (ν) from the classical λ-calculus (see [17]), but not rule (ξ). This is due to the fact that a λ-abstraction always denotes a function (to be identified with the singleton of that function), then a total object, that is an element of $\mathcal{V}$; so that it doesn't make sense to evaluate further its body (see [81]).

To cope with commutativity, associativity and idempotency of $+$ and $\|$ we introduce a congruence relation $\approx$ such that e.g. $M + N \approx N + M$, and then put

$$M \approx M',\ M' \longrightarrow N',\ N' \approx N \ \Rightarrow\ M \longrightarrow N. \tag{4.8}$$

Beside that, the reduction rules for $+$ and $\|$ will be the same: first if $M \longrightarrow M'$, then both $M + N \longrightarrow M' + N$ and $M\|N \longrightarrow M'\|N$. Second, since anything can be applied to anything else, we have to add rules for reducing an application where the leftmost term is either a sum or a parallel composition. These are

$$(M + N)L \longrightarrow (ML) + (NL) \qquad \text{and} \qquad (M\|N)L \longrightarrow (ML)\|(NL).$$

To understand these rules we may think, for the leftmost one, at the semantic definition of application (4.7). The rightmost one is explained by the fact that the ordering of the function space is the pointwise ordering, so that the join of two functions f and g is the map $\lambda x.\ f(x) \sqcup g(x)$. We account for the distributivity of the join over the union in the domain D by adding the rule

$$(M + N)\|L \longrightarrow (M\|L) + (N\|L). \tag{4.9}$$

Although also union distributes over join, we do not have the rule

$$(M\|N) + L \longrightarrow (M + L)\|(N + L), \tag{4.10}$$

essentially because, as it will appear clear, the calculus requires to bubble sums, and not to nest them.

The point of having the same operational semantics for both $+$ and $\|$ is that, as it is the case for the denotational semantics, we expect that they are discriminated by call-by-value functions. It is then crucial to give the definition of the set $\mathbf{V}$ of syntactical values and the corresponding rule of call-by-value β-contraction.

Starting with the definition of $\mathcal{V}$, we have the following grammar for the set $\mathbf{V}$ of syntactical values:

$$V ::= v \mid \lambda x.M \mid \lambda v.M \mid V + V \mid V \| M \mid M \| V. \tag{4.11}$$

Notice that a sum of values was not a value in chapter 3.
Indeed a call-by-value variable is meant to range over $\mathcal{V}$, hence it is a syntactical value. Abstractions, both call-by-name and call-by-value, are objects in $[D \to D]$, and therefore, as elements of D, singletons of non-bottom elements. If V_1 and V_2 are values, hence sets of non bottom elements, their union is such, therefore $V_1 + V_2$ is a value. Finally it is not difficult to show that $\mathcal{V}$ is an upper closed set, hence $V \| M$ and $M \| V$ are values for any M. Any other term, that is something of the shape MN or of the shape $M + N$ and $M \| N$ but not generated by the grammar (4.11), is not immediately recognizable as a value (even if it possibly will evaluate to a value).

We come to the definition of the call-by-value contraction. This rule takes usually the form

$$(\lambda v.M)V \longrightarrow M[V/v] \qquad \text{if } V \in \mathbf{V}. \tag{4.12}$$

In view of (4.3), however, this rule is sound just in case V is a syntactical value that denotes a singleton. Indeed, if V has the shape $V_1 + \cdots + V_n$ (where each V_i is a singleton value), then (4.3) and (4.2) show that the effect of reducing $(\lambda v.M)V$ should be the following. First compute the "distribution" of the function denoted by $\lambda v.M$ over the elements of the "set" V, including their join:

$$(\lambda v.M)(V_1 + \cdots + V_n) \longrightarrow (\lambda v.M)V_1 + \cdots + (\lambda v.M)V_n + (\lambda v.M)(V_1 \| \cdots \| V_n) \tag{4.13}$$

and then apply (4.12).

To put this to use, it is essential to discriminate among values in general and singleton values. Tentatively one just drops $V + V$ from (4.11). Unfortunately this is not enough. Consider the term $(M + N)\|V$, where V represents a singleton. By (4.9) this evaluates to $(M\|V) + (N\|V)$, which doesn't denote a singleton in general. Here the trouble is that we cannot admit a semantics in which, if $V \xrightarrow{*} V'$, it is not the case that $(\lambda v.M)V$ and $(\lambda v.M)V'$ are equivalent (for the moment: denote the same object).

Even worse, we could have to consider the term $L\|V$, where L is not a sum itself, but just reduces to a sum: something which cannot be effectively foretold. This says that we cannot hope in any simple syntactical definition (like a grammar) which discriminates between syntactical values denoting singletons and terms which do not have such a denotation.

To approximate the solution we observe that this is similar to the general problem of recognizing whether a term denotes a value or not in the classical call-by-value λ-calculus (see [81]). In that case to evaluate $(\lambda v.M)N$ one reduces N until (eventually) a value is obtained, and then the contraction takes place. Here we do the same until a value is reached from N (if any), but then we have to distinguish among several cases. If V is a closed term in the subset $\mathbf{W}$ of $\mathbf{V}$ generated by the grammar

$$W ::= v \mid \lambda x.M \mid \lambda v.M \mid W \| W \tag{4.14}$$

then rule (4.12) can be safely applied, since V surely denotes a singleton value. We observe that in such a case V is irreducible. If V is in the subset $\mathbf{U}$ of $\mathbf{V}$ generated by the grammar

$$U ::= W \mid U \| M \mid M \| U \tag{4.15}$$

but $V \notin \mathbf{W}$, then V is not a normal form, even if it is always equivalent (up to distributivity of $\|$ over $+$) to a term of the shape $W \| N$. For a similar notion of value which is not irreducible see [22, 34]. In this case we cannot simply substitute V for v in M as in (4.12), since V could evaluate to a sum. Indeed, if $V \in \mathbf{U} \setminus \mathbf{W}$, then $V \approx W \| N$ for some $W \in \mathbf{W}$ and $N \notin \mathbf{U}$. It is clear that W is not guaranteed to be the "best" among W and N, since N could evaluate to something which is better behaved than W, at least in some contexts. Therefore we replace v by W and we put the result in parallel with $(\lambda v.M)V'$, where V' is obtained from V by one step of reduction.

$$(\lambda v.M)V \longrightarrow M[W/v] \| (\lambda v.M)V' \qquad \text{if } V \approx W \| N \ \& \ W \in \mathbf{W} \ \& \ N \notin \mathbf{U} \ \& \ V \longrightarrow V'. \quad (4.16)$$

Finally, if $V \in \mathbf{V} \setminus \mathbf{U}$ (in such a case it is a proper sum), then rule (4.13) applies.

The content of this chapter is essentially [8].

4.2. The Powerdomain Construction

In order to meet the requirements illustrated in the introduction, we construct a new powerdomain functor, based on a preorder which coincides with Egli-Milner preorder on those (finite) sets which contain a maximum.

Next we study the solution of a domain equation which yields an applicative structure in which objects are "sets" of functions from objects to objects.

We recall some standard definitions and notions. A *complete lattice* is a poset $\langle D, \sqsubseteq \rangle$, in which every subset $X \subseteq D$ has a sup ($\bigsqcup X$) and, therefore, an inf ($\sqcap X$). An element $c \in D$ is *compact* if and only if $c \sqsubseteq \bigsqcup X$ implies $c \sqsubseteq \bigsqcup Y$ for some finite $Y \subseteq X$, where X is an arbitrary subset of D. As usual $K(D)$ denotes the set of compact elements of D. For $X \subseteq D$, we write $\downarrow X = \{d \in D \mid \exists d' \in X.\ d \sqsubseteq d'\}$ and $K(d) = K(D) \cap \downarrow \{d\}$. A lattice is *algebraic* if it is complete, and for every $d \in D$ $K(d)$ is directed and $d = \bigsqcup K(d)$. D is *ω-algebraic* if it is algebraic and such that the set $K(D)$ is denumerable.

Let $M(D)$ be the set of all finite non-empty subsets of $K(D)$. We introduce three preorders on $M(D)$: Smyth's preorder ($\sqsubseteq^S$), Egli-Milner preorder ($\sqsubseteq^{EM}$) and the new preorder ($\preceq$).

4.2.1. Definition. Let $u, v \in M(D)$:
(i) $u \sqsubseteq^S v$ if and only if for all $e \in v$ there is $d \in u$ such that $d \sqsubseteq e$;
(ii) $u \sqsubseteq^{EM} v$ if and only if $u \sqsubseteq^S v$ and for all $d \in u$ there is $e \in v$ such that $d \sqsubseteq e$;
(iii) $u \preceq v$ if and only if $\bigsqcup u \sqsubseteq \bigsqcup v$ and $u \sqsubseteq^S v$;
(iv) $\simeq^S$, $\simeq^{EM}$ and $\simeq$ are the equivalence over $M(D)$ induced by $\sqsubseteq^S$, $\sqsubseteq^{EM}$ and $\preceq$ respectively.

As an immediate consequence of the above definition we have the following properties of $\preceq$.

4.2.2. Lemma. *For all $u, v \in M(D)$:*
(i) $u \simeq u \cup \{\bigsqcup u\}$;
(ii) $u \preceq v \Leftrightarrow u \cup \{\bigsqcup u\} \sqsubseteq^{EM} v \cup \{\bigsqcup v\}$.

$(M(D), \preceq)$ is a sup semilattice. Indeed the join is given by

$$u \,\tilde{\sqcup}\, v = \{\bigsqcup (u' \cup v') \mid u' \subseteq^{\text{fne}} u \ \& \ v' \subseteq^{\text{fne}} v\},$$

where $\subseteq^{\text{fne}}$ is short for "is a finite non-empty subset of". The correctness of this definition relies on the closure of $K(D)$ under finite join. By the way notice that $u \,\tilde{\sqcup}\, v$ is equivalent to $\{a \sqcup b \mid a \in u \ \& \ b \in v\}$.

A feature of this preorder is that set theoretic union distributes over $\tilde{\sqcup}$ and vice-versa, up to $\simeq$, that is

$$(u \,\tilde{\sqcup}\, v) \cup w \simeq (u \cup w) \,\tilde{\sqcup}\, (v \cup w) \quad \text{and} \quad (u \cup v) \,\tilde{\sqcup}\, w \simeq (u \,\tilde{\sqcup}\, w) \cup (v \,\tilde{\sqcup}\, w).$$

Recall that an *ideal* is a non-empty left-closed set, closed under upper-bounds of finite subsets. The *ideal completion* $Idl(P, \sqsubseteq)$ of the poset $(P, \sqsubseteq)$ is the set of all ideals ordered by subset inclusion. We now define our powerdomain constructor.

4.2.3. Definition. Let D be an algebraic lattice:

(i) $\mathcal{P}(D) = Idl(M(D), \preceq)$;

(ii) $\{\!|\_|\!\} : D \to \mathcal{P}(D)$ is defined by: $\{\!|d|\!\} = \{v \in M(D) \mid \exists c \in K(d)\,.\, v \preceq \{c\}\}$;

(iii) $\uplus : \mathcal{P}(D) \times \mathcal{P}(D) \to \mathcal{P}(D)$ is defined by: $s \uplus t = \downarrow\{u \cup v \mid u \in s,\ v \in t\}$.

It is easy to check that $\{\!|\_|\!\}$ and $\uplus$ are well defined. To shorten notation $\{\!|d_1, \ldots, d_n|\!\}$ will abbreviate $\{\!|d_1|\!\} \uplus \ldots \uplus \{\!|d_n|\!\}$.

$\mathcal{P}(D)$ is naturally ordered by subset inclusion, and moreover it is a complete lattice

$$\bigvee X = \{u \in M(D) \mid \exists n, u_1, \ldots, u_n \in \bigcup X.\ u \preceq u_1 \,\tilde{\sqcup}\, \cdots \,\tilde{\sqcup}\, u_n\} \quad \text{for an arbitrary } X \subseteq \mathcal{P}(D).$$

By construction, $\mathcal{P}(D)$ is algebraic, with basis $K(\mathcal{P}(D)) = \{\downarrow u \mid u \in M(D)\}$. Therefore, if D is ω-algebraic, then $\mathcal{P}(D)$ is such.

The distributivity of join with respect to union in $M(D)$ induces that of join over $\uplus$ in $\mathcal{P}(D)$, and vice-versa:

$$(r \vee s) \uplus t = (r \uplus t) \vee (s \uplus t) \quad \text{and} \quad (r \uplus s) \vee t = (r \vee t) \uplus (s \vee t).$$

The powerdomain $\mathcal{P}(D)$ enjoys some set theoretical properties, among which the most interesting is that the set of $\uplus$-irreducible elements coincides with the set of "singletons".

4.2.4. Definition. Let D be an algebraic lattice, then:

(i) the set $\mathcal{C}(\mathcal{P}(D))$ of $\uplus$-*irreducible elements* of $\mathcal{P}(D)$ is defined by:

$$r \in \mathcal{C}(\mathcal{P}(D)) \;\Leftrightarrow\; \forall s, t \in \mathcal{P}(D).\ s \uplus t \subseteq r \Rightarrow s \vee t \subseteq r;$$

(ii) given $s \in \mathcal{P}(D)$ define $\mathcal{C}(s) = \{C \subseteq^{\text{fne}} \mathcal{C}(\mathcal{P}(D)) \mid \biguplus C \in K(s)\}$.

Notice that for all s and for all $C \in \mathcal{C}(s)$, $C \subseteq K(D)$ (otherwise a contradiction immediately arises to $\biguplus C \in K(s)$).

4.2.5. Lemma. *Let D be an algebraic lattice, then:*

(i) the set $\mathcal{C}(\mathcal{P}(D))$ is the image of D under the mapping $\{\!|\_|\!\}$;

(ii) for all $s \in K(\mathcal{P}(D))$, if $s \neq \bot$ then there exists $C \subseteq^{\text{fne}} \mathcal{C}(\mathcal{P}(D))$ such that $s = \biguplus C$;

(iii) for all $s \in \mathcal{P}(D)$, $s = \bigvee\{\biguplus C \mid C \in \mathcal{C}(s)\}$.

Proof.

(i) For any $d \in D$ and $s, t \in \mathcal{P}(D)$, we show that if $s \uplus r \subseteq \{\!|d|\!\}$, then $s \vee t \subseteq \{\!|d|\!\}$. Notice that by Definition 4.2.1 $u \preceq \{c\}$ iff $\bigsqcup u \sqsubseteq c$. Given any $w \in s \vee t$ there are $u, v \in M(D)$ such that $u \in s, v \in t$ and $w \preceq u \,\tilde{\sqcup}\, v$. This implies that

$$\bigsqcup w \sqsubseteq \bigsqcup (u \,\tilde{\sqcup}\, v) = \bigsqcup (u \cup v),$$

where the joins are taken in D. By the definition of $\{\!|d|\!\}$, $s \uplus r \subseteq \{\!|d|\!\}$ implies that there exists $c \in K(d)$ such that $u \cup v \preceq \{c\}$, and also that $\bigsqcup(u \cup v) \sqsubseteq c$. Therefore $\bigsqcup w \sqsubseteq c$, so we conclude $w \in \{\!|d|\!\}$. This shows that $s \vee t \subseteq \{\!|d|\!\}$, by the arbitrariness of w.

Vice-versa suppose that $s \in \mathcal{P}(D)$; if $u \in s$ and $u = \{e_1, \ldots, e_n\}$ then $\{\!|e_1|\!\} \uplus \cdots \uplus \{\!|e_n|\!\} \subseteq s$. If s is $\uplus$-irreducible, this implies $\{\!|e_1|\!\} \vee \cdots \vee \{\!|e_n|\!\} = \{\!| \bigsqcup_{i \leq n} e_i |\!\} \subseteq s$.

Given an $\uplus$-irreducible s, let $S = \bigcup\{u \mid u \in s\}$, and $d = \bigsqcup S$. We show that S is directed. If $c, c' \in S$ then there are $u, v \in s$ such that $c \in u, c' \in v$. Since s is an ideal with respect to $\preceq$, $u \,\tilde{\sqcup}\, v \in s$. Moreover $c \sqcup c' \in u \,\tilde{\sqcup}\, v$, and therefore $c \sqcup c' \in S$.

Now on one hand we have

$$u \in s \Rightarrow u \subseteq S \Rightarrow \bigsqcup u \in K(d) \Rightarrow u \in \{\!|d|\!\}.$$

On the other hand, using the $\uplus$-irreducibility of s, we have

$$\begin{array}{rcl} w \in \{\!|d|\!\} & \Rightarrow & \exists c \in K(d).\ w \preceq \{c\} \\ & \Rightarrow & \exists c' \in S.\ w \preceq \{c'\} \\ & \Rightarrow & \exists u \in s, c' \in u.\ w \preceq \{c'\} \\ & \Rightarrow & \exists u \in s, c_1, \ldots, c_n \in K(D).\ u = \{c_1, \ldots, c_n\}\ \&\ w \in \{\!| \bigsqcup_{i \leq n} c_i |\!\} \\ & \Rightarrow & w \in s, \end{array}$$

where the second implication follows from the facts that c is compact and S directed. We conclude that $s = \{\!|d|\!\}$.

(ii) If $s \in K(\mathcal{P}(D))$, then it is in the image of $M(D)$ under the natural embedding of $M(D)$ into $\mathcal{P}(D)$. Therefore, for some $e_1, \ldots, e_m \in K(D)$, $s = \{\!|e_1, \ldots, e_m|\!\}$. Now the statement follows from *(i)*.

(iii) Immediate from *(ii)* and the algebraicity of $\mathcal{P}(D)$. □

4.2.6. Corollary. *Let r be a compact and $\uplus$-irreducible element of $\mathcal{P}(D)$, then $r = s \uplus t$, where s and t are compacts, implies $r = s = t$.*

□

In order to get a domain suitable to our purposes we look for the initial solution, D°, of the domain equation

$$D = \mathcal{P}([D \to D]_\perp). \tag{4.17}$$

The existence of the domain D° is assured by general results on fixpoint domain equations, provided that we can prove that $\mathcal{P}$ is locally continuous. This last result is easily proved. Consider in fact any chain $f_n : D \to E$ with $f_n \sqsubseteq f_{n+1}$. Let $f = \bigsqcup_{n \in \mathsf{N}} f_n$. Then $\mathcal{P}(f)$ is the unique continuous extension of

$$\{\!|d_1, \ldots, d_m|\!\} \quad \mapsto \quad \{\!|f(d_1), \ldots, f(d_m), f(d_1 \sqcup \ldots \sqcup d_m)|\!\}.$$

Since

$$\begin{array}{rcl} \{\!|f(d_1), \ldots, f(d_m), f(d_1 \sqcup \ldots \sqcup d_m)|\!\} & = & \{\!| \bigsqcup_{n \in \mathsf{N}} f_n(d_1), \ldots, \bigsqcup_n f_n(d_m), \bigsqcup_{n \in \mathsf{N}} f_n((d_1 \sqcup \ldots \sqcup d_m)|\!\} \\ & = & \bigsqcup_{n \in \mathsf{N}} (\{\!|f_n(d_1), \ldots, f_n(d_m), f_n(d_1 \sqcup \ldots \sqcup d_m)|\!\}, \end{array}$$

we have that $\mathcal{P}(f)$ extends

$$\{\!|d_1, \ldots, d_m|\!\} \quad \mapsto \quad \bigsqcup_{n \in \mathsf{N}} (\{\!|f_n(d_1), \ldots, f_n(d_m), f_n(d_1 \sqcup \ldots \sqcup d_m)|\!\},$$

hence $\mathcal{P}$ is locally continuous.

Because of the continuity of $\mathcal{P}$, the direct limit technique to compute the initial solution D° of equation (4.17) carries over (being $(\_)_\perp$ locally continuous as well). It is the colimit of the ω-chain $(D_i, \varepsilon_i)_{i\in\mathbb{N}}$, where $D_0 = \{\perp\}$, $D_{n+1} = \mathcal{P}([D_n \to D_n]_\perp)$. The embedding e_0 is just $\lambda x.\ \perp$. To construct ε_{n+1} suppose $\varepsilon_n : D_n \to D_{n+1}$ be given; then $\varepsilon_{n+1} = \mathcal{P}(h_\perp)$ where $h = \lambda f \in [D_n \to D_n].\ \varepsilon_n \circ f \circ \pi_n$ and π_n is the projection from D_{n+1} to D_n determined by ε_n.

As in the case of the simpler equation modeling lazy λ-calculus, this is a non-trivial solution because of the lifting functor.

4.2.7. Definition. We call Ψ the isomorphism $D^\circ \simeq \mathcal{P}([D^\circ \to D^\circ]_\perp)$, but we shall often identify elements of both domains without explicitly mentioning it.

Because of Ψ we have a "union" operation over D°, formally defined by

$$d \uplus e = \Psi^{-1}(\Psi(d) \uplus \Psi(e)).$$

Compact elements of domains constructed by direct limit are images of compact elements of the approximating domains, so that, up to the embedding-projection of each D_n into D°, we have $K(D^\circ) = \bigcup_{n\in N} K(D_n)$. They are inductively characterized by the following lemma.

4.2.8. Lemma.

(i) $K(D_0) = \{\perp\}$;

(ii) $K(D_{n+1}) = \{\{\!|f_1, \ldots, f_m|\!\} \mid 1 \le m,\ f_1, \ldots, f_m \in K([D_n \to D_n]_\perp)\}$;

(iii) $K([D_{n+1} \to D_{n+1}]_\perp) = \{\bigsqcup_{i\le k}(e_i \Rightarrow d_i) \mid 1 \le k,\ d_i, e_i \in K(D_n)\} \cup \{\perp\}$,

where $(e \Rightarrow d)(x)$ *is the step function: if* $e \le x$ *then* d *else* $\perp$.

Proof. Immediate by the construction of D_n. □

Note that we drop the usual consistency condition in taking finite joins of step functions. This is sound since we work inside lattices.

We give an explicit definition of the application over D°.

4.2.9. Definition. Define the map $\cdot : \mathcal{P}([D^\circ \to D^\circ]_\perp) \times D^\circ \to D^\circ$ as the unique continuous extension of the map from $K(\mathcal{P}([D^\circ \to D^\circ]_\perp) \times D^\circ) = K(\mathcal{P}([D^\circ \to D^\circ]_\perp)) \times K(D^\circ)$ to $K(D^\circ)$ defined by

$$\{\!|f_1, \ldots, f_m|\!\} \cdot d = \begin{cases} f_1(d) \uplus \cdots \uplus f_m(d) & \text{if } \forall i \le m.\ f_i \ne \perp \\ f_1(d) \uplus \cdots \uplus f_{m-1}(d) \uplus \perp & \text{if } \forall i \le m-1.\ f_i \ne \perp \text{ and } f_m = \perp. \end{cases}$$

We finally include the definition of an operator $\coprod$ which roughly computes the join of the elements of a set.

4.2.10. Definition. Define $\coprod : D^\circ \to D^\circ$ as the unique continuous extension of

$$\begin{aligned} \coprod(\perp) &= \perp \\ \coprod(\{\!|d_1, \ldots, d_m|\!\}) &= \{\!|d_1 \vee \cdots \vee d_m|\!\}. \end{aligned}$$

4.3. The Filter Model

Logical presentations are suitable tools for defining and studying models [5]. We use intersection types, which provide simple descriptions of ω-algebraic lattices. Types are formed starting with the universal type ω and closing under the arrow, to describe the function space, under the intersection, corresponding to the semantic join, as in [22, 33, 34], and under a new type constructor $\oplus$, which is intended to model the semantic operation $*$ on compacts.

In determining the structure of the space of filters, i.e. the structure of the domain of the model, the choice of the ordering over types is crucial. We describe the order and its properties in subsection 4.3.1. Subsection 4.3.2 gives the isomorphism between the space of filters and the initial solution of our powerdomain equation.

4.3.1. The Set of Types

The set *Type* of types is defined by adding the type constructor "sum" $\oplus$ to the intersection types [18, 24]

$$\sigma ::= \omega \mid \sigma \to \tau \mid \sigma \wedge \tau \mid \sigma \oplus \tau.$$

In writing types we assume that $\wedge$ and $\oplus$ take precedence over $\to$. For its relevance and because of frequent usage, $\omega \to \omega$ will be abbreviated by ω_1.

We look for a partial order over types which corresponds in a natural way to the (quotient of the) preorder $\preceq$ defined on $M(D)$ in section 4.2. This will be clarified by the properties we will show of this order.

4.3.1. Definition (Preorder on Types). Let $\sigma \leq \tau$ be the least preorder over *Type* such that:

(i) $\sigma \leq \omega$;
(ii) $\sigma \to \omega \leq \omega \to \omega$;
(iii) $\sigma \leq \sigma \wedge \sigma$;
(iv) $\sigma \wedge \tau \leq \sigma, \quad \sigma \wedge \tau \leq \tau$;
(v) $\sigma \leq \sigma',\ \tau \leq \tau' \Rightarrow \sigma \wedge \tau \leq \sigma' \wedge \tau'$;
(vi) $\sigma \leq \sigma \oplus \sigma \leq \sigma$;
(vii) $\sigma \oplus \tau \leq \tau \oplus \sigma$;
(viii) $\sigma \oplus (\tau \oplus \tau') \leq (\sigma \oplus \tau) \oplus \tau'$;
(ix) $\sigma \leq \sigma',\ \tau \leq \tau' \Rightarrow \sigma \oplus \tau \leq \sigma' \oplus \tau'$;
(x) $(\sigma \oplus \sigma') \wedge \tau \leq (\sigma \wedge \tau) \oplus (\sigma' \wedge \tau)$;
(xi) $(\sigma \to \tau) \wedge (\sigma \to \tau') \leq \sigma \to (\tau \wedge \tau')$;
(xii) $\sigma' \leq \sigma,\ \tau \leq \tau' \Rightarrow \sigma \to \tau \leq \sigma' \to \tau'$;

Let $\sigma = \tau$ be defined as "$\sigma \leq \tau$ and $\tau \leq \sigma$". Then the quotient structure $\langle Type_{/=}, \leq_{/=}\rangle$ is a topped inf-semilattice, where the top is $[\omega]$ and $[\sigma] \wedge_{/=} [\tau] = [\sigma \wedge \tau]$. As usual we identify *Type* with $Type_{/=}$ and $\leq$ with $\leq_{/=}$.
We write $\sigma < \tau$ if and only if $\sigma \leq \tau$ and $\sigma \neq \tau$.

4.3.2. Proposition. The axiomatic presentation of 4.3.1 is consistent, and therefore $\langle Type, \leq\rangle$ is a non-trivial structure. In particular:

$$\omega \not\leq \omega \to \omega.$$

Proof. Let $f : Type \to \{\bot, \top\}$ be the following map into the two points lattice, ordered by $\bot \sqsubseteq \top$:

$$\begin{array}{lcl} f(\omega) & = & \top \\ f(\sigma \to \tau) & = & \bot \\ f(\sigma \wedge \tau) & = & f(\sigma \oplus \tau) \quad = \quad f(\sigma) \sqcap f(\tau). \end{array}$$

Checking through Definition 4.3.1 we see that

$$\sigma \leq \tau \;\Rightarrow\; f(\sigma) \sqsubseteq f(\tau),$$

for all σ and τ (actually f is a meet-semilattice morphism). Now

$$f(\omega) = \top \not\sqsubseteq \bot = f(\omega \to \omega)$$

implies the statement by contraposition. □

Notation. Because of its relevance and frequent occurrences in the technical development, we abbreviate $\omega \to \omega$ by ω_1.

We have the distributivity of $\wedge$ with respect to $\oplus$ and the vice-versa.

4.3.3. LEMMA. *For all* $\sigma, \sigma', \tau, \tau' \in Type$*:*
(i) $(\sigma \oplus \sigma') \wedge \tau = (\sigma \wedge \tau) \oplus (\sigma' \wedge \tau)$*;*
(ii) $(\sigma \wedge \sigma') \oplus \tau = (\sigma \oplus \tau) \wedge (\sigma' \oplus \tau)$.

Proof.

(i) Because of 4.3.1*(x)*, it suffices to show that $(\sigma \wedge \tau) \oplus (\sigma' \wedge \tau) \leq (\sigma \oplus \sigma') \wedge \tau$. This follows from $(\sigma \wedge \tau) \oplus (\sigma' \wedge \tau) \leq \sigma \oplus \sigma'$, by 4.3.1*(iv)* and *(ix)*, and from $(\sigma \wedge \tau) \oplus (\sigma' \wedge \tau) \leq \tau \oplus \tau \leq \tau$, by 4.3.1*(iv)*, *(ix)* and *(vi)*.

(ii) $(\sigma \wedge \sigma') \oplus \tau \leq (\sigma \oplus \tau) \wedge (\sigma' \oplus \tau)$ follows, by 4.3.1*(v)*, from $(\sigma \wedge \sigma') \oplus \tau \leq \sigma \oplus \tau$ and $(\sigma \wedge \sigma') \oplus \tau \leq \sigma' \oplus \tau$, which in turn hold by 4.3.1*(iv)* and *(ix)*.

For the opposite inclusion, using *(i)* of the present Lemma, we have:

$$\begin{array}{rcl} (\sigma \oplus \tau) \wedge (\sigma' \oplus \tau) & = & (\sigma \wedge (\sigma' \oplus \tau)) \oplus (\tau \wedge (\sigma' \oplus \tau)) \\ & = & (\sigma \wedge \sigma') \oplus (\sigma \wedge \tau) \oplus (\sigma' \wedge \tau) \oplus (\tau \wedge \tau) \\ & \leq & (\sigma \wedge \sigma') \oplus \tau \oplus \tau \oplus \tau \\ & = & (\sigma \wedge \sigma') \oplus \tau. \end{array}$$ □

Notation. Let $I, J, H, K, \ldots$ be finite sets of indexes. Since $\wedge$ and $\oplus$ are commutative and associative, we will freely use the following notations:

$$\textstyle\bigwedge_{i \leq n} \sigma_i, \qquad \bigwedge_{i \in I} \sigma_i, \qquad \bigoplus_{i \leq n} \sigma_i, \qquad \bigoplus_{i \in I} \sigma_i,$$

with obvious meanings.

A key equation, derivable from the above axioms, is $\sigma \oplus \tau = \sigma \oplus \tau \oplus (\sigma \wedge \tau)$. This can be generalized as follows:

4.3.4. PROPOSITION.

$$(\mathbf{P0}) \qquad \bigoplus_{i \leq n} \sigma_i = \bigoplus_{i \leq n} \sigma_i \oplus \bigwedge_{i \leq n} \sigma_i.$$

Proof. By induction on n.

$$\begin{array}{rcl}
\bigoplus_{i\leq n+1}\sigma_i \oplus \bigwedge_{i\leq n+1}\sigma_i & = & \bigoplus_{i\leq n}\sigma_i \oplus \sigma_{n+1} \oplus (\bigwedge_{i\leq n}\sigma_i \wedge \sigma_{n+1}) \\
& = & (\bigoplus_{i\leq n}\sigma_i \oplus \sigma_{n+1} \oplus \bigwedge_{i\leq n}\sigma_i) \wedge (\bigoplus_{i\leq n}\sigma_i \oplus \sigma_{n+1} \oplus \sigma_{n+1}) \\
& & \text{by distributivity} \\
& = & (\bigoplus_{i\leq n}\sigma_i \oplus \sigma_{n+1}) \wedge (\bigoplus_{i\leq n}\sigma_i \oplus \sigma_{n+1}) \\
& & \text{by induction and idempotency of } \oplus \\
& = & \bigoplus_{i\leq n+1}\sigma_i. \qquad \square
\end{array}$$

The following definition of *irreducible* types is analogously to that of coprime types (see page 3.3.1).

4.3.5. Definition. A type σ is *irreducible* if and only if

$$\forall \tau, \rho \in \mathit{Type}.\ \sigma = \tau \oplus \rho \Rightarrow \sigma = \tau = \rho.$$

IType is the set of irreducible types different from ω.

The exclusion of ω from *IType* is motivated by the fact that ω itself is irreducible (see 4.3.9 *(i)* below), and because irreducible types different from ω play a central role in the semantical construction. We first prove that any type is equivalent to a finite sum of irreducible types (possibly including ω), and that the restriction of the inequality relation to irreducible types characterizes the relation itself over the whole set *Type*.

It is useful to introduce a map Θ which drops external $\oplus$; we will see that Θ returns finite sets of irreducible types.

4.3.6. Definition. Let $\Theta : \mathit{Type} \to \wp(\mathit{Type})$ be defined as follows:

(i) $\Theta(\omega) = \{\omega\}$

(ii) $\Theta(\sigma \to \tau) = \{\sigma \to \tau\}$

(iii) $\Theta(\sigma \oplus \tau) = \Theta(\sigma) \cup \Theta(\tau)$

(iv) $\Theta(\sigma \wedge \tau) = \begin{cases} \{\sigma' \in \Theta(\sigma) \mid \omega \in \Theta(\tau)\} \cup \\ \{\tau' \in \Theta(\tau) \mid \omega \in \Theta(\sigma)\} \cup \\ \{\sigma' \wedge \tau' \mid \sigma', \tau' \not\equiv \omega \ \&\ \sigma' \in \Theta(\sigma) \ \&\ \tau' \in \Theta(\tau)\}. \end{cases}$

4.3.7. Example.

$$\begin{array}{rcl}
\Theta((\omega \to \omega) \wedge \omega) & = & \{\omega \to \omega\}, \\
\Theta((\omega \to \omega) \oplus \omega) & = & \{\omega \to \omega, \omega\}, \\
\Theta((\omega \to \omega) \wedge ((\omega \to \omega \to \omega) \oplus \omega)) & = & \{(\omega \to \omega) \wedge (\omega \to \omega \to \omega), \omega \to \omega\} \\
& = & \Theta(((\omega \to \omega) \wedge (\omega \to \omega \to \omega)) \oplus ((\omega \to \omega) \wedge \omega)).
\end{array}$$

4.3.8. Proposition. For all $\sigma, \tau \in \mathit{Type}$:

(i) $\sigma = \bigoplus_{\sigma' \in \Theta(\sigma)} \sigma'$;

(ii) $\sigma \leq \tau$ if and only if $\bigwedge_{\sigma' \in \Theta(\sigma)} \sigma' \leq \bigwedge_{\tau' \in \Theta(\tau)} \tau'$ and $\forall \sigma' \in \Theta(\sigma)\ \exists \tau' \in \Theta(\tau).\ \sigma' \leq \tau'$;

(iii) σ is irreducible if and only if $\forall \sigma' \in \Theta(\sigma).\ \sigma = \sigma'$;

(iv) *IType* and $\mathit{IType} \cup \{\omega\}$ are closed under intersection;

(v) for all I, σ_i, τ_i we have $\bigwedge_{i\in I}(\sigma_i \to \tau_i) \in \mathit{IType}$;

(vi) $\sigma' \in \Theta(\sigma)$ implies either that $\sigma' \equiv \omega$ or $\sigma' \equiv \bigwedge_{i\in I}(\rho_i \to \tau_i)$ for some ρ_i, τ_i and I;

(vii) $\Theta(\sigma) \subseteq \mathit{IType} \cup \{\omega\}$.

Proof.

(i) By easy induction on the definition of Θ using the distributivity of $\oplus$ over $\wedge$.

(ii) ($\Rightarrow$). By induction on the presentation of $\leq$.

($\Leftarrow$). Let $\Theta(\sigma) = \{\sigma_i \mid i \in I\}$ and $\Theta(\tau) = \{\tau_j \mid j \in J\}$. By hypothesis for all $i \in I$ there exists $j_i \in J$ such that $\sigma_i \leq \tau_{j_i}$ and $\bigwedge_{i\in I} \sigma_i \leq \bigwedge_{j\in J} \tau_j$. Now

$$\begin{array}{rcll} \sigma & = & \bigoplus_{i\in I} \sigma_i \oplus (\bigwedge_{i\in I} \sigma_i) & \text{by } (i) \text{ and } \mathbf{(P0)} \\ & \leq & \bigoplus_{i\in I} \tau_{j_i} \oplus (\bigwedge_{j\in J} \tau_j) & \text{by hypothesis} \\ & \leq & \bigoplus_{j\in J} \tau_j & \text{by monotonicity and idempotency of } \oplus \\ & = & \tau. & \end{array}$$

(iii) ($\Rightarrow$) follows by definition of $IType$ and by *(i)*. To see ($\Leftarrow$), let $\sigma = \tau \oplus \rho$; in this case $\Theta(\sigma) = \Theta(\tau) \cup \Theta(\rho)$. Then the hypothesis implies $\sigma = \tau'$ and $\sigma = \rho'$ for all $\tau' \in \Theta(\tau)$ and $\rho' \in \Theta(\rho)$. By *(i)* and the idempotency of $\oplus$ we get $\sigma = \tau$ and $\sigma = \rho$.

(iv) Immediate by *(iii)* and by definition of $\Theta(\sigma \wedge \tau)$.

(v) It is easy to verify that $\Theta(\bigwedge_{i\in I}(\sigma_i \to \tau_i)) = \{\bigwedge_{i\in I}(\sigma_i \to \tau_i)\}$, so we are done by *(iii)*.

(vi) By a straightforward induction on the definition of Θ.

(vii) Immediate by *(v)* and *(vi)*. □

4.3.1. Remark. The key point in Proposition 4.3.8 is *(ii)*. Comparing this statement with the definition of the (pre)order $\preceq$ over $M(D)$ it is apparent that $\leq$ and $\preceq$ are dual each other. A stronger correspondence is property (**P2**) in Lemma 4.3.10.

Point *(vi)* clearly gives a characterization of irreducible types, while *(i)* assures that the function Θ is the right tool to compute type inequalities.

4.3.9. Corollary.

(i) ω *is irreducible;*

(ii) $\omega_1 < \omega_1 \oplus \omega < \omega$*;*

(iii) $\sigma \leq \omega_1 \;\Leftrightarrow\; \omega \notin \Theta(\sigma) \;\Leftrightarrow\; \sigma \neq \sigma \oplus \omega$

(iv) $\sigma = \sigma \oplus \omega \;\&\; \sigma \neq \omega \Rightarrow \sigma \leq \omega_1 \oplus \omega$ *and* $\omega \in \Theta(\sigma)$.

Proof.

(i) For any $\sigma, \tau \in Type$,

$$\begin{array}{rcll} \omega \leq \sigma \oplus \tau & \Rightarrow & \omega \leq \bigwedge_{\sigma'\in\Theta(\sigma)} \sigma' \wedge \bigwedge_{\tau'\in\Theta(\tau)} \tau' & \text{by } 4.3.8(ii) \\ & \Rightarrow & \forall \sigma' \in \Theta(\sigma)\, \forall \tau' \in \Theta(\tau).\ \omega \leq \sigma' \;\&\; \omega \leq \tau' & \\ & \Rightarrow & \omega = \sigma = \tau & \text{by } 4.3.8(i). \end{array}$$

(ii) By 4.3.1*(i)*, *(vi)* and *(ix)* we prove

$$\omega_1 = \omega_1 \oplus \omega_1 \leq \omega_1 \oplus \omega \leq \omega \oplus \omega = \omega.$$

$\omega_1 \oplus \omega \not\leq \omega_1$ and $\omega \not\leq \omega_1 \oplus \omega$ are consequences of 4.3.8*(ii)* using the fact that $\omega \not\leq \omega_1$, by 4.3.2.

(iii) By 4.3.8*(i)* and *(ii)* we know that $\sigma \neq \sigma \oplus \omega$ if and only if $\omega \notin \Theta(\sigma)$. This implies by 4.3.8*(vi)* that all $\sigma' \in \Theta(\sigma)$ have the shape $\bigwedge_{i\in I}(\rho_i \to \tau_i)$, for some I, ρ_i, τ_i. By this, 4.3.1*(i)*, 4.3.1*(ii)*, 4.3.1*(xii)* 4.3.8*(i)* and 4.3.1*(vi)*, the statement follows.

(iv)

$$\begin{array}{rcl} \sigma = \sigma \oplus \omega \;\&\; \sigma \neq \omega & \Rightarrow & \sigma = \bigoplus_{\sigma'\in\Theta(\sigma)\ \sigma'\neq\omega} \sigma' \oplus \omega \\ & \Rightarrow & \sigma \leq \omega_1 \oplus \omega \end{array}$$

using 4.3.8*(i)*, 4.3.8*(vi)* and point *(iii)* of this Corollary.
$\omega \in \Theta(\sigma \oplus \omega)$ by definition, so $\omega \in \Theta(\sigma)$ by 4.3.8*(ii)*. □

The main properties of the relation $\leq$ are given in the following two lemmas. Notice that property **(P4)** implies the property stated in 3.3.7(ii).

4.3.10. LEMMA.

(P1) *$\sigma \oplus \tau \leq \rho$ implies that for some $\mu \geq \sigma$ one has $\rho = \mu \oplus \rho$;*

(P2) *Let σ_i, τ_j be irreducible types for $i \in I$, $j \in J$. $\bigoplus_{i\in I} \sigma_i \leq \bigoplus_{j\in J} \tau_j$ iff $\bigwedge_{i\in I} \sigma_i \leq \bigwedge_{j\in J} \tau_j$ and, for all $i \in I$, there is $j \in J$ such that $\sigma_i \leq \tau_j$;*

(P3) *$\bigoplus_{i\in I} \sigma_i \oplus \omega = (\bigwedge_{i\in I} \sigma_i) \oplus \omega$;*

(P4) *$\bigwedge_{i\in I}(\sigma_i \to \tau_i) \leq \bigwedge_{j\in J}(\mu_j \to \nu_j)$ if and only if, for all $j \in J$ such that $\nu_j \neq \omega$, there is $I_j \subseteq I$ such that $\mu_j \leq \bigwedge_{i\in I_j} \sigma_i$ and $\bigwedge_{i\in I_j} \tau_i \leq \nu_j$;*

Proof.

(P1) Let $\Theta(\sigma) = \{\sigma_i \mid i \in I\}$ and $\Theta(\rho) = \{\rho_h \mid h \in H\}$, for some I, σ_i, H, ρ_h. By 4.3.8*(ii)* we know that

$$\sigma \oplus \tau \leq \rho \Rightarrow \forall i \in I\, \exists\, h_i \in H.\ \sigma_i \leq \rho_{h_i}.$$

Set $\mu = \bigoplus_{i\in I} \rho_{h_i}$: then $\sigma \leq \mu$ by construction and $\rho = \mu \oplus \rho$ by the idempotency, commutativity and associativity of $\oplus$.

(P2) It follows easily from 4.3.8*(ii)*, considering that by definition $\Theta(\bigoplus_{i\in I} \sigma_i) = \bigcup_{i\in I} \Theta(\sigma_i)$, and that $\Theta(\bigoplus_{j\in J} \tau_j) = \bigcup_{j\in J} \Theta(\tau_j)$. Moreover $\sigma_i = \sigma'_i$ for all $\sigma'_i \in \Theta(\sigma_i)$ and $\tau_j = \tau'_j$ for all $\tau'_j \in \Theta(\tau_j)$ by 4.3.8*(iii)*, since σ_i and τ_j are irreducible by hypothesis.

(P3) Immediate from **(P2)**.

(P4) Define the map θ: $Type \to \wp(IType)$ as follows:

(i) $\theta(\omega) = \emptyset$;
(ii) $\theta(\sigma \to \tau) = \{\sigma \to \tau\}$;
(iii) $\theta(\sigma \oplus \tau) = \theta(\sigma \wedge \tau) = \theta(\sigma) \cup \theta(\tau)$.

Let $\theta(\alpha) = \{\sigma_i \to \tau_i \mid i \in I\}$ and $\theta(\beta) = \{\mu_j \to \nu_j \mid j \in J\}$ for some (possibly empty) I, J.

We will show the following more general statement:

$$\alpha \leq \beta \iff$$

$$[\theta(\beta) = \emptyset] \text{ or } [(\theta(\alpha) \neq \emptyset) \text{ and } (\forall j \in J.\ \nu_j \neq \omega \Rightarrow\ \exists\, I_j \subseteq I.\ \mu_j \leq \bigwedge_{i\in I_j} \sigma_i \ \&\ \bigwedge_{i\in I_j} \tau_i \leq \nu_j)].$$

The proof is by induction on the definition of $\leq$. All cases are immediate but transitivity. Suppose that $\alpha \leq \beta$ because of $\alpha \leq \gamma \leq \beta$ for some γ. Let $\theta(\gamma) = \{\xi_h \to \rho_h \mid h \in H\}$. If $\theta(\beta) \neq \emptyset$ then by induction $\theta(\gamma) \neq \emptyset$. The induction hypothesis gives us:

$$\forall h \in H.\ \rho_h \neq \omega \quad \Rightarrow \quad \exists\, I_h \subseteq I.\ \xi_h \leq \bigwedge_{i\in I_h} \sigma_i \ \&\ \bigwedge_{i\in I_h} \tau_i \leq \rho_h, \tag{4.18}$$
$$\forall j \in J.\ \nu_j \neq \omega \quad \Rightarrow \quad \exists\, H_j \subseteq H.\ \mu_j \leq \bigwedge_{h\in H_j} \xi_h \ \&\ \bigwedge_{h\in H_j} \rho_h \leq \nu_j. \tag{4.19}$$

For any $\nu_j \neq \omega$ set $H'_j = \{h \in H_j \mid \rho_h \neq \omega\}$, which is nonempty as a consequence of (4.19). Then

$$\mu_j \leq \bigwedge_{h\in H'_j} \xi_h \quad \text{and} \quad \bigwedge_{h\in H'_j} \rho_h \leq \nu_j. \tag{4.20}$$

By (4.18) we know that

$$h \in H'_j \quad \Rightarrow \quad \xi_h \leq \bigwedge_{i \in I_h} \sigma_i \ \& \ \bigwedge_{i \in I_h} \tau_i \leq \rho_h, \tag{4.21}$$

therefore

$$\begin{array}{ll} \mu_j \leq \bigwedge_{h \in H'_j} \xi_h \leq \bigwedge_{h \in H'_j} (\bigwedge_{i \in I_h} \sigma_i) & \text{by } \textit{(4.20)} \text{ and } \textit{(4.21)} \\ \bigwedge_{h \in H'_j} (\bigwedge_{i \in I_h} \tau_i) \leq \bigwedge_{h \in H'_j} \rho_h \leq \nu_j & \text{by } \textit{(4.21)} \text{ and } \textit{(4.20)} \end{array}$$

so that we set $I_j = \bigcup_{h \in H'_j} I_h$. □

Property **(P2)** explicitly relates $\leq$ with the ordering in the powerdomain constructed in Section 2. More precisely the ordering $\leq$ on types is dual to the preorder $\preceq$ on $M(D)$, when we interpret $\oplus$ as union operator ($\uplus$) and $\wedge$ as join ($\tilde{\sqcup}$). In fact, in Section 4.6 we will see that the quotient of types under $=$ gives the dual of a op-lattice.

Property **(P4)** assures the representability of continuous functions. Let us assume that σ, τ, μ, ν correspond to the domain elements s, t, m, n (possibly with indexes). If we regard $\sigma \rightarrow \tau$ as corresponding to the step function $s \Rightarrow t$, we can rewrite **(P4)** as the following standard implication:

$$\bigsqcup_{i \in I} (s_i \Rightarrow t_i) \ \preceq \ \bigsqcup_{j \in J} (m_j \Rightarrow n_j) \text{ if and only if}$$

$$\forall i \in I. \, \exists J' \subseteq J. \ \bigsqcup_{j \in J'} m_j \ \preceq \ s_i \ \& \ t_i \ \preceq \ \bigsqcup_{j \in J'} n_j,$$

which implies the representability of continuous functions as joins of step functions as usual (see also the proof of 4.3.14 and [24, 27]).

4.3.2. Remark. The hypothesis $\nu_j \neq \omega$ cannot be dropped in **(P4)**. Indeed, by 4.3.1*(ii)* and *(xii)* we have, for any σ,

$$\sigma \rightarrow \omega = \omega \rightarrow \omega.$$

This implies that, for any choice of μ, ν and σ,

$$\mu \rightarrow \nu \leq \omega \rightarrow \omega = \sigma \rightarrow \omega.$$

So that **(P4)** would be false as soon as $\sigma \not\leq \mu$.

4.3.11. Lemma. *Suppose that* $\rho \wedge \xi \leq \bigoplus_{i \in I} (\sigma \rightarrow \tau_i)$, *then:*

(P5) *if* $\rho, \xi \leq \omega_1$ *then there exist* $\{\mu_j \mid j \in J\}$ *and* $\{\nu_l \mid l \in L\}$ *such that*

$$\rho \leq \bigoplus_{j \in J} (\sigma \rightarrow \mu_j), \quad \xi \leq \bigoplus_{l \in L} (\sigma \rightarrow \nu_l), \quad \textit{and} \quad (\bigoplus_{j \in J} \mu_j) \wedge (\bigoplus_{l \in L} \nu_l) \leq \bigoplus_{i \in I} \tau_i;$$

(P6) *if* $\rho \leq \omega_1$ *and* $\xi = \xi \oplus \omega \neq \omega$ *then there exist* $\{\mu_j \mid j \in J\}$ *and* $\{\nu_l \mid l \in L\}$ *such that*

$$\rho \leq \bigoplus_{j \in J} (\sigma \rightarrow \mu_j), \ \xi \leq \bigoplus_{l \in L} (\sigma \rightarrow \nu_l) \oplus \omega, \ \textit{and} \ (\bigoplus_{j \in J} \mu_j) \wedge (\bigoplus_{l \in L} \nu_l \oplus \omega) \leq \bigoplus_{i \in I} \tau_i.$$

If instead $\rho \wedge \xi \leq \bigoplus_{i \in I} (\sigma \rightarrow \tau_i) \oplus \omega$, *then:*

(P7) *if $\rho, \xi \neq \omega$ then there exist $\{\mu_j \mid j \in J\}$ and $\{\nu_l \mid l \in L\}$ such that*

$$\rho \leq \bigoplus_{j\in J}(\sigma \to \mu_j) \oplus \omega,\ \xi \leq \bigoplus_{l\in L}(\sigma \to \nu_l) \oplus \omega \text{ and}$$

$$[(\bigoplus_{j\in J}\mu_j) \wedge (\bigoplus_{l\in L}\nu_l)] \oplus \omega \leq \bigoplus_{i\in I}\tau_i \oplus \omega.$$

Proof.

(P5) Let $\Theta(\rho) = \{\rho_h \mid h \in H\}$ and $\Theta(\xi) = \{\xi_k \mid k \in K\}$. As $\rho, \xi \leq \omega_1$, $\omega \notin \Theta(\rho) \cup \Theta(\xi)$ by 4.3.9*(iii)*. Therefore $\Theta(\rho\wedge\xi) = \{\rho_h \wedge \xi_k \mid h \in H, k \in K\}$. Moreover let $\rho_h = \bigwedge_{r\in R_h}(\alpha_{h,r} \to \beta_{h,r})$ and $\xi_k = \bigwedge_{s\in S_k}(\gamma_{k,s} \to \delta_{k,s})$. Define $R'_h = \{r \mid \sigma \leq \alpha_{h,r}\}$ and $S'_k = \{s \mid \sigma \leq \gamma_{k,s}\}$. Clearly $R'_h \subseteq R_h$, $S'_k \subseteq S_k$ and some of them may be empty. We set intersections indexed by the empty set to ω.

$$\begin{array}{lll} \rho \wedge \xi \leq \bigoplus_{i\in I}(\sigma \to \tau_i) & \Rightarrow & \forall h \in H, k \in K.\ \exists f(h,k) \in I.\rho_h \wedge \xi_k \leq \sigma \to \tau_{f(h,k)} \text{ by } \mathbf{(P2)} \\ & \Rightarrow & \forall h \in H, k \in K.\ \exists f(h,k) \in I. \\ & & \bigwedge_{r\in R'_h}\beta_{h,r} \wedge \bigwedge_{s\in S'_k}\delta_{k,s} \leq \tau_{f(h,k)} \qquad \text{by } \mathbf{(P4)}. \end{array} \tag{4.22}$$

$$\begin{array}{lll} \rho \wedge \xi \leq \bigoplus_{i\in I}(\sigma \to \tau_i) & \Rightarrow & \bigwedge_{h\in H,k\in K}(\rho_h \wedge \xi_k) \leq \bigwedge_{i\in I}(\sigma \to \tau_i) \qquad \text{by } \mathbf{(P2)} \\ & \Rightarrow & \forall i \in I.\ \bigwedge_{h\in H,k\in K}(\rho_h \wedge \xi_k) \leq \sigma \to \tau_i \\ & \Rightarrow & \forall i \in I.\ \bigwedge_{h\in H,k\in K}(\bigwedge_{r\in R'_h}\beta_{h,r} \wedge \bigwedge_{s\in S'_k}\delta_{k,s}) \leq \tau_i \quad \text{by } \mathbf{(P4)} \\ & \Rightarrow & \bigwedge_{h\in H,k\in K}(\bigwedge_{r\in R'_h}\beta_{h,r} \wedge \bigwedge_{s\in S'_k}\delta_{k,s}) \leq \bigoplus_{i\in I}\tau_i. \end{array} \tag{4.23}$$

Therefore we can put $J = H$, $L = K$, $\mu_j \equiv \bigwedge_{r\in R'_j}\beta_{j,r}$, and $\nu_l \equiv \bigwedge_{s\in S'_l}\delta_{l,s}$, where $j \in J$ and $l \in L$ (always equating intersections on the empty set to ω).
Now (4.22) and (4.23) can be rewritten respectively as:

$$\forall j \in J, l \in L.\ \exists f(j,l) \in I.\mu_j \wedge \nu_l \leq \tau_{f(j,l)}, \tag{4.24}$$

$$\bigwedge_{j\in J,l\in L}(\mu_j \wedge \nu_l) \leq \bigoplus_{i\in I}\tau_i. \tag{4.25}$$

It is easy to verify that $\rho_j \leq \sigma \to \mu_j$ and $\xi_l \leq \sigma \to \nu_l$. This implies $\rho \leq \bigoplus_{j\in J}(\sigma \to \mu_j)$ and $\xi \leq \bigoplus_{l\in L}(\sigma \to \nu_l)$. Moreover we have:

$$\begin{array}{lll} (\bigoplus_{j\in J}\mu_j) \wedge (\bigoplus_{l\in L}\nu_l) & = & \bigoplus_{j\in J,l\in L}(\mu_j \wedge \nu_l) \\ & = & \bigoplus_{j\in J,l\in L}(\mu_j \wedge \nu_l) \oplus (\bigwedge_{j\in J,l\in L}(\mu_j \wedge \nu_l)) \quad \text{by } \mathbf{(P0)} \\ & \leq & \bigoplus_{j\in J,l\in L}\tau_{f(j,l)} \oplus \bigoplus_{i\in I}\tau_i \qquad \text{by (4.24) and (4.25)} \\ & = & \bigoplus_{i\in I}\tau_i. \end{array}$$

(P6) By 4.3.8*(i)* and **(P3)** $\xi = (\bigwedge_{\xi'\in\Theta(\xi)}\xi') \oplus \omega$, which implies $\rho \wedge \xi = (\rho \wedge \bigwedge_{\xi'\in\Theta(\xi)}\xi') \oplus \rho$. Let $\bigwedge_{\xi'\in\Theta(\xi)}\xi' = \bigwedge_{k\in K}(\gamma_k \to \delta_k)$, $K' = \{k \mid \sigma \leq \gamma_k\}$, and ρ_h, R'_h be as in the proof of **(P5)**.

$$\begin{array}{lll} \rho \wedge \xi \leq \bigoplus_{i\in I}(\sigma \to \tau_i) & \Rightarrow & \forall h \in H.\exists g(h) \in I.\ \rho_h \leq \sigma \to \tau_{g(h)} \qquad \text{by } \mathbf{(P2)} \\ & \Rightarrow & \forall h \in H.\exists g(h) \in I.\ \bigwedge_{r\in R'_h}\beta_{h,r} \leq \tau_{g(h)} \quad \text{by } \mathbf{(P4)} \end{array} \tag{4.26}$$

and

$$\bigwedge_{h\in H}(\bigwedge_{r\in R'_h}\beta_{h,r})\wedge\bigwedge_{k\in K'}\delta_k\leq\bigoplus_{i\in I}\tau_i. \tag{4.27}$$

We choose J,μ_j,L as in the proof of **(P5)**, $\nu_l=\delta_l$ if $l\in K'$, $\nu_l=\omega$ otherwise; then (4.26) and (4.27) can be rewritten as follows:

$$\forall j\in J.\exists g(j)\in I.\ \mu_j\leq\tau_{g(j)}, \tag{4.28}$$

$$\bigwedge_{j\in J,l\in L}(\mu_j\wedge\nu_l)\leq\bigoplus_{i\in I}\tau_i. \tag{4.29}$$

It is easy to verify that $\rho\leq\bigoplus_{j\in J}(\sigma\to\mu_j)$ and $\xi\leq\bigoplus_{l\in L}(\sigma\to\nu_l)\oplus\omega$. We conclude observing that:

$$\begin{aligned}(\bigoplus_{j\in J}\mu_j)\wedge[\bigoplus_{l\in L}\nu_l\oplus\omega] &= [\bigoplus_{j\in J,l\in L}(\mu_j\wedge\nu_l)]\oplus[\bigoplus_{j\in J}\mu_j]\\ &= [\bigoplus_{j\in J,l\in L}(\mu_j\wedge\nu_l)]\oplus[\bigoplus_{j\in J}\mu_j]\oplus[\bigwedge_{j\in J,l\in L}(\mu_j\wedge\nu_l)]\\ &\qquad\qquad\text{by } \mathbf{(P0)}\\ &\leq \bigoplus_{j\in J}\tau_{g(j)}\oplus\bigoplus_{j\in J}\tau_{g(j)}\oplus\bigoplus_{i\in I}\tau_i\\ &\qquad\qquad\text{by (4.28), (4.29)}\\ &= \bigoplus_{i\in I}\tau_i.\end{aligned}$$

(P7) Notice that $\rho\wedge\xi\leq\bigoplus_{i\in I}(\sigma\to\tau_i)\oplus\omega$ implies $(\rho\wedge\xi)\oplus\omega\leq\bigoplus_{i\in I}(\sigma\to\tau_i)\oplus\omega$. Now, let $\bigwedge_{\rho'\in\Theta(\rho)}\rho'=\bigwedge_{h\in H}(\alpha_h\to\beta_h)$, and $\bigwedge_{\xi\in\Theta(\xi)}\xi'=\bigwedge_{k\in K}(\gamma_k\to\delta_k)$. By **(P3)** and **(P2)**

$$(\rho\wedge\xi)\oplus\omega\leq\bigoplus_{i\in I}(\sigma\to\tau_i)\oplus\omega\Rightarrow\bigwedge_{h\in H}(\alpha_h\to\beta_h)\wedge\bigwedge_{k\in K}(\gamma_k\to\delta_k)\leq\bigwedge_{i\in I}(\sigma\to\tau_i).$$

Let again $H'=\{h\mid\sigma\leq\alpha_h\}$, $K'=\{k\mid\sigma\leq\gamma_k\}$. We get, by **(P4)**,

$$\bigwedge_{h\in H}(\alpha_h\to\beta_h)\wedge\bigwedge_{,k\in K}(\gamma_k\to\delta_k)\leq\bigwedge_{i\in I}(\sigma\to\tau_i)\Rightarrow$$

$$\bigwedge_{h\in H'}\beta_h\wedge\bigwedge_{k\in K'}\delta_k\leq\bigwedge_{i\in I}\tau_i.$$

Therefore we can choose $J=H$, $L=K$, $\mu_j=\beta_j$ if $j\in H'$, $\mu_j=\omega$ otherwise, $\nu_l=\delta_l$ if $l\in K'$, $\nu_l=\omega$ otherwise. In fact we have:

$$\rho\leq\bigwedge_{j\in J}(\sigma\to\mu_j)\oplus\omega,\ \xi\leq\bigwedge_{l\in L}(\sigma\to\nu_l)\oplus\omega.$$

Finally

$$[(\bigoplus_{j\in J}\mu_j)\wedge(\bigoplus_{l\in L}\nu_l)]\oplus\omega\leq\bigoplus_{i\in I}\tau_i\oplus\omega,$$

since $\bigwedge_{j\in J}\mu_j\wedge\bigwedge_{l\in L}\nu_l\leq\bigwedge_{i\in I}\tau_i$. □

4.3.2. Isomorphism between D° and $\mathcal{F}$

We claim that the set $\mathcal{F}$ of filters over *Type*, ordered by subset inclusion, is isomorphic to D° as $\mathcal{P}$-algebra. We first show that they are isomorphic as ω-algebraic lattices.

4.3.12. PROPOSITION. The poset $(\mathcal{F}, \subseteq)$ is an ω-algebraic lattice where the meet is the intersection and the join is the least filter including the union, i.e. $F \mathbin{\bar{\cup}} F' = \uparrow \{\sigma \wedge \tau \mid \sigma \in F, \tau \in F'\}$. The compact elements of $(\mathcal{F}, \subseteq)$ are the principal filters $\uparrow\sigma =^{Df} \{\tau \in Type \mid \sigma \leq \tau\}$.

Proof. Standard. □

By this proposition the isomorphism $\mathcal{F} \simeq D^\circ$ inside the category **ALG** is established if there is a one-to-one order preserving map between compact elements of these domains. This is equivalent to have a surjective map $(\ )^+ : Type \to K(D)$ such that $\sigma \leq \tau$ if and only if $(\tau)^+ \subseteq (\sigma)^+$, for all types σ, τ.

4.3.13. DEFINITION. *(i)* The map $(\ )^+ : Type \to K(D^\circ)$ is defined by:

$$\begin{array}{rcl}(\omega)^+ & = & \bot; \\ (\sigma \to \tau)^+ & = & \{\!|(\sigma)^+ \Rightarrow (\tau)^+|\!\}; \\ (\sigma \wedge \tau)^+ & = & (\sigma)^+ \vee (\tau)^+; \\ (\sigma \oplus \tau)^+ & = & (\sigma)^+ \uplus (\tau)^+.\end{array}$$

(ii) $\Phi : \mathcal{F} \to D^\circ$ is the unique continuous extension of the map $(\ )^+$, that is:

$$\Phi(F) = \bigsqcup\{(\sigma)^+ \mid \sigma \in F\},$$

The map is well defined by Lemma 4.2.8 and by the fact that the join of compact elements is compact.

4.3.14. THEOREM. *The map $(\ )^+$ is surjective and such that $(\sigma)^+ \subseteq (\tau)^+$ if and only if $\tau \leq \sigma$. Therefore Φ is an isomorphism in the category* **ALG**.

Proof. To see that $(\ )^+$ is surjective we show that for all n, $K(D_n) \subseteq (Type)^+$ by induction on n.

In fact $\{(\omega)^+\} = K(D_0) = \{\bot\}$ which establishes the thesis when $n = 0$.

Let $e = \{\!|f_1, \ldots, f_m|\!\} \in K(D_{n+1})$. Then for all $i \leq m$, either $f_i = \bigsqcup_{j \in J_i}(e_j^{(i)} \Rightarrow d_j^{(i)})$, where J_i is a finite set of indexes and $e_j^{(i)}, d_j^{(i)} \in K(D_n)$ for all $j \in J_i$, or $f_i = \bot$. In the latter case, by the idempotency of $\uplus$, we can freely suppose that there is just one $i \leq m$ such that $f_i = \bot$. By induction there are types $\mu_j^{(i)}, \nu_j^{(i)}$ such that $(\mu_j^{(i)})^+ = e_j^{(i)}$ and $(\nu_j^{(i)})^+ = d_j^{(i)}$. Then, if $f_i \neq \bot$ for all i, we have

$$e = \left(\bigoplus_{i \leq m} \bigwedge_{j \in J_i} \mu_j^{(i)} \to \nu_j^{(i)}\right)^+,$$

otherwise let $f_m = \bot$, then

$$e = \left(\left(\bigoplus_{i \leq m-1} \bigwedge_{j \in J_i} \mu_j^{(i)} \to \nu_j^{(i)}\right) \oplus \omega\right)^+.$$

We are left to show that

$$\sigma \leq \tau \ \Leftrightarrow \ (\tau)^+ \sqsubseteq (\sigma)^+.$$

The $(\Rightarrow)$ part can be easily proved by induction on $\leq$. The proof of the $(\Leftarrow)$ part is by structural induction on σ and τ. The more interesting case is when

$$\sigma = \bigoplus_{i \in I} \bigwedge_{h \in H_i} (\mu_h^{(i)} \to \nu_h^{(i)}) \quad \text{and} \quad \tau = \bigoplus_{j \in J} \bigwedge_{k \in K_j} (\rho_k^{(j)} \to \xi_k^{(j)}),$$

for suitable finite sets of indexes and types. Then, if $(\mu_h^{(i)})^+ = e_h^{(i)}$, $(\nu_h^{(i)})^+ = d_h^{(i)}$, $(\rho_k^{(j)})^+ = r_k^{(j)}$, and $(\xi_k^{(j)})^+ = s_k^{(j)}$, we have:

$$(\sigma)^+ = \{\!|\bigsqcup_{h\in H_i}(e_h^{(i)} \Rightarrow d_h^{(i)}) \mid i \in I|\!\} \quad \text{and} \quad (\tau)^+ = \{\!|\bigsqcup_{k\in K_j}(r_k^{(j)} \Rightarrow s_k^{(j)}) \mid j \in J|\!\}.$$

Now $\{\!|\bigsqcup_{k\in K_j}(r_k^{(j)} \Rightarrow s_k^{(j)}) \mid j \in J|\!\} \subseteq \{\!|\bigsqcup_{h\in H_i}(e_h^{(i)} \Rightarrow d_h^{(i)}) \mid i \in I|\!\}$ if an only if:

$$\bigsqcup_{j\in J}[\bigsqcup_{k\in K_j}(r_k^{(j)} \Rightarrow s_k^{(j)})] \sqsubseteq \bigsqcup_{i\in I}[\bigsqcup_{h\in H_i}(e_h^{(i)} \Rightarrow d_h^{(i)})] \tag{4.30}$$

$$\forall i \in I.\exists j \in J \bigsqcup_{k\in K_j}(r_k^{(j)} \Rightarrow s_k^{(j)}) \sqsubseteq \bigsqcup_{h\in H_i}(e_h^{(i)} \Rightarrow d_h^{(i)}). \tag{4.31}$$

Inequation (4.30) implies that

$$\forall j \in J.\forall k \in K_j.\forall i \in I.\exists H_i' \subseteq H_i.\ \bigsqcup_{i\in I}(\bigsqcup_{h\in H_i'} e_h^{(i)}) \sqsubseteq r_k^{(j)} \ \&\ s_k^{(j)} \sqsubseteq \bigsqcup_{i\in I}(\bigsqcup_{h\in H_i'} d_h^{(i)}).$$

Notice that some H_i' may be empty, in which case the corresponding sup is equated to $\bot$. By induction we have

$$\forall j \in J.\forall k \in K_j.\forall i \in I.\exists H_i' \subseteq H_i.\ \rho_k^{(j)} \leq \bigwedge_{i\in I}(\bigwedge_{h\in H_i'} \mu_h^{(i)}) \ \&\ \bigwedge_{i\in I}(\bigwedge_{h\in H_i'} \nu_h^{(i)}) \leq \xi_k^{(j)},$$

therefore by **(P4)** we get

$$\bigwedge_{i\in I}[\bigwedge_{h\in H_i}(\mu_h^{(i)} \to \nu_h^{(i)})] \leq \bigwedge_{j\in J}[\bigwedge_{k\in K_j}(\rho_k^{(j)} \to \xi_k^{(j)})]. \tag{4.32}$$

Inequation (4.31) implies that

$$\forall i \in I.\exists j \in J.\forall k \in K_j.\exists H_i' \subseteq H_i.\ \bigsqcup_{h\in H_i'} e_h^{(i)} \sqsubseteq r_k^{(j)} \ \&\ s_k^{(j)} \sqsubseteq \bigsqcup_{h\in H_i'} d_h^{(i)}.$$

By induction we have

$$\forall i \in I.\exists j \in J.\forall k \in K_j\ \exists H_i' \subseteq H_i.\ \rho_k^{(j)} \leq \bigwedge_{h\in H_i'} \mu_h^{(i)} \ \&\ \bigwedge_{h\in H_i'} \nu_h^{(i)} \leq \xi_k^{(j)},$$

therefore by **(P4)** it follows that

$$\forall i \in I\ \exists j \in J.\ \bigwedge_{h\in H_i}(\mu_h^{(i)} \to \nu_h^{(i)}) \leq \bigwedge_{k\in K_j}(\rho_k^{(j)} \to \xi_k^{(j)}). \tag{4.33}$$

Now (4.32) and (4.33) imply, by **(P2)**, that

$$\bigoplus_{i\in I}\left(\bigwedge_{h\in H_i}(\mu_h^{(i)} \to \nu_h^{(i)})\right) \leq \left(\bigoplus_{j\in J}\bigwedge_{k\in K_j}(\rho_k^{(j)} \to \xi_k^{(j)})\right),$$

that is $\sigma \leq \tau$. □

The isomorphism Φ and the union operation $\uplus$ over D° induce on $\mathcal{F}$ a continuous binary map $* : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$. Indeed, since $\Phi(\uparrow\sigma) = (\sigma)^+$, we immediately have from Definition 4.3.13

$$\Phi(\uparrow\sigma) \uplus \Phi(\uparrow\tau) = (\sigma)^+ \uplus (\tau)^+ = (\sigma \oplus \tau)^+ = \Phi(\uparrow(\sigma \oplus \tau)).$$

This extends by continuity to

$$F * F' = \Phi^{-1}(\Phi(F) \uplus \Phi(F')) = \uparrow \{\sigma \oplus \tau \mid \sigma \in F \;\&\; \tau \in F'\}.$$

Therefore $\mathcal{F}$ is turned into a $\mathcal{P}$-algebra by Φ and we conclude:

4.3.15. Corollary. *Let $* : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$ be defined by:*

$$F * F' = \uparrow \{\sigma \oplus \tau \mid \sigma \in F \;\&\; \tau \in F'\}.$$

Then the structures $(\mathcal{F}, *, \subseteq)$ and $(D^\circ, \uplus, \sqsubseteq)$ are isomorphic in $\mathcal{A}$.

We say that a filter F is $\oplus$-*complete* if and only if $\sigma \oplus \tau \in F$ implies $\sigma \wedge \tau \in F$. Let $\mathcal{F}_C$ be the set of $\oplus$-complete filters. Clearly $\oplus$-complete filters are related to $\uplus$-irreducible elements of D°. Define

$$C^\circ(D^\circ) = \Psi^{-1}(\mathcal{C}(\mathcal{P}([D^\circ \to D^\circ]_\perp))),$$

where Ψ and $\mathcal{C}(\;)$ are respectively defined in 4.2.7 and 4.2.4*(ii)*.

4.3.16. Proposition. $C^\circ(D^\circ) = \Phi(\mathcal{F}_C)$, therefore $\mathcal{F}_C$ is the set of "singletons" in $\mathcal{F}$.

Proof. It is straightforward to show that

$$G \in \mathcal{F}_C \;\;\Leftrightarrow\;\; \forall F, F' \in \mathcal{F}.\; F * F' \subseteq G \Rightarrow F \,\bar{\cup} F' \subseteq G.$$

Now the statement follows from Definition 4.2.4*(i)* , since Φ, Ψ are isomorphisms. □

As a consequence, we immediately have that $\Phi^{-1}(C^\circ(D^\circ) \cap K(D^\circ))$ is the set of the principal filters $\uparrow\sigma$ such that $\sigma \in IType$. Indeed, if $\mu \oplus \nu \in \uparrow\sigma$ and σ is irreducible, then

$$\sigma = \sigma \wedge (\mu \oplus \tau) = (\sigma \wedge \mu) \oplus (\sigma \wedge \nu),$$

which implies, by irreducibility, $\sigma = \sigma \wedge \mu = \sigma \wedge \nu$, that is $\sigma \leq \mu \wedge \nu$.

Lastly we examine the application between filters induced by the application in D°.

4.3.17. Lemma. *(i) The functional application over D° induces, via the isomorphism, an application operation over $\mathcal{F}$. More precisely, for all $F, F' \in \mathcal{F}$, if $F \cdot F'$ is defined as $\Phi^{-1}(\Phi(F) \cdot \Phi(F'))$ then we have:*

$$\begin{array}{rcl} F \cdot F' & = & \left\{\bigoplus_{i \in I} \tau_i \mid \exists \sigma \in F'.\; \bigoplus_{i \in I}(\sigma \to \tau_i) \in F\right\} \cup \\ & & \cup \left\{\bigoplus_{i \in I} \tau_i \oplus \omega \mid \exists \sigma \in F'.\; \bigoplus_{i \in I}(\sigma \to \tau_i) \oplus \omega \in F\right\} \;\cup \uparrow\omega. \end{array}$$

Moreover the application distributes to the left over the "union" operation $*$, that is for any $F, F', G \in \mathcal{F}$, we have:

$$(F * F') \cdot G = (F \cdot G) * (F' \cdot G).$$

(ii) Let $\coprod : D^\circ \to D^\circ$ *be defined as in 4.2.9. If* $\coprod(F) = \Phi^{-1}(\coprod \Phi(F))$ *is the induced operation over* $\mathcal{F}$*, then*

$$\coprod(F) = \uparrow\{\sigma \wedge \tau \mid \sigma \oplus \tau \in F\}.$$

Proof.

(i) Once the main formula has been established, the left distributivity is shown by straightforward calculations.

To prove the main formula it suffices to consider F and F' to be principal filters. Let $\Theta(\sigma) = \{\sigma_1, \ldots, \sigma_m\}$. We consider two cases, according to whether $\Theta(\sigma)$ contains ω or not. If $\omega \notin \Theta(\sigma)$, then by 4.3.8*(iii)* and 4.3.8*(vii)* $\sigma_i = \bigwedge_{j \in J_i}(\mu_j^{(i)} \to \nu_i^{(i)})$ for some $J_i, \mu_j^{(i)}, \nu_i^{(i)}$. Therefore $\Phi(\uparrow\sigma) = \{\!|f_1, \ldots, f_m|\!\}$, where $f_i = (\sigma_i)^+ = (\bigwedge_{j \in J_i}(\mu_j^{(i)} \to \nu_i^{(i)}))^+$.
Let $\Phi(\uparrow\tau) = d$, $\Phi(\uparrow\mu_j^{(i)}) = e_j^{(i)}$, $\Phi(\uparrow\nu_j^{(i)}) = g_j^{(i)}$.

$$\begin{array}{rcl}
\Phi(\uparrow\sigma) \cdot \Phi(\uparrow\tau) & = & \left(\biguplus_{i \le m} \Phi(\uparrow\sigma_i)\right) \cdot \Phi(\uparrow\tau) \\
& = & \biguplus_{i \le m} \{\!|f_i(d)|\!\} \\
& = & \biguplus_{i \le m} \bigsqcup_{j \in J_i'} g_j^{(i)} \\
& = & \biguplus_{i \le m} \Phi(\bigwedge_{j \in J_i'} \nu_j^{(i)}) \\
& = & \Phi(\uparrow \bigoplus_{i \le m}(\bigwedge_{j \in J_i'} \nu_j^{(i)}))
\end{array}$$

where $J_i' = \{j \in J_i \mid e_j^{(i)} \sqsubseteq d\} = \{j \in J_i \mid \tau \le \mu_j^{(i)}\}$.
Define $\mu_i = \bigwedge_{j \in J_i'} \mu_j^{(i)}$, $\nu_i = \bigwedge_{j \in J_i'} \nu_j^{(i)}$ and $\mu = \bigwedge_{i \le m} \mu_i$. Then

$$\begin{array}{rcl}
\sigma & = & \bigoplus_{i \le m}(\bigwedge_{j \in J_i}(\mu_j^{(i)} \to \nu_j^{(i)})) \\
& \le & \bigoplus_{i \le m}(\bigwedge_{j \in J_i'}(\mu_j^{(i)} \to \nu_j^{(i)})) \\
& \le & \bigoplus_{i \le m}(\bigwedge_{j \in J_i'}(\mu \to \nu_j^{(i)})) \\
& = & \bigoplus_{i \le m}(\mu \to \nu_i).
\end{array}$$

Hence we conclude

$$\uparrow\sigma \cdot \uparrow\tau = \uparrow\left\{\bigoplus_{i \le m} \nu_i \mid \exists \mu \in \uparrow\tau.\ \bigoplus_{i \le m}(\mu \to \nu_i) \in \uparrow\sigma\right\}.$$

If $\omega \in \Theta(\sigma)$, say $\sigma_m = \omega$, then $\Phi(\uparrow\sigma) = \{\!|f_1, \ldots, f_{m-1}, \bot|\!\}$ and $\uparrow\sigma \cdot \uparrow\tau$ becomes

$$\uparrow\left\{\bigoplus_{i \le m} \nu_i \oplus \omega \mid \exists \mu \in \uparrow\tau.\ \bigoplus_{i \le m}(\mu \to \nu_i) \oplus \omega \in \uparrow\sigma\right\}.$$

Finally, if $\sigma = \omega$, then $\Phi(\uparrow\sigma) = \bot = \{\!|\bot|\!\}$ so that $\uparrow\sigma \cdot \uparrow\tau$ has to be $\uparrow\omega$. Since in any other case ω is in the set $\uparrow\sigma \cdot \uparrow\tau$, we conclude that in the general case the formula stated in this Lemma is valid.

(ii) Easy. $\square$

4.4. A Calculus of Multivalued Functions

In this section we introduce a calculus expressing parallel and non-deterministic constructors on higher-order functions. As explained in the Introduction, the semantic operators are the guidelines for this definition. We extend the syntax of λ-calculus with a parallel operator $\|$, a non-deterministic choice operator $+$, and a may-must operator $\coprod$. The operator $\|$ corresponds to the semantic join and therefore to the intersection type constructor. The operator $+$ corresponds to the semantic operator $*$ and therefore to the type constructor $\oplus$. Finally the unary operator $\coprod$ corresponds to the semantic operator which applied to a set gives the join of its elements.

We use two sorts of variables, namely the set Vn of call-by-name variables ranged over by x and the set Vv of call-by-value variables ranged over by v. The symbol χ will range over the set $\mathsf{Vn} \cup \mathsf{Vv}$.

$$M ::= x \mid v \mid \lambda\chi.M \mid (MM) \mid (M\|M) \mid (M+M) \mid (\coprod M).$$

We call $\Lambda_{+\|}$ the set of terms; $\Lambda^0_{+\|}$ is the set of closed terms. For any $M \in \Lambda_{+\|}$, we denote the set of free variables of M by $FV(M)$.

We shall write $M \equiv N$ for syntactical equality up to α-congruence, that is up to renaming of bound variables (of course any bound variable will be renamed by a variable of the same sort). By $M[N/\chi]$ is meant the result of substituting in M all free occurrences of χ by N, renaming bound variables of M that occur free in N.

We assume the following precedence between operators: application, abstraction, $\coprod$, $\|$, $+$.

4.4.1. Values and Reduction

The abstractions $\lambda x.M$ and $\lambda v.M$ express in our formalism non-strict and strict functions respectively, using a notion of value which was discussed in the introduction and is specified in the following definition.

4.4.1. Definition (Values). We define the sets $\mathbf{W} \subseteq \mathbf{U} \subseteq \mathbf{V} \subseteq \Lambda_{+\|}$ as follows:

(i) $\mathbf{W}$ is the set of terms generated by the following grammar:

$$W ::= v \mid \lambda x.M \mid \lambda v.M \mid W\|W$$

(ii) $\mathbf{U}$ is the set of terms generated by the following grammar:

$$U ::= W \mid U\|M \mid M\|U$$

(iii) $\mathbf{V}$ is the set of terms generated by the following grammar:

$$V ::= U \mid V+V.$$

$\mathbf{W}^0 = \mathbf{W} \cap \Lambda^0_{+\|}$, and similarly for $\mathbf{U}^0$ and $\mathbf{V}^0$.

Let us introduce the congruence relation $\approx$ which accounts for the idempotency, commutativity and associativity of $\|$ and $+$.

4.4.2. Definition. The binary relation $\approx$ is the minimal congruence over $\Lambda_{+\|}$ such that:

$$M+M \approx M \qquad M\|M \approx M$$
$$M+N \approx N+M \qquad M\|N \approx N\|M$$
$$M+(N+L) \approx (M+N)+L \qquad M\|(N\|L) \approx M\|(N\|L).$$

Since the reduction and conversion relations that we are going to define are up to $\approx$, we shall abstract away from the bracketing of sums or parallels, and we shall freely make use of the following abbreviations:

$$\sum_{i\in I} M_i \equiv M_1 + \cdots + M_n \qquad \text{and} \qquad \|_{i\in I} M_i \equiv M_1 \| \cdots \| M_n$$

where $I = \{1, \ldots, n\}$.

The following map extracts from a term in $\mathbf{U}$ the maximal subterm belonging to $\mathbf{W}$.

4.4.3. Definition. The map $m : \mathbf{U} \to \mathbf{W}$ is defined as follows

- $m(W) \equiv W$ if $W \in \mathbf{W}$;
- $m(U\|U') \equiv m(U)\|m(U')$ if $U, U' \in \mathbf{U}$;
- $m(U\|M) \equiv m(M\|U) \equiv m(U)$ if $U \in \mathbf{U}$ and $M \notin \mathbf{U}$.

4.4.1. Remark. $m(U)$ is either U itself, when $U \in \mathbf{W}$, or there exist $W \in \mathbf{W}$ and $M \notin \mathbf{U}$ such that $U \approx W\|M$ and $m(U) \equiv W$.

4.4.4. Definition (Reduction and Conversion Relations).

(i) The reduction relation $\longrightarrow$ is the least binary relation over $\Lambda^0_{+\|}$ such that:

$$(\beta)\quad (\lambda x.M)N \longrightarrow M[N/x] \qquad (\mu)\quad \frac{N \longrightarrow N'}{MN \longrightarrow MN'} \qquad (\nu)\quad \frac{M \longrightarrow M'}{MN \longrightarrow M'N}$$

$$(\beta_v)\quad \frac{W \in \mathbf{W}}{(\lambda v.M)W \longrightarrow M[W/v]} \qquad (\beta_v\|)\quad \frac{U \longrightarrow V \quad U \in \mathbf{U}}{(\lambda v.M)U \longrightarrow M[m(U)/v]\|(\lambda v.M)V}$$

$$(\beta_v+)\quad \frac{V \approx \sum_{i\in I} U_i \qquad \forall i \in I \quad U_i \in \mathbf{U}}{(\lambda v.M)V \longrightarrow \sum_{i\in I}((\lambda v.M)U_i) \;+\; (\lambda v.M)(\|_{i\in I} U_i)}$$

$$(\|_{app})\quad (M\|N)L \longrightarrow (ML)\|(NL) \qquad (+_{app})\quad (M+N)L \longrightarrow (ML)+(NL)$$

$$(+\|)\quad (M+N)\|L \longrightarrow (M\|L)+(N\|L) \qquad (\approx)\quad \frac{M \approx M' \qquad M' \longrightarrow N' \qquad N' \approx N}{M \longrightarrow N}$$

$$(\|_a)\quad \frac{M \longrightarrow M'}{M\|N \longrightarrow M'\|N} \qquad (+_a)\quad \frac{M \longrightarrow M'}{M+N \longrightarrow M'+N}$$

$$(\textstyle\coprod_+)\quad \coprod(M+N) \longrightarrow (\coprod M)\|(\coprod N) \qquad (\coprod_\|)\quad \coprod(M\|N) \longrightarrow (\coprod M)\|(\coprod N)$$

$$(\textstyle\coprod_{TV})\quad \frac{W \in \mathbf{W}}{\coprod W \longrightarrow W} \qquad (\coprod_{red})\quad \frac{M \longrightarrow N}{\coprod M \longrightarrow \coprod N.}$$

(ii) $\xrightarrow{*}$ is the reflexive and transitive closure of $\longrightarrow$, $=$ is the symmetric closure of $\xrightarrow{*}$.

4.4.2. Remark.

(i) $M \in \Lambda^0_{+\|}$ is a normal form with respect to $\longrightarrow$ if and only if $M \in \mathbf{W}$. Therefore (closed) values in $\mathbf{V} - \mathbf{W}$ are reducible. By inspection of the rules of reduction, it is easily seen that the set $\mathbf{V}$ is closed under $\longrightarrow$. For similar notions of reducible values (that is further refinable) in presence of parallel operators see e.g. [22] and chapter 3.

(ii) The relation $\longrightarrow$ is not Church-Rosser. A simple counter-example is

$$\lambda y.(\lambda x.x)(\lambda x.x) \longleftarrow (\lambda xy.x)((\lambda x.x)(\lambda x.x)) \xrightarrow{*} \lambda y.(\lambda x.x),$$

which is due to the presence of rule (μ) and the absence of rule (ξ) of the classical λ-calculus, allowing to reduce under abstraction. The problem, however, is not remedied by adding this last rule, as shown by a second counter-example. Let $\mathbf{K} \equiv \lambda xy.x$, $\mathbf{O} \equiv \lambda xy.y$ and $\mathbf{I} \equiv \lambda x.x$, then

$$\begin{array}{rcll} (\lambda vx.v)(\mathbf{K}\|(\mathbf{I}+\mathbf{O})) & \longrightarrow & (\lambda x.\mathbf{K})\|(\lambda vx.v)(\mathbf{K}\|\mathbf{I}+\mathbf{K}\|\mathbf{O}) & \text{by } (\beta_v\|) \\ & \xrightarrow{*} & (\lambda x.\mathbf{K})\|\lambda x.(\mathbf{K}\|\mathbf{I}) + (\lambda x.\mathbf{K})\|\lambda x.(\mathbf{K}\|\mathbf{O}) & \\ & & +(\lambda x.\mathbf{K})\|\lambda x.(\mathbf{K}\|\mathbf{I}\|\mathbf{O}) & \end{array}$$

but also

$$\begin{array}{rcll} (\lambda vx.v)(\mathbf{K}\|(\mathbf{I}+\mathbf{O})) & \longrightarrow & (\lambda vx.v)(\mathbf{K}\|\mathbf{I}+\mathbf{K}\|\mathbf{O}) & \text{by } (\mu) \text{ and } (+\|) \\ & \xrightarrow{*} & \lambda x.(\mathbf{K}\|\mathbf{I}) + \lambda x.(\mathbf{K}\|\mathbf{O}) + \lambda x.(\mathbf{K}\|\mathbf{I}\|\mathbf{O}). & \end{array}$$

4.4.2. May and Must Convergence

To compare the operational semantics with the denotational semantics and hence to check whether the expressibility criterion has been satisfied or not, we introduce a (pre)-congruence among terms which is based on the notion of convergence that the previous definition of $\longrightarrow$ naturally induces. Let the predicate $M \Downarrow^{must}$, to be read as "M must converge", be as follows:

$$M \Downarrow^{must} \;\Leftrightarrow\; \exists V \in \mathbf{V}.\; M \xrightarrow{*} V. \tag{4.34}$$

At a first glance this looks like a *may* convergence predicate. Indeed "*must* converge" should be reserved to those terms such that any reduction out of them reaches a value. However, a notion of must convergence related to considering all possible reductions of a term is problematic, since, as it has been shown in Remark 4.4.2, our reduction relation is not Church-Rosser.

However there are two relevant facts: first we shall prove that, if $M \Downarrow^{must}$ and $M \longrightarrow N$, then $N \Downarrow^{must}$, which is a strong property especially because of lack of confluence. Second our values are sets, representing all possible outputs of a non-deterministic computation; but they are also, as values, total objects, hence the convergence notion they induce is a must convergence predicate.

At the same time we introduce a *may* convergence predicate, which is motivated by the possible choice of $\mathcal{V}^\flat$ as the set of values. Its definition is (without changing the definition of syntactical values)

$$M \Downarrow^{may} \Leftrightarrow \exists V \in \mathbf{V}, N.\; M \xrightarrow{*} V + N.$$

Note that if $M \Downarrow^{must}$ then $M \Downarrow^{may}$: indeed if $M \xrightarrow{*} V$ for some $V \in \mathbf{V}$, then it is also the case that $M \xrightarrow{*} V + V$, thanks to $\approx$.

We shall study the theory induced by the predicate $\Downarrow^{must}$ for the same reason we choosed $\mathcal{V}^\sharp$ instead of $\mathcal{V}^\flat$ as our semantical notion of value. Indeed with the second choice union and join collapse; similarly, choosing $\Downarrow^{may}$ as our convergence predicate, $+$ and $\|$ collapse.

4.4.5. Definition (Convergence). Let $M \in \Lambda^0_{+\|}$ and $V \in \mathbf{V} \cap \Lambda^0_{+\|}$, then:

(i) $M\Downarrow^{must}$ if and only if $\exists V' \in \mathbf{V} \cap \Lambda^0_{+\|}.\ M \xrightarrow{*} V'$;
(ii) $M\Downarrow^{may} V$ if and only if $\exists N.\ M \xrightarrow{*} V + N$;
(iii) $M\Downarrow^{may}$ if and only if $M\Downarrow^{may} V'$ for some value $V' \in \mathbf{V} \cap \Lambda^0_{+\|}$;
(iv) $M\Uparrow^{must}$ if and only if $\neg M\Downarrow^{may}$;
(v) $M\Uparrow^{may}$ if and only if $\neg M\Downarrow^{must}$.

For the above definition to make sense we have to assure that the convergence predicates are preserved under reduction (and hence by conversion, since they are trivially preserved by expansion). As observed, the reduction relation is not confluent, so that the following is a non trivial result.

4.4.6. PROPOSITION. Let $\Downarrow$ be either $\Downarrow^{must}$ or $\Downarrow^{may}$. Then for any closed M and N:

$$M\Downarrow \quad \text{and} \quad M \longrightarrow N \quad \text{imply} \quad N\Downarrow\ .$$

We do not prove this Proposition here, since it is an immediate consequence of Theorem 4.5.6 and Corollaries 4.5.7, 4.7.7. Of course, to avoid circularity, Proposition 4.4.6 will not be used anywhere in the proofs of the subsequent statements.

4.4.3. REMARK.

(i) A typical divergent term, both with respect to $\Downarrow^{must}$ and to $\Downarrow^{may}$ is $\Omega \equiv \Delta\Delta$, where $\Delta \equiv \lambda x.xx$. The same is true for any term of the shape $\Omega M_1 \cdots M_n$ or $(\lambda v.N)\Omega M_1 \cdots M_n$.

(ii) Although $+$ represents the non-deterministic choice, the rule

$$(+) \qquad M + N \longrightarrow M,$$

which was considered in previous chapters, would invalidate Proposition 4.4.6. As an example consider $\Omega + \mathbf{I}$ and $\Downarrow^{may}$.

(iii) The classical call-by-value β-rule is unsound in the present context with respect to the $\Downarrow^{must}$ predicate. Let $M \equiv \lambda v.(v\Delta\mathbf{I}\Delta\|v\mathbf{I}\Delta\Delta)$, then

$$\begin{array}{rcll} M(\Delta\|(\mathbf{K}+\mathbf{O})) & \longrightarrow & M(\Delta\|\mathbf{K}+\Delta\|\mathbf{O}) & \text{by } (\mu) \text{ and } (+\|) \\ & \xrightarrow{*} & \Omega\|\Delta \in \mathbf{V} & \end{array}$$

but also, by an unconstrained (β_v) rule, since $\Delta\|(\mathbf{K}+\mathbf{O}) \in \mathbf{V}$,

$$\begin{array}{rcl} M(\Delta\|(\mathbf{K}+\mathbf{O})) & \longrightarrow & (\Delta\|(\mathbf{K}+\mathbf{O}))\Delta\mathbf{I}\Delta\|(\Delta\|(\mathbf{K}+\mathbf{O}))\mathbf{I}\Delta\Delta \\ & \xrightarrow{*} & \Omega\mathbf{I}\Delta\|\Omega + \Omega\mathbf{I}\Delta\|\Delta + \Omega\|\Delta \end{array}$$

which neither is a value nor is reducible to a value at all.

(iv) We cannot get rid of the mapping m in the rule $(\beta_v\|)$ because of the $\coprod$ operator and of the rule (μ). Indeed suppose that we have instead

$$(\beta_v\|)' \ \frac{U \longrightarrow V \quad U \in \mathbf{U}}{(\lambda v.M)U \longrightarrow M[U/v]\|(\lambda v.M)V}$$

as in chapter 3. Then consider the term $N \equiv \lambda v.((\coprod v)\mathbf{I}\Omega\Omega + (\coprod v)\Omega\mathbf{I}\Omega)$; take further $P \equiv \lambda xyz.x$, $Q \equiv \lambda xyz.y$ and $R \equiv \lambda xyx.z$, which are all in $\mathbf{W}$. Then

$$N((P\|R) + (Q\|R)) \longrightarrow N(P\|R) + N(Q\|R) + N(P\|Q\|R)$$

since $P\|R+Q\|R \in \mathbf{V}-\mathbf{U}$. Now

$$\begin{array}{rcl} N(P\|R) & \longrightarrow & (\coprod(P\|R))\mathbf{I}\Omega\Omega + (\coprod(P\|R))\Omega\mathbf{I}\Omega \\ & \xrightarrow{*} & (P\|R)\mathbf{I}\Omega\Omega + (P\|R)\Omega\mathbf{I}\Omega \\ & \xrightarrow{*} & (P\mathbf{I}\Omega\Omega\|R\mathbf{I}\Omega\Omega) + (P\Omega\mathbf{I}\Omega\|R\Omega\mathbf{I}\Omega) \\ & \xrightarrow{*} & (\mathbf{I}\|\Omega) + (\Omega\|\Omega) \approx (\mathbf{I}\|\Omega) + \Omega. \end{array}$$

Similarly one checks that $N(Q\|R) \xrightarrow{*} (\mathbf{I}\|\Omega)+\Omega$. Finally

$$\begin{array}{rcl} N(P\|Q\|R) & \longrightarrow & (\coprod(P\|Q\|R))\mathbf{I}\Omega\Omega + (\coprod(P\|Q\|R))\Omega\mathbf{I}\Omega \\ & \xrightarrow{*} & (P\|Q\|R)\mathbf{I}\Omega\Omega + (P\|Q\|R)\Omega\mathbf{I}\Omega, \end{array}$$

where

$$(P\|Q\|R)\mathbf{I}\Omega\Omega \xrightarrow{*} P\mathbf{I}\Omega\Omega\|Q\mathbf{I}\Omega\Omega\|R\mathbf{I}\Omega\Omega \xrightarrow{*} \mathbf{I}\|\Omega\|\Omega \approx \mathbf{I}\|\Omega$$

and

$$(P\|Q\|R)\Omega\mathbf{I}\Omega \xrightarrow{*} P\Omega\mathbf{I}\Omega\|Q\Omega\mathbf{I}\Omega\|R\Omega\mathbf{I}\Omega \xrightarrow{*} \Omega\|\mathbf{I}\|\Omega \approx \mathbf{I}\|\Omega,$$

so that we conclude

$$N(P\|R+Q\|R) \xrightarrow{*} \mathbf{I}\|\Omega+\Omega+\mathbf{I}\|\Omega+\Omega+\mathbf{I}\|\Omega \approx \mathbf{I}\|\Omega+\Omega.$$

On the other hand, using rule $(\beta_v\|)'$ and observing that $(P+Q)\|R \in \mathbf{U}-\mathbf{W}$, we have

$$\begin{array}{rcl} N((P+Q)\|R) & \longrightarrow & ((\coprod((P+Q)\|R))\mathbf{I}\Omega\Omega + (\coprod((P+Q)\|R))\Omega\mathbf{I}\Omega)\|N((P\|R)+(Q\|R)) \\ & \xrightarrow{*} & ((P\|Q\|R)\mathbf{I}\Omega\Omega + (P\|Q\|R)\Omega\mathbf{I}\Omega)\|N((P\|R)+(Q\|R)) \\ & \xrightarrow{*} & (\mathbf{I}\|\Omega+\mathbf{I}\|\Omega)\|(\mathbf{I}\|\Omega+\Omega) \approx \mathbf{I}\|\Omega\|(\mathbf{I}\|\Omega+\Omega) \\ & \longrightarrow & (\mathbf{I}\|\Omega)\|(\mathbf{I}\|\Omega)+(\mathbf{I}\|\Omega)\|\Omega \approx \mathbf{I}\|\Omega. \end{array}$$

To sum up, we have found that $N((P+Q)\|R)\Downarrow^{must}$ but $N((P\|R)+(Q\|R))\Uparrow^{must}$, and of course, by rule (μ), $N((P+Q)\|R) \longrightarrow N((P\|R)+(Q\|R))$. Therefore the rule $(\beta_v\|)'$ is not good.

The next Proposition states some characteristic properties of term constructors with respect to the may and must convergence predicates. Proofs are immediate by definition.

4.4.7. PROPOSITION. Let $M, N \in \Lambda^0_{+\|}$, then:

(i) $(M+N)\Downarrow^{must}$ if and only if $M\Downarrow^{must}$ and $N\Downarrow^{must}$.
(ii) $(M\|N)\Downarrow^{must}$ if and only if $M\Downarrow^{must}$ or $N\Downarrow^{must}$.
(iii) $(M+N)\Downarrow^{may}$ if and only if $M\Downarrow^{may}$ or $N\Downarrow^{may}$.
(iv) $(M\|N)\Downarrow^{may}$ if and only if $M\Downarrow^{may}$ or $N\Downarrow^{may}$.
(v) $(\coprod M)\Downarrow^{must}$ if and only if $M\Downarrow^{may}$.

Based on the convergence predicate the following definition adapts to the present setting the notion of contextual theories. This notion stems from [75] and is widely used, e.g. in [17] for the classical theory of solvability, in [6], [22], and [78], where it is shown to be equivalent to applicative bisimulation. The idea is that two terms are operationally equivalent if and only if in all contexts they exhibit the same behaviour with respect to some observable properties. Here the only observable is convergence. Now Proposition 4.4.7 shows that "may" convergence does not distinguish between the behavior of "+" and "$\|$", while "must" convergence does, hence we put:

4.4.8. DEFINITION (Operational Preorder). Let $M, N \in \Lambda_{+\|}$. Then:
(i) $M \sqsubseteq^{\mathcal{O}} N$ iff for all $C[\,]:\ C[M]\Downarrow^{must}$ implies $C[N]\Downarrow^{must}$.
(ii) $\simeq^{\mathcal{O}} = \sqsubseteq^{\mathcal{O}} \cap \sqsupseteq^{\mathcal{O}}$.

It should be noted that $M = N$ implies $M \simeq^{\mathcal{O}} N$, but the opposite does not hold.

4.5. The Type Assignment System

Subsection 4.5.1 presents the typing rules, which make the relation between type constructors and syntactic operators explicit. The types which can be deduced for a term turn out to be strictly connected to the syntactic structure of the term itself (Theorem 4.5.3). In Subsection 4.5.2 we prove some properties of values.

The main results are contained in Subsection 4.5.3, where we will prove that:

- types are preserved under conversion of terms;
- must convergent terms are exactly those which can be typed by $\omega \to \omega$,
- may convergent terms are exactly those which can be typed by $(\omega \to \omega) \oplus \omega$.

4.5.1. Typing Rules

As it is clear from the definition of the reduction relation, a call-by-value variable can be substituted only by a (closed) term in $\mathbf{W}$. This explains way we put $v \in \mathbf{W}$ and the choice, made below, to assume that call-by-value variables are assigned, by the bases, to irreducible types different from ω.

4.5.1. DEFINITION. A *basis* is a mapping $\Gamma : (\mathsf{Vn} \cup \mathsf{Vv} \to \mathit{Type})$ such that:
(i) $\Gamma(x) = \omega$ for all x but a finite subset of Vn;
(ii) $\Gamma(\mathsf{Vv}) \subseteq \mathit{IType}$, and $\Gamma(v) = \omega_1$ for all v but a finite subset of Vv.

The notation $\Gamma, \chi : \sigma$ is a shorthand for the function $\Gamma'(\chi') = \sigma$ if $\chi' \equiv \chi$, $\Gamma(\chi')$ otherwise.

4.5.2. DEFINITION (The Type Assignment System). The axioms and rules of the type assignment system are the following:

$$(\mathrm{Ax})\quad \Gamma \vdash \chi : \Gamma(\chi) \qquad\qquad (\omega)\quad \Gamma \vdash M : \omega$$

$$(\to \mathrm{I}_n)\ \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad\qquad (\to \mathrm{I}_v)\ \frac{\Gamma, v : \sigma_i \vdash M : \tau_i \quad i \in I}{\Gamma \vdash \lambda v.M : \bigoplus_{i\in I}\sigma_i \to \bigoplus_{i \in I}\tau_i}$$

$$(\to \mathrm{E})\ \frac{\Gamma \vdash M : \bigoplus_{i\in I}(\sigma \to \tau_i) \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \bigoplus_{i\in I}\tau_i} \qquad (\to \mathrm{E}\omega)\frac{\Gamma \vdash M : \bigoplus_{i\in I}(\sigma \to \tau_i)\oplus\omega \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \bigoplus_{i\in I}\tau_i \oplus \omega}$$

$$(\wedge\mathrm{I}_{\coprod})\ \frac{\Gamma \vdash M : \sigma \oplus \tau}{\Gamma \vdash \coprod M : \sigma \wedge \tau} \qquad\qquad (\wedge\mathrm{I}_{\|})\ \frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \tau}{\Gamma \vdash M\|N : \sigma \wedge \tau}$$

$$(\oplus\mathrm{I}_{+})\ \frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \tau}{\Gamma \vdash M + N : \sigma \oplus \tau} \qquad\qquad (\wedge\mathrm{I})\ \frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \wedge \tau}$$

$$(\leq)\ \frac{\Gamma \vdash M : \sigma \quad \sigma \leq \tau}{\Gamma \vdash M : \tau}.$$

From now on $\Gamma \vdash M : \sigma$ abbreviates "$\Gamma \vdash M : \sigma$ is derivable using previous axioms and rules".

A few comments are in order. Axiom (ω) is standard for the universal type ω. For call-by-name variables, we have the usual arrow introduction rule $(\to \mathrm{I}_n)$.

The rule $(\to \mathrm{I}_v)$ for call-by-value variables can be explained by considering the reduction rule (β_v+), which is the syntactical tool making call-by-value abstractions into morphisms of $\mathcal{P}$-algebras. Indeed, thanks to the equality **(P0)**, we have:

$$\bigoplus_{i\in I}\sigma_i \to \bigoplus_{i\in I}\tau_i = [(\bigoplus_{i\in I}\sigma_i)\oplus(\bigwedge_{i\in I}\sigma_i)] \to [(\bigoplus_{i\in I}\tau_i)\oplus(\bigwedge_{i\in I}\tau_i)].$$

Moreover in our system $\Gamma, v:\sigma_i \vdash M:\tau_i$ for all $i\in I$ implies $\Gamma, v:\bigwedge_{i\in I}\sigma_i \vdash M:\bigwedge_{i\in I}\tau_i$ (see also 4.5.5). This is why we do not put explicitly the intersections $\bigwedge_{i\in I}\sigma_i$ and $\bigwedge_{i\in I}\tau_i$ in rule $(\to \mathrm{I}_v)$.

For example, assume that W_1, W_2 are in $\mathbf{W}$, of types σ_1,σ_2 respectively, and that $M[W_1/v]$, $M[W_2/v]$ have types τ_1,τ_2 respectively. It follows that $M[W_1\|W_2/v]$ will have type $\tau_1\wedge\tau_2$. In this case we expect $(\lambda v.M)(W_1+W_2)$ to have type $\tau_1\oplus\tau_2$, since

$$\begin{array}{rcl}(\lambda v.M)(W_1+W_2) & \longrightarrow & (\lambda v.M)W_1+(\lambda v.M)W_2+(\lambda v.M)W_1\|W_2\\ & \longrightarrow & M[W_1/v]+M[W_2/v]+M[W_1\|W_2/v].\end{array}$$

As W_1+W_2 has type $\sigma_1\oplus\sigma_2$, a suitable type for $\lambda v.M$ turns out to be $\sigma_1\oplus\sigma_2\to\tau_1\oplus\tau_2$.

Rules $(\to \mathrm{E})$ and $(\to \mathrm{E}\omega)$ take care of the fact that a term may correspond to a sum of functions, instead of a single function. The involved reduction rule is $(+_{app})$, which makes the sum distribute with respect to application.

For example if M_1, M_2 have types $\sigma\to\tau_1$, $\sigma\to\tau_2$ and N has type σ, we expect $(M_1+M_2)N$ to have type $\tau_1\oplus\tau_2$, since $(M_1+M_2)N \longrightarrow M_1N+M_2N$. Instead, if M_1 has type $\sigma\to\tau$, M_2 has type ω, and N has type σ, using $(\to \mathrm{E}\omega)$ we can deduce the type $\tau\oplus\omega$ for $(M_1+M_2)N$, which reduces to M_1N+M_2N.

The typing rule $(\wedge\mathrm{I}_{\coprod})$ reflects the reduction rules $(\coprod_+)$ and $(\coprod_{TV})$. For example if $W_1, W_2 \in \mathbf{W}$ have types σ,τ, respectively, then W_1+W_2 will have type $\sigma\oplus\tau$ and $\coprod(W_1+W_2)$ (which reduces to $W_1\|W_2$) will have type $\sigma\wedge\tau$.

Rule $(\wedge\mathrm{I})$ is standard in intersection type systems. Lastly $(\leq)$ is the subsumption rule which includes in the system the preorder relation on types defined in the previous section.

One can easily show that the following rules are admissible, therefore we will freely use them in the following proofs.

$$(+\mathrm{I})\quad \frac{\Gamma\vdash M:\sigma \qquad \Gamma\vdash N:\sigma}{\Gamma\vdash M+N:\sigma}$$

$$(\|\mathrm{I})\quad \frac{\Gamma\vdash M:\sigma}{\Gamma\vdash M\|N:\sigma} \qquad\qquad \frac{\Gamma\vdash N:\sigma}{\Gamma\vdash M\|N:\sigma}$$

$$(\wedge\ \mathrm{E})\quad \frac{\Gamma\vdash M:\sigma\wedge\tau}{\Gamma\vdash M:\sigma} \qquad\qquad \frac{\Gamma\vdash M:\sigma\wedge\tau}{\Gamma\vdash M:\tau}$$

$$(\leq_L)\quad \frac{\Gamma, x:\sigma\vdash M:\tau \qquad \sigma'\leq\sigma}{\Gamma, x:\sigma'\vdash M:\tau}$$

$$(\oplus\wedge\coprod)\frac{\Gamma\vdash M:\bigoplus_{i\in I}\sigma_i}{\Gamma\vdash \coprod M:\bigwedge_{i\in I}\sigma_i}.$$

In particular, to see that $(\oplus \wedge \coprod)$ is derivable, consider that for all $i \in I$ we have:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash M\colon \bigoplus_{i\in I}\sigma_i}{\Gamma \vdash M{:}\sigma_i \oplus \bigoplus_{j\in I}\sigma_j}\,(\leq)}{\Gamma \vdash \coprod M{:}\sigma_i \wedge \bigoplus_{j\in I}\sigma_j}\,(\wedge \mathrm{I}_{\coprod})}{\Gamma \vdash \coprod M{:}\sigma_i}\,(\leq)$$

from which, by repeated applications of $(\wedge\mathrm{I})$, we derive $\Gamma \vdash \coprod M\colon \bigwedge_{i\in I}\sigma_i$.

Our assignment system enjoys structural properties which show how the types deducible for a term depend on the syntactic structure of the term itself.

4.5.3. Theorem (Derivability properties).

(i) $\Gamma \vdash \chi : \tau$ *if and only if* $\Gamma(\chi) \leq \tau$*;*

(ii) $\Gamma \vdash \lambda\chi.M : \rho$ *if and only if exist* $n,\ \sigma_i, \tau_i\ (i \leq n)$ *such that*

$$\Gamma \vdash \lambda\chi.M : \sigma_i \to \tau_i, \quad \textit{and} \quad \bigwedge_{i\leq n}(\sigma_i \to \tau_i) \leq \rho;$$

(iii) $\Gamma \vdash \lambda x.M : \sigma \to \tau$ *if and only if* $\Gamma, x : \sigma \vdash M : \tau$*;*

(iv) $\Gamma \vdash \lambda v.M : \sigma \to \tau$*, and* $\sigma \leq \omega_1$ *if and only if exist* $m,\ \mu_l, \nu_l\ (l \leq m)$ *such that*

$$\Gamma, v : \mu_l \vdash M : \nu_l, \quad \bigoplus_{l\leq m}\mu_l = \sigma \quad \textit{and} \quad \bigoplus_{l\leq m}\nu_l = \tau;$$

(v) $\Gamma \vdash \lambda v.M : \sigma \to \tau$ *and* $\sigma \not\leq \omega_1$ *if and only if* $\tau = \omega$*;*

(vi) $\Gamma \vdash MN : \tau$ *and* $\tau \leq \omega_1$ *if and only if exist* $n, \tau_i\ (i \leq n), \sigma$ *such that*

$$\Gamma \vdash M : \bigoplus_{i\leq n}(\sigma \to \tau_i), \quad \Gamma \vdash N : \sigma \quad \textit{and} \quad \bigoplus_{i\leq n}\tau_i = \tau;$$

(vii) $\Gamma \vdash MN : \tau$*,* $\tau = \tau \oplus \omega \neq \omega$ *if and only if exist* $n, \tau_i\ (i \leq n), \sigma$ *such that*

$$\Gamma \vdash M : \bigoplus_{i\leq n}(\sigma \to \tau_i) \oplus \omega, \quad \Gamma \vdash N : \sigma \quad \textit{and} \quad \bigoplus_{i\leq n}\tau_i \oplus \omega = \tau;$$

(viii) $\Gamma \vdash M + N : \tau$ *if and only if* $\Gamma \vdash M : \sigma$*,* $\Gamma \vdash N : \sigma'$ *and* $\sigma \oplus \sigma' \leq \tau$*, for some* σ, σ'*;*

(ix) $\Gamma \vdash M \| N : \tau$ *if and only if* $\Gamma \vdash M : \sigma$*,* $\Gamma \vdash N : \sigma'$ *and* $\sigma \wedge \sigma' \leq \tau$*, for some* σ, σ'*;*

(x) $\Gamma \vdash \coprod M : \tau$ *if and only if* $\Gamma \vdash M : \bigoplus_{i\leq n}\tau_i$ *and* $\bigwedge_{i\leq n}\tau_i \leq \tau$*, for some* $n,\ \tau_i\ (i \leq n)$*.*

Proof. All $(\Leftarrow)$ follow immediately from the typing rules. For $(\Rightarrow)$ we consider only the interesting cases.

(ii) Given a derivation of $\Gamma \vdash \lambda\chi.M{:}\rho$, let

$$\Gamma \vdash \lambda\chi.M{:}\sigma_1 \to \tau_1, \ldots, \Gamma \vdash \lambda\chi.M{:}\sigma_n \to \tau_n$$

be all the statements in this deduction on which $\Gamma \vdash \lambda\chi.M{:}\tau$ depends and which are conclusions of rule $(\to \mathrm{I}_n)$ or of rule $(\to \mathrm{I}_v)$. Then, observing that the only rules that leave the subject unchanged are $(\wedge\,\mathrm{I})$ and $(\leq)$, we conclude that

$$\bigwedge_{i\leq n}(\sigma_i \to \tau_i) \leq \rho.$$

(iii) Let $\sigma_1, \ldots, \sigma_n, \tau_1, \ldots, \tau_n$ be as in the proof of *(ii)* where ρ is replaced by $\sigma \to \tau$. Then

$$\bigwedge_{i \le n} (\sigma_i \to \tau_i) \le \sigma \to \tau$$

which implies, by **(P4)**,

$$\exists J \subseteq \{1, \cdots, n\}.\ \sigma \le \bigwedge_{j \in J} \sigma_j \ \&\ \bigwedge_{j \in J} \tau_j \le \tau.$$

Moreover, by looking at the premises of inferences by which the judgments $\Gamma \vdash \lambda\chi.M : \sigma_i \to \tau_i$ are derived, we know that $\Gamma, x{:}\,\sigma_i \vdash M : \tau_i$ for $1 \le i \le n$, so that we conclude that $\Gamma, x{:}\,\sigma \vdash M{:}\,\tau$ by $(\le_L)$, $(\wedge\,\mathrm{I})$ and $(\le)$.

(iv) Let $\sigma_1, \ldots, \sigma_n, \tau_1, \ldots, \tau_n$ be as in the proof of *(ii)* where ρ is replaced by $\sigma \to \tau$. Similarly to case *(iii)* we have

$$\exists J \subseteq \{1, \cdots, n\}.\ \sigma \le \bigwedge_{j \in J} \sigma_j \ \&\ \bigwedge_{j \in J} \tau_j \le \tau.$$

Let $\Theta(\sigma) = \{\xi_r \mid r \le p\}$.

By construction

$$\Gamma, v{:}\,\alpha_{i,h} \vdash M{:}\,\delta_{i,h} \tag{4.35}$$

for some $\alpha_{i,h}, \delta_{i,h}$ such that $\sigma_i = \bigoplus_{h \in H_i} \alpha_{i,h}$ and $\tau_i = \bigoplus_{h \in H_i} \delta_{i,h}$. By **(P2)**

$$\sigma \le \bigwedge_{j \in J} \sigma_j \quad \text{implies} \quad \bigwedge_{r \le p} \xi_r \le \bigwedge_{j \in J} (\bigwedge_{h \in H_j} \alpha_{j,h}) \tag{4.36}$$

$$\begin{array}{lll}
\forall i \in I, h \in H_i.\ \Gamma, v{:}\,\alpha_{i,h} \vdash M{:}\,\delta_{i,h} & \Rightarrow & \forall j \in J, h \in H_j.\ \Gamma, v{:}\ \bigwedge_{r \le p} \xi_r \vdash M{:}\,\delta_{j,h} \\
& & \text{by } (4.36) \text{ and rule } (\le_L) \\
& \Rightarrow & \Gamma, v{:}\ \bigwedge_{r \le p} \xi_r \vdash M{:}\ \bigwedge_{j \in J} (\bigwedge_{h \in H_j} \delta_{j,h}) \\
& & \text{by rule } (\wedge\,\mathrm{I}) \\
& \Rightarrow & \Gamma, v{:}\ \bigwedge_{r \le p} \xi_r \vdash M{:}\ \bigwedge_{j \in J} \tau_j \\
& & \text{by rule } (\le) \text{ since} \\
& & \bigwedge_{h \in H_j} \delta_{j,h} \le \bigoplus_{h \in H_j} \delta_{j,h} = \tau_j \\
& \Rightarrow & \Gamma, v{:}\ \bigwedge_{r \le p} \xi_r \vdash M{:}\,\tau \qquad (4.37) \\
& & \text{by rule } (\le).
\end{array}$$

If $\sigma \le \sigma_j$ for all $j \in J$ then, by **(P2)**,

$$\forall j \in J\ \forall r \le p\ \exists f(j,r) \in H_j.\ \xi_r \le \alpha_{j,f(j,r)}. \tag{4.38}$$

Set $G = \{g : J \to \bigcup_{j \in J} H_j \mid \forall j \in J. g(j) \in H_j\}$.

$$\begin{array}{rcll}
\bigwedge_{j\in J}\tau_j \leq \tau & \Rightarrow & \bigwedge_{j\in J}(\bigoplus_{h\in H_j}\delta_{j,h}) \leq \tau & \\
& \Rightarrow & \bigoplus_{g\in G}(\bigwedge_{j\in J}\delta_{j,g(j)}) \leq \tau & \text{by distributivity} \\
& \Rightarrow & \forall g\in G.\exists\nu_g.\ \bigwedge_{j\in J}\delta_{j,g(j)} \leq \nu_g\ \& & \\
& & \tau = \tau\oplus\nu_g & \text{by } \mathbf{(P1)} \\
& \Rightarrow & \forall r\leq p\ \exists\nu_r.\ \bigwedge_{j\in J}\delta_{j,f(j,r)} \leq \nu_r\ \&\ \tau=\tau\oplus\nu_r & \\
& & & \text{choosing } g(j)=f(j,r) \\
& & & \text{in the above statement.}
\end{array} \tag{4.39}$$

From (4.35) we have $\Gamma, v\colon \bigwedge_{j\in J}\alpha_{j,f(j,r)} \vdash M\colon \bigwedge_{j\in J}\delta_{j,f(j,r)}$ by rules $(\leq_L)$, and $(\wedge\mathrm{I})$. Therefore by (4.38) and (4.39) we deduce $\forall r\leq p.\ \Gamma, v\colon\xi_r \vdash M\colon\nu_r$. Taking into account (4.37), we choose $m=p+1,\ \mu_1=\xi_1,\ldots,\ \mu_p=\xi_p,\ \mu_{p+1}=\bigwedge_{r\leq p}\xi_r$ and $\nu_{p+1}=\tau$.

(v) Let $\sigma_1,\ldots,\sigma_n,\tau_1,\ldots,\tau_n,J$ be as in the proof of *(iv)*. By definition of basis we have $\sigma_i\leq\omega_1$ for all $i\in I$, therefore $\sigma\leq\bigwedge_{i\in I}\sigma_i$ implies $\sigma\leq\omega_1$. This is a contradiction. Therefore $\Gamma\vdash M\colon\sigma\to\tau$ can only be obtained by rule $(\leq)$ stating $\tau=\omega$.

(vi) Given a deduction of $\Gamma\vdash MN\colon\tau$, let

$$\Gamma\vdash MN\colon\rho_1,\ldots,\Gamma\vdash MN\colon\rho_m$$

be all the statements in this deduction on which $\Gamma\vdash MN\colon\tau$ depends, and which are conclusions of rule $(\to\mathrm{E})$ or of rule $(\to\mathrm{E}\omega)$. Let $H=\{h\leq m\mid \Gamma\vdash MN\colon\rho_h$ is conclusion of rule $(\to\mathrm{E})\}$ and $K=\{k\leq m\mid \Gamma\vdash MN\colon\rho_k$ is conclusion of rule $(\to\mathrm{E}\omega)\}$, $(H\cup K=\{1,\cdots,m\})$. Clearly for all $k\in K$ we have $\rho_k=\rho_k\oplus\omega$.
If $\tau\leq\omega_1$, since $\bigwedge_{j\leq m}\rho_j\leq\tau$, we must have $\rho_j\leq\omega_1$ for some $j\leq m$. So H is non-empty, while K can be empty.
For all $h\in H$ exist $L_h,\mu_h,\nu_{h,l}$, where $l\in L_h$, such that:
$\Gamma\vdash M\colon\bigoplus_{l\in L_h}(\mu_h\to\nu_{h,l})$, $\Gamma\vdash N\colon\mu_h$, and $\bigoplus_{l\in L_h}\nu_{h,l}=\rho_h$. Similarly, $\forall k\in K.\exists L_k,\mu_k,\nu_{k,l}$, where $l\in L_k$, such that:
$\Gamma\vdash M\colon\bigoplus_{l\in L_k}(\mu_k\to\nu_{k,l})\oplus\omega$, $\Gamma\vdash N\colon\mu_k$ and $\bigoplus_{l\in L_k}\nu_{k,l}\oplus\omega=\rho_k$.

Let $\sigma=\bigwedge_{j\leq m}\mu_j$. We can deduce $\Gamma\vdash N\colon\sigma$ using $(\wedge\ \mathrm{I})$. Since $\sigma\leq\mu_j$ for all $j\in J$, we have

$$\begin{array}{l}
[\bigoplus\limits_{l\in L_h}(\mu_h\to\nu_{h,l})]\wedge[\bigoplus\limits_{k\in L_k}(\mu_k\to\nu_{k,l})\oplus\omega] = \\
\quad = \ [(\bigoplus\limits_{l\in L_h}(\mu_h\to\nu_{h,l}))\wedge(\bigoplus\limits_{k\in L_k}(\mu_k\to\nu_{k,l}))]\oplus[\bigoplus\limits_{l\in L_h}(\mu_h\to\nu_{h,l})] \\
\quad \leq \ [(\bigoplus\limits_{l\in L_h}(\sigma\to\nu_{h,l}))\wedge(\bigoplus\limits_{l\in L_k}(\sigma\to\nu_{k,l}))]\oplus[\bigwedge\limits_{l\in L_h}(\sigma\to\nu_{h,l})] \\
\quad = \ [\bigoplus\limits_{l\in L_h,l'\in L_k}(\sigma\to\nu_{h,l}\wedge\nu_{k,l'})]\oplus[\bigoplus\limits_{l\in L_h}(\sigma\to\nu_{h,l})].
\end{array}$$

Now let $G = \{g : J \to \bigcup_{j\in J} L_j \mid \forall j \in J.\ g(j) \in L_j\}$. Because of

$$\begin{array}{rcl}\bigwedge_{j\leq m}[\bigoplus_{l\in L_j}(\sigma \to \nu_{j,l})] & = & \bigoplus_{g\in G}[\bigwedge_{j\leq m}(\sigma \to \nu_{j,g(j)})] \\ & & \text{by distributivity} \\ & = & \bigoplus_{g\in G}(\sigma \to \bigwedge_{j\leq m} \nu_{j,g(j)}) \\ & = & \bigoplus_{g\in G}(\sigma \to \bigwedge_{j\leq m} \nu_{j,g(j)}) \oplus (\sigma \to \bigwedge_{j\leq m, g\in G} \nu_{j,g(j)}) \\ & & \text{by } \mathbf{(P0)} \\ & = & \bigoplus_{g\in G}(\sigma \to \bigwedge_{j\leq m} \nu_{j,g(j)}) \oplus (\sigma \to \bigwedge_{j\leq m, l\in L_j} \nu_{j,l})\end{array}$$

we deduce the last type for M using ($\wedge$ I) and ($\leq$). Therefore we deduce $\Gamma \vdash MN : \bigoplus_{g\in G}(\bigwedge_{j\leq m} \nu_{j,g(j)}) \oplus (\bigwedge_{j\leq m, l\in L_j} \nu_{j,l})$ by rule ($\to$ E).

$$\begin{array}{rcl}\bigwedge_{j\leq m} \rho_j \leq \tau & \Rightarrow & \bigwedge_{j\leq m}(\bigoplus_{l\in L_j} \nu_{j,l}) \leq \tau \\ & \Rightarrow & \bigoplus_{g\in G}(\bigwedge_{j\leq m} \nu_{j,g(j)}) \leq \tau \\ & & \text{by distributivity} \\ & \Rightarrow & \forall g \in G.\ \exists \tau_g.\ \bigwedge_{j\leq m} \nu_{j,g(j)} \leq \tau_g \ \&\ \tau = \tau_g \oplus \tau \\ & & \text{by } \mathbf{(P1)} \\ & \Rightarrow & \bigoplus_{g\in G}(\bigwedge_{j\leq m} \nu_{j,g(j)}) \oplus (\bigwedge_{j\leq m, l\in L_j} \nu_{j,l}) \leq \bigoplus_{g\in G} \tau_g \oplus \tau = \tau.\end{array}$$

So we conclude $\Gamma \vdash MN : \tau$ by rule ($\leq$).

(vii) Given a deduction of $\Gamma \vdash MN\colon \tau = \tau \oplus \omega$, let

$$\Gamma \vdash MN\colon \rho_1, \ldots, \Gamma \vdash MN\colon \rho_m,$$

$H, L_h, \mu_h, \nu_h, K, \mu_k, \nu_k, \sigma, G$ be as in the proof of *(vi)*. Now H or K can be empty, but not both. Since $\bigoplus_{l\in L_h}(\mu_h \to \nu_{h,l}) \leq \bigoplus_{l\in L_h}(\mu_h \to \nu_{h,l}) \oplus \omega$, we have $\bigwedge_{h\in H}[\bigoplus_{l\in L_h}(\mu_h \to \nu_{h,l})] \wedge \bigwedge_{k\in K}[\bigoplus_{l\in L_k}(\mu_k \to \nu_{k,l}) \oplus \omega] \leq \bigwedge_{j\leq m}[\bigoplus_{l\in L_j}(\mu_j \to \nu_{j,l}) \oplus \omega]$. As in the proof of previous point we have:

$$\bigwedge_{j\leq m}[\bigoplus_{l\in L_j}(\mu_j \to \nu_{j,l}) \oplus \omega] \leq \bigoplus_{g\in G}(\sigma \to \bigwedge_{j\leq m} \nu_{j,g(j)}) \oplus (\sigma \to \bigwedge_{j\leq m, l\in L_j} \nu_{j,l}) \oplus \omega,$$

$$\text{and } \bigoplus_{g\in G}(\bigwedge_{j\leq m} \nu_{j,g(j)}) \oplus (\bigwedge_{j\leq m, l\in L_j} \nu_{j,l}) \oplus \omega \leq \tau.$$

So we can deduce the type $\bigoplus_{g\in G}(\sigma \to \bigwedge_{j\leq m} \nu_{j,g(j)}) \oplus (\sigma \to \bigwedge_{j\leq m, l\in L_j} \nu_{j,l}) \oplus \omega$ for M and by ($\to$ Eω) the type $\bigoplus_{g\in G}(\bigwedge_{j\leq m} \nu_{j,g(j)}) \oplus (\bigwedge_{j\leq m, l\in L_j} \nu_{j,l}) \oplus \omega$ for MN. We conclude using ($\leq$).

(viii) Again, given a deduction of $\Gamma \vdash M + N\colon \tau$, let

$$\Gamma \vdash M + N\colon \tau_1, \ldots, \Gamma \vdash M + N\colon \tau_n$$

be all the statements in this deduction on which $\Gamma \vdash M + N\colon \tau$ depends and which are conclusions of rule ($\oplus$I$_+$). Then $\bigwedge_{i\leq n} \tau_i \leq \tau$ and there are σ_i, σ'_i such that $\Gamma \vdash M\colon \sigma_i, \Gamma \vdash N\colon \sigma'_i$ and $\tau_i = \sigma_i \oplus \sigma'_i$ for $1 \leq i \leq n$. So we can choose $\sigma \equiv \bigwedge_{i\leq n} \sigma_i, \sigma' \equiv \bigwedge_{i\leq n} \sigma'_i$ since $\sigma \oplus \sigma' \leq \tau_i$ for all $i \leq n$ and we can deduce $\Gamma \vdash M\colon \sigma$, $\Gamma \vdash N\colon \sigma'$ using ($\wedge$ I).

(ix) Given a deduction of $\Gamma \vdash M \| N : \tau$, let

$$\Gamma \vdash M \| N : \tau_1, \ldots, \Gamma \vdash M \| N : \tau_n$$

be all the statements in this deduction on which $\Gamma \vdash M \| N : \tau$ depends and which are conclusions of rule $(\wedge \mathrm{I}_{\|})$. Then $\bigwedge_{i \leq n} \tau_i \leq \tau$ and there are σ_i, σ_i' such that $\Gamma \vdash_{\mathcal{L}} M : \sigma_i, \Gamma \vdash N : \sigma_i'$ and $\tau_i = \sigma_i \wedge \sigma_i'$ for $1 \leq i \leq n$. So we can proceed as in previous case.

(x) Finally, given a deduction of $\Gamma \vdash \coprod M : \tau$, let

$$\Gamma \vdash \coprod M : \sigma_1, \ldots, \Gamma \vdash \coprod M : \sigma_m$$

be all the statements in this deduction on which $\Gamma \vdash \coprod M : \tau$ depends and which are conclusions of rule $(\wedge \mathrm{I}_{\coprod})$. Then $\bigwedge_{i \leq m} \sigma_i \leq \tau$ and there are ρ_i, ρ_{i+m} such that $\Gamma \vdash M : \rho_i \oplus \rho_{i+m}$ and $\sigma_i = \rho_i \wedge \rho_{i+m}$ for $1 \leq i \leq m$. So we can choose $n = 2m$. We have $\Gamma \vdash M : \bigoplus_{i \leq 2m} \rho_i$. We are done since $\bigwedge_{i \leq 2m} \rho_i \leq \tau$.

The previous Lemma shows that the types which can be deduced for a given term respect its formation. In other words, all rules obtainable by reversing the structural rules of our type assignment system are admissible.

4.5.2. Properties of Values

Inside the set of terms, values play surely a special role, thank to their syntactic form. We prove in this section some properties of the types which can be deduced for values and for terms containing them.

4.5.4. Theorem.

(i) $V \in \mathbf{V}$ *implies* $\vdash V : \omega_1$*;*
(ii) $W \in \mathbf{W}$ *and* $\Gamma \vdash W : \bigoplus_{i \in I} \sigma_i$ *imply* $\Gamma \vdash W : \bigwedge_{i \in I} \sigma_i$*;*
(iii) $\Gamma \vdash WM : \tau$ *and* $W \in \mathbf{W}$ *imply* $\Gamma \vdash W : \sigma \to \tau$ *and* $\Gamma \vdash M : \sigma$*, for some* σ*;*
(iv) $\Gamma, \chi : \sigma \vdash W\chi : \tau$ *and* $W \in \mathbf{W}^0$ *imply* $\vdash W : \sigma \to \tau$*;*
(v) $\Gamma, v : \sigma \vdash vM : \tau$ *implies* $\sigma \leq \rho \to \tau$ *and* $\Gamma \vdash M : \rho$*, for some* ρ*;*
(vi) $\Gamma \vdash m(V) : \tau$ *implies* $\Gamma \vdash V : \tau$*.*

Proof. *(i).* By induction on the definition of values. If $V \equiv v \in \mathsf{Vv}$ then $\Gamma(v) = \omega_1$, since $Dom(\Gamma) = \emptyset$, and the statement follows by (Ax). If $V \equiv \lambda x.M$ then $\Gamma, x : \omega \vdash M : \omega$ is derivable, by rule (ω); hence the statement follows using $(\to \mathrm{I}_n)$. If $V \equiv \lambda v.M$ we do the same as before but assuming $v : \omega_1$. The statement follows using $(\to \mathrm{I}_v)$ and $(\leq)$. If $V \equiv U \| M$ for $U \in \mathbf{U}$, the statement follows by the induction hypothesis using $(\wedge I_{\|})$. Finally if $V \equiv V' + V''$ the statement follows by the induction hypothesis and $(+\mathrm{I})$.

(ii). By induction on W. If $W \equiv v$ then by definition of basis and by 4.5.3*(i)* W is typeable by an irreducible type σ less than $\bigoplus_{i \in I} \sigma_i$. If $W \equiv \lambda \chi.M$ then by 4.5.3*(ii)* W is typeable by an irreducible type σ less than $\bigoplus_{i \in I} \sigma_i$, since an intersection of arrows is in $IType$ by 4.3.8*(v)*. It follows that $\sigma \leq \sigma_i$ for all $i \in I$ by **(P2)**. If $W \equiv W' \| W''$, by 4.5.3*(ix)* there exist ρ' and ρ'' such that $\Gamma \vdash W' : \rho'$, $\Gamma \vdash W'' : \rho''$ and $\rho' \wedge \rho'' \leq \bigoplus_{i \in I} \sigma_i$. By the induction hypothesis it is not restrictive to assume that ρ' and ρ'' are themselves irreducible, so that $\rho' \wedge \rho'' \in IType$ by 4.3.8*(iv)* and the statement follows.

(iii). The case $\tau = \omega$ is trivial. Let $\tau \neq \tau \oplus \omega$.

$$\begin{array}{lcll}
\Gamma \vdash WM : \tau & \Rightarrow & \exists I, \tau_i, \sigma.\, \Gamma \vdash W : \bigoplus_{i\in I}(\sigma \to \tau_i) \ \& & \\
& & \Gamma \vdash M : \sigma \ \& \ \bigoplus_{i\in I} \tau_i = \tau & \text{by } 4.5.3\textit{(vi)} \\
& \Rightarrow & \exists I, \tau_i, \sigma.\, \Gamma \vdash W : \bigwedge_{i\in I}(\sigma \to \tau_i) & \text{by } \textit{(ii)} \\
& \Rightarrow & \Gamma \vdash W : \sigma \to \tau & \text{by } (\leq) \text{ since } \bigwedge_{i\in I} \tau_i \leq \tau.
\end{array}$$

If $\tau = \tau \oplus \omega$ the proof is similar using 4.5.3*(vii)* instead of 4.5.3*(vi)*.

(iv). Immediate from *(iii)* and 4.5.3*(i)*.

(v). Immediate from *(iii)* and 4.5.3*(i)*.

(vi). By induction on the definition of values using rule ($\|$I). □

From 4.5.4*(ii)* we obtain a restricted form of $\oplus$ elimination. More precisely, the following rule ($\oplus$ E) turns out to be admissible:

$$(\oplus\,\mathrm{E}) \ \frac{\Gamma, \chi : \sigma_i \vdash M : \tau_i \quad \forall i \in I \quad \Gamma \vdash W : \bigoplus_{i\in I} \sigma_i \quad W \in \mathbf{W}}{\Gamma \vdash M[W/\chi] : \bigoplus_{i\in I} \tau_i}$$

In fact $\Gamma \vdash W : \bigoplus_{i\in I} \sigma_i$ implies $\Gamma \vdash W : \bigwedge_{i\in I} \sigma_i$. From this and $\Gamma, \chi : \sigma_i \vdash M : \tau_i$ it is easy to find a derivation of $\Gamma \vdash M[W/\chi] : \tau_i$ for all $i \in I$. By the way, notice that one can also deduce $\Gamma \vdash M[W/\chi] : \bigwedge_{i\in I} \tau_i$.

Among values in $\mathbf{W}$, call-by-value abstractions set up a subset whose types satisfy particular properties. More specifically, the following Proposition shows the linearity of call-by-value abstractions with respect to the non-deterministic choice operator.

4.5.5. Proposition.

(i) $\Gamma \vdash \lambda v.M : \sigma \to \tau$ implies either $\sigma \leq \omega_1$ or $\tau = \omega$;

(ii) If $\Gamma \vdash \lambda v.M : \sigma \to \tau$, $\sigma \leq \omega_1$ and $\Theta(\sigma) = \{\sigma_i \mid i \in I\}$ then $\Gamma, v : \bigwedge_{i\in I} \sigma_i \vdash M : \tau$ and for all $i \in I$ there is τ_i such that $\Gamma, v : \sigma_i \vdash M : \tau_i$, and $\tau = (\bigoplus_{i\in I} \tau_i) \oplus \tau$;

(iii) $\Gamma \vdash \lambda v.M : \sigma \to \tau$, $\sigma \leq \omega_1$, and $\Gamma \vdash \lambda v.M : \mu \to \nu$, $\mu \leq \omega_1$ imply $\Gamma \vdash \lambda v.M : \sigma \oplus \mu \to \tau \oplus \nu$;

(iv) $\Gamma \vdash \lambda v.M : \sigma \oplus \tau \to \rho$ implies that there are ρ_1, ρ_2 such that $\Gamma \vdash \lambda v.M : \sigma \to \rho_1$, $\Gamma \vdash \lambda v.M : \tau \to \rho_2$, and $\rho = \rho_1 \oplus \rho_2 \oplus \rho$.

Proof.

(i). Immediate from 4.5.3*(iv)* and 4.5.3*(v)*.

(ii). By 4.5.3*(iv)* $\Gamma \vdash \lambda v.M : \sigma \to \tau$ implies

$$\exists J, \mu_j, \nu_j.\ \sigma = \bigoplus_{j\in J} \mu_j \ \& \ \tau = \bigoplus_{j\in J} \nu_j \ \& \ \forall j \in J.\, \Gamma, v : \mu_j \vdash M : \nu_j. \tag{4.40}$$

Notice that each $\mu_j \in \mathit{IType}$ by definition of basis.

By 4.3.8*(i)* $\sigma = \bigoplus_{i\in I} \sigma_i$ and each σ_i is irreducible by 4.3.8*(vii)*. Therefore by **(P2)** $\bigwedge_{i\in I} \sigma_i = \bigwedge_{j\in J} \mu_j$, so we can deduce from (4.40) that $\Gamma, v : \bigwedge_{i\in I} \sigma_i \vdash M : \tau$ using ($\leq_L$), ($\wedge$ I) and ($\leq$). Again by **(P2)** we have $\forall i \in I.\exists f(i) \in J.\ \sigma_i \leq \mu_{f(i)}$. So we can choose $\tau_i \equiv \nu_{f(i)}$.

(iii). By 4.5.3*(iv)* $\Gamma \vdash \lambda v.M : \sigma \to \tau$ implies

$$\exists I, \sigma_i, \tau_i.\ \sigma = \bigoplus_{i \in I} \sigma_i \ \&\ \tau = \bigoplus_{i \in I} \tau_i \ \&\ \forall i \in I.\ \Gamma, v : \sigma_i \vdash M : \tau_i. \tag{4.41}$$

In the same way $\Gamma \vdash \lambda v.M : \mu \to \nu$ implies

$$\exists J, \mu_j, \nu_j.\ \mu = \bigoplus_{j \in J} \mu_j \ \&\ \nu = \bigoplus_{j \in J} \nu_j \ \&\ \forall j \in J.\ \Gamma, v : \mu_j \vdash M : \nu_j. \tag{4.42}$$

Applying rule $(\to \mathrm{I}_v)$ to (4.41) and (4.42) we can derive the desired statement.

(iv). The case $\rho = \omega$ is trivial. Otherwise let $\Theta(\sigma) = \{\sigma_i \mid i \leq n\}, \Theta(\tau) = \{\tau_j \mid j \leq m\}$. This implies $\Theta(\sigma \oplus \tau) = \Theta(\sigma) \cup \Theta(\tau)$.

$$\begin{array}{lll} \Gamma \vdash \lambda v.M : \sigma \oplus \tau \to \rho & \Rightarrow & \exists \mu_1, \ldots, \mu_{n+m}.\ \rho = (\bigoplus_{l \leq n+m} \mu_l) \oplus \rho\ \& \\ & & \forall i \leq n.\ \Gamma, v : \sigma_i \vdash M : \mu_i\ \& \\ & & \forall j \leq m.\ \Gamma, v : \tau_j \vdash \mu_{n+j} \qquad\qquad \text{by } \textit{(ii)} \\ & \Rightarrow & \Gamma \vdash \lambda v.M : \sigma \to \bigoplus_{i \leq n} \mu_i\ \& \\ & & \Gamma \vdash \lambda v.M : \tau \to \bigoplus_{j \leq m} \mu_{n+j} \text{by } (\to \mathrm{I}_v). \end{array}$$

Therefore we can choose $\rho_1 = \bigoplus_{i \leq n} \mu_i$ and $\rho_2 = \bigoplus_{j \leq m} \mu_{n+j}$. □

4.5.3. Convergence implies Typability

The following statement is the type invariance theorem, which gives a first evidence of the matching between operational and logical semantics. The proof is a rather tedious but straightforward consequence of the properties of the preorder on types and of the structural properties of deductions.

4.5.6. THEOREM (Type Invariance). *$\Gamma \vdash M : \tau$ and $M = N$ imply $\Gamma \vdash N : \tau$.*

Proof. First we claim that

$$M \approx N \ \&\ \Gamma \vdash M : \tau \Rightarrow \Gamma \vdash N : \tau.$$

The claim is easily established by induction on M using 4.5.3 and the properties of $\oplus$ and $\wedge$ with respect to the relation $\leq$.

To prove the theorem it suffices to prove the statement when $M \longrightarrow N$ or $N \longrightarrow M$. We consider only the interesting cases.

$M \longrightarrow N$ by rule $(\beta_v \|)$. In this case we have $M \equiv (\lambda v.L)U$ for some $U \in \mathbf{U}$ and $N \equiv L[m(U)/v] \| (\lambda v.L)V$ under the hypothesis $U \longrightarrow V$. Clearly by induction M and $(\lambda v.L)V$ have the same types, so it suffices to prove that $\Gamma \vdash L[m(U)/v] : \tau$ implies $\Gamma \vdash M : \tau$.

Fix a derivation $\mathcal{D}$ of $\Gamma \vdash L[m(U)/v] : \tau$. Let σ be the intersection of all the types assigned to $m(U)$ in $\mathcal{D}$. The case $\sigma = \omega$ is trivial. Otherwise, let $\Theta(\sigma) = \{\sigma_i \mid i \in I\}$. We have by 4.5.4*(ii)* that $\Gamma \vdash m(U) : \bigwedge_{i \in I} \sigma_i$, being $m(U) \in \mathbf{W}$. By 4.5.4*(vi)* this implies

$$\Gamma \vdash U : \bigwedge_{i \in I} \sigma_i. \tag{4.43}$$

Moreover from $\mathcal{D}$ we can build a derivation of

$$\Gamma, v : \bigwedge_{i\in I} \sigma_i \vdash L : \tau \tag{4.44}$$

using ($\leq_L$). Therefore from (4.43) and (4.44) by ($\rightarrow$ I_v) and ($\rightarrow$ E) we conclude $\Gamma \vdash M : \tau$.

$M \longrightarrow N$ by rule (β_v+). In this case we have $M \approx (\lambda v.L)(\sum_{i\leq n} U_i)$ for $U_1, \ldots, U_n \in \mathbf{U}$, and $N \equiv \sum_{i\leq n}(\lambda v.L)U_i + (\lambda v.L)(U_1\| \ldots \|U_n)$ for some n.

$$\begin{array}{rcl} \Gamma \vdash M : \tau & \Rightarrow & \exists\sigma.\, \Gamma \vdash \lambda v.L : \sigma \rightarrow \tau \\ & & \&\; \Gamma \vdash \sum_{i\leq n} U_i : \sigma \\ & & \qquad\qquad\qquad \text{by 4.5.4}\textit{(iii)} \end{array} \tag{4.45}$$

We assume that $\sigma \leq \omega_1$, since otherwise we have $\tau = \omega$ by 4.5.3*(v)*.

$$\Gamma \vdash \textstyle\sum_{i\leq n} U_i : \sigma \;\Rightarrow\; \exists\sigma_i.\, \Gamma \vdash U_i : \sigma_i \;\&\; \bigoplus_{i\leq n} \sigma_i \leq \sigma \quad \text{by 4.5.3}\textit{(viii)} \tag{4.46}$$

$$\Gamma \vdash \lambda v.L : \sigma \rightarrow \tau \;\Rightarrow\; \Gamma \vdash \lambda v.L : \textstyle\bigoplus_{i\leq n} \sigma_i \rightarrow \tau \quad \text{by rule } (\leq) \tag{4.47}$$

$$\Rightarrow\; \exists\rho_i.\, \Gamma \vdash \lambda v.L : \sigma_i \rightarrow \rho_i \;\&\; \textstyle\bigoplus_{i\leq n} \rho_i \oplus \tau = \tau \quad \text{by 4.5.5}\textit{(iv)}. \tag{4.48}$$

(4.49)

Then we can derive $\Gamma \vdash (\lambda v.L)U_i : \rho_i$, (from (4.48) and (4.46)), and $\Gamma \vdash (\lambda v.L)(U_1\| \ldots \|U_n) : \tau$, from (4.47), since $\Gamma \vdash U_1\| \ldots \|U_n : \sigma_1 \oplus \ldots \oplus \sigma_n$. Therefore we have $\Gamma \vdash \sum_{i\leq n}(\lambda v.L)U_i + (\lambda v.L)(U_1\| \ldots \|U_n) : \bigoplus_{i\leq n} \rho_i \oplus \tau$ by ($\oplus\mathrm{I}_+$), and we are done.
Vice-versa suppose that $\Gamma \vdash N : \tau$; then

$$\begin{array}{rcl} \Gamma \vdash N : \tau & \Rightarrow & \exists\nu_i, \nu.\, \Gamma \vdash (\lambda v.L)U_i : \nu_i \;\& \\ & & \Gamma \vdash (\lambda v.L)(U_1\| \ldots \|U_n) : \nu \;\&\; \bigoplus_{i\leq n} \nu_i \oplus \nu \leq \tau \\ & & \text{by 4.5.3}\textit{(viii)} \end{array}$$

$$\begin{array}{rcll} \Gamma \vdash (\lambda v.L)U_i : \nu_i & \Rightarrow & \exists\mu_i.\, \Gamma \vdash \lambda v.L : \mu_i \rightarrow \nu_i \;\& & (4.50) \\ & & \Gamma \vdash U_i : \mu_i \qquad \text{by 4.5.4}\textit{(iii)} & (4.51) \end{array}$$

$$\begin{array}{rcll} \Gamma \vdash (\lambda v.L)(U_1\| \ldots \|U_n) : \nu & \Rightarrow & \exists\sigma.\, \Gamma \vdash \lambda v.L : \sigma \rightarrow \nu \;\&\; \Gamma \vdash U_1\| \ldots \|U_n : \sigma & (4.52) \\ & & \text{by 4.5.4}\textit{(iii)} & \\ \Gamma \vdash U_1\| \ldots \|U_n : \sigma & \Rightarrow & \exists\sigma_i.\, \Gamma \vdash U_i : \sigma_i \;\&\; \bigwedge_{i\leq n} \sigma_i \leq \sigma \text{ by 4.5.3}\textit{(ix)}. & (4.53) \end{array}$$

Let $\rho_i = \mu_i \wedge \sigma_i \wedge \omega_1$ and $\rho = \bigwedge_{i\leq n} \rho_i$. From (4.50), and (4.52) we can deduce using ($\leq$) that $\Gamma \vdash \lambda v.L : \rho_i \rightarrow \nu_i$ and $\Gamma \vdash \lambda v.L : \rho \rightarrow \nu$. Notice that $\rho_i \leq \omega_1$, therefore by 4.5.5*(iii)*

$$\Gamma \vdash \lambda v.L : \bigoplus_{i\leq n} \rho_i \oplus \rho \rightarrow \bigoplus_{i\leq n} \nu_i \oplus \nu.$$

(4.53), (4.51), and 4.5.4*(i)* imply $\Gamma \vdash U_i : \rho_i$, so by $(\oplus \mathrm{I}_+)$ and $(\leq)$ we have

$$\Gamma \vdash \sum_{i \leq n} U_i : \bigoplus_{i \leq n} \rho_i \oplus \rho.$$

since, by **(P0)**, $\bigoplus_{i \leq n} \rho_i \oplus \rho = \bigoplus_{i \leq n} \rho_i$. So we conclude

$$\Gamma \vdash (\lambda v.L)(\sum_{i \leq n} U_i) : \bigoplus_{i \leq n} \nu_i \oplus \nu.$$

$M \longrightarrow N$ by rule $(\|_{app})$. In this case we have $M \equiv (P\|Q)L$ and $N \equiv PL\|QL$ for some P, Q, L. We distinguish two cases, according to whether $\tau \leq \omega_1$ or $\tau = \tau \oplus \omega \neq \omega$ (the case of $\tau = \omega$ being trivial).

Case $\tau \leq \omega_1$: by 4.5.3*(vi)*

$$\exists I, \tau_i, \sigma.\, \Gamma \vdash P\|Q : \bigoplus_{i \in I} (\sigma \to \tau_i) \;\&\; \Gamma \vdash L : \sigma \;\&\; \tau = \bigoplus_{i \in I} \tau_i.$$

By 4.5.3*(x)*

$$\Gamma \vdash P\|Q : \bigoplus_{i \in I} (\sigma \to \tau_i) \Rightarrow \exists \mu, \nu.\, \Gamma \vdash P : \mu \;\&\; \Gamma \vdash Q : \nu \;\&\; \mu \wedge \nu \leq \bigoplus_{i \in I} (\sigma \to \tau_i).$$

It follows that $\mu \wedge \nu \leq \omega_1$, so that at least one type among μ and ν is less than ω_1 (notice that $\mu = \mu \oplus \omega$ and $\nu = \nu \oplus \omega$ imply $\mu \wedge \nu = (\mu \wedge \nu) \oplus \mu \oplus \nu \oplus \omega \not\leq \omega_1$). Assume $\mu \leq \omega_1$; then we distinguish among three subcases.

Subcase $\nu \leq \omega_1$:

$$\begin{array}{lcll}
 & \exists J, L, \mu_j, \nu_l.\, \Gamma \vdash P : \bigoplus_{j \in J} (\sigma \to \mu_j) \;\& & & \\
 & \Gamma \vdash Q : \bigoplus_{l \in L} (\sigma \to \nu_l) \;\& & & \\
 & (\bigoplus_{j \in J} \mu_j) \wedge (\bigoplus_{l \in L} \nu_l) \leq \bigoplus_{i \in I} \tau_i & \text{by } \mathbf{(P5)} & (4.54) \\
\Rightarrow & \exists J, L, \mu_j, \nu_l.\, \Gamma \vdash PL : \bigoplus_{j \in J} \mu_j \;\& & & \\
 & \Gamma \vdash QL : \bigoplus_{l \in L} \nu_l & \text{by } (\to \mathrm{E}) & \\
\Rightarrow & \Gamma \vdash PL\|QL : (\bigoplus_{j \in J} \mu_j) \wedge (\bigoplus_{l \in L} \nu_l) & \text{by } (\wedge \mathrm{I}_{\|}) & \\
\Rightarrow & \Gamma \vdash PL\|QL : \tau & \text{by (4.54) and } (\leq). &
\end{array}$$

Subcase $\nu = \nu \oplus \omega \neq \omega$:

$$\begin{array}{lcll}
 & \exists J, L, \mu_j, \nu_l.\, \Gamma \vdash P : \bigoplus_{j \in J} (\sigma \to \mu_j) \;\& & & \\
 & \Gamma \vdash Q : \bigoplus_{l \in L} (\sigma \to \nu_l) \oplus \omega \;\& & & \\
 & (\bigoplus_{j \in J} \mu_j) \wedge [(\bigoplus_{l \in L} \nu_l) \oplus \omega] \leq \bigoplus_{i \in I} \tau_i & \text{by } \mathbf{(P6)} & (4.55) \\
\Rightarrow & \exists J, L, \mu_j, \nu_l.\, \Gamma \vdash PL : \bigoplus_{j \in J} \mu_j \;\& & & \\
 & \Gamma \vdash QL : \bigoplus_{l \in L} \nu_l \oplus \omega & \text{by } (\to \mathrm{E}) \text{ and } (\to \mathrm{E}\omega) & \\
\Rightarrow & \Gamma \vdash PL\|QL : (\bigoplus_{j \in J} \mu_j) \wedge [(\bigoplus_{l \in L} \nu_l) \oplus \omega] & \text{by } (\wedge \mathrm{I}_{\|}) & \\
\Rightarrow & \Gamma \vdash PL\|QL : \tau & \text{by (4.55) and } (\leq). &
\end{array}$$

Subcase $\nu = \omega$: immediate since then $\mu \wedge \nu = \mu$ so that $\Gamma \vdash PL : \tau$ and the statement follows by $(\wedge\, \mathrm{I}_{\|})$ since trivially $\Gamma \vdash QL : \omega$.

Case $\tau = \tau \oplus \omega$:

$$\begin{array}{lll} \Gamma \vdash (P\|Q)L : \tau \oplus \omega & \Rightarrow & \exists I, \tau_i, \sigma.\, \Gamma \vdash P\|Q : \bigoplus_{i\in I}(\sigma \to \tau_i) \oplus \omega \;\& \\ & & \Gamma \vdash L : \sigma \;\&\; \tau = \bigoplus_{i\in I} \tau_i \oplus \omega \qquad \text{by } 4.5.3\textit{(vii)} \end{array}$$

$\Gamma \vdash P\|Q : \bigoplus_{i\in I}(\sigma \to \tau_i) \oplus \omega$

$$\begin{array}{lll} \Rightarrow & \exists \mu, \nu.\, \Gamma \vdash P : \mu \;\&\; \Gamma \vdash Q : \nu \;\& & \\ & \mu \wedge \nu \leq \bigoplus_{i\in I}(\sigma \to \tau_i) \oplus \omega & \text{by } 4.5.3\textit{(x)} \\ \Rightarrow & \exists J, L, \mu_j, \nu_l.\, \Gamma \vdash P : \bigoplus_{j\in J}(\sigma \to \mu_j) \oplus \omega \;\& & \\ & \Gamma \vdash Q : \bigoplus_{l\in L}(\sigma \to \nu_l) \oplus \omega \;\& & \\ & [(\bigoplus_{j\in J} \mu_j) \wedge (\bigoplus_{l\in L} \nu_l)] \oplus \omega \leq \bigoplus_{i\in I} \tau_i \oplus \omega & \text{by } \mathbf{(P7)} \qquad (4.56) \\ \Rightarrow & \exists J, L, \mu_j, \nu_l.\, \Gamma \vdash PL : \bigoplus_{j\in J} \mu_j \oplus \omega \;\& & \\ & \Gamma \vdash QL : \bigoplus_{l\in L} \nu_l \oplus \omega & \text{by } (\to \mathrm{E}\omega) \\ \Rightarrow & \Gamma \vdash PL\|QL : (\bigoplus_{j\in J} \mu_j \oplus \omega) \wedge (\bigoplus_{l\in L} \nu_l \oplus \omega) & \text{by } (\wedge \mathrm{I}_{\|}) \\ \Rightarrow & \Gamma \vdash PL\|QL : \tau & \text{by (4.56) and } (\leq). \qquad \square \end{array}$$

The main result of the present section is that must convergence implies typability by ω_1 and may convergence implies typability by $\omega_1 \oplus \omega$. We will see in section 4.7 that also the converse is true, so that these types will completely characterize convergence properties of terms.

4.5.7. Corollary.

(i) $M \Downarrow^{must}$ *implies* $\vdash M : \omega_1$*;*
(ii) $M \Downarrow^{may} V$ *and* $\vdash V : \sigma$ *imply* $\vdash M : \sigma \oplus \omega$*;*
(iii) $M \Downarrow^{may}$ *implies* $\vdash M : \omega_1 \oplus \omega$.

Proof.
(i).

$$\begin{array}{llll} M \Downarrow^{must} & \Rightarrow & \exists V \in \mathbf{V}.\; M \stackrel{*}{\longrightarrow} V & \text{by } 4.4.5\textit{(i)} \\ & \Rightarrow & \vdash M : \omega_1 & \text{by } 4.5.4\textit{(i)} \text{ and } 4.5.6. \end{array}$$

(ii).

$$\begin{array}{llll} M \Downarrow^{may} V \;\&\; \vdash V : \sigma & \Rightarrow & M = V \text{ or } \exists N.\; M = V + N & \text{by } 4.4.5\textit{(ii)} \\ & \Rightarrow & \vdash M : \sigma \oplus \omega & \text{by 4.5.6 using } (\leq) \text{ if } M = V, \\ & & & \text{and } (\omega), (\oplus I_+) \text{ if } M = V + N. \end{array}$$

(iii). Immediate from *(ii)* and 4.5.4*(i)*. □

4.6. The Interpretation

Now we define formally how to interpret our language $\Lambda_{+\|}$ in D°. The correct notion of environment should send the call-by-value variables into $\uplus$-irreducible elements different from $\bot$. Call-by-name λ-abstractions are interpreted as functions in the usual way. For call-by-value λ-abstractions, we must take into account the set of $\uplus$-irreducible elements whose $\uplus$ is less than or equal to the current d, i.e. the set $C^\circ(d) = \Psi^{-1}(\mathcal{C}(\Psi(d)))$, where $\mathcal{C}(\ )$ is defined in 4.2.4 (ii). The interpretation of the syntactic operators is the expected one. This justifies the following definition.

4.6.1. DEFINITION. An *environment* is a map $\eta : \mathsf{Vn} \cup \mathsf{Vv} \to D^\circ$ such that $\eta(\mathsf{Vv}) \subseteq C^\circ(D^\circ) - \{\bot\}$. Call Env^{D° the set of environments over D°. Then the interpretation map $[\![\cdot]\!]^{D^\circ} : Env^{D^\circ} \times \Lambda_{+\|} \to D^\circ$ is defined as follows (writing $[\![\cdot]\!]$ for $[\![\cdot]\!]^{D^\circ}$ and omitting Ψ):

$$
\begin{array}{lcl}
[\![x]\!]_\eta & = & \eta(x) \\
[\![v]\!]_\eta & = & \eta(v) \\
[\![\lambda x.M]\!]_\eta & = & \{\!|\lambda d \in D^\circ.[\![M]\!]_{\eta[d/x]}|\!\} \\
[\![\lambda v.M]\!]_\eta & = & \{\!|\lambda d \in D^\circ.\bigsqcup_{C \in C^\circ(d)} \langle M, \eta, C\rangle |\!\} \\
[\![MN]\!]_\eta & = & [\![M]\!]_\eta \cdot [\![N]\!]_\eta \\
[\![M \| N]\!]_\eta & = & [\![M]\!]_\eta \sqcup [\![N]\!]_\eta \\
[\![M + N]\!]_\eta & = & [\![M]\!]_\eta * [\![N]\!]_\eta \\
[\![\coprod M]\!]_\eta & = & \coprod [\![M]\!]_\eta,
\end{array}
$$

where $\langle M, \eta, C\rangle =$ if $\bot \notin C$ then $*_{c \in C}\ [\![M]\!]_{\eta[c/v]}$ else $\bot$.

This is a good definition, as the functions $\lambda d \in D^\circ.[\![M]\!]_{\eta[d/x]}$ and $\lambda d \in D^\circ. \bigsqcup_{C \in C^\circ(d)} \langle (M, \eta, C) \rangle$ are both in $[D^\circ \to D^\circ]$, being defined in terms of continuous operators. This can be shown by an easy induction on M, which we omit.

We show how the interpretation $[\![\cdot]\!]^{D^\circ}$ agrees with the type assignment system. It is clear that both definitions of environments and of the interpretation map apply to $\mathcal{F}$ via Φ. Then $Env^{\mathcal{F}}$ will be the set of mappings sending variables into filters such that Vv is sent to $\mathcal{F}_C - \{\uparrow \omega\}$. In fact we have

$$\eta \in Env^{\mathcal{F}} \ \Leftrightarrow\ \Phi \circ \eta \in Env^{D^\circ}$$

If the previous definition is rephrased using the corresponding operators over $\mathcal{F}$, we get a notion of interpretation $[\![\cdot]\!]^{\mathcal{F}}_\eta$ such that for all $M \in \Lambda_{+\|}$

$$[\![M]\!]^{\mathcal{F}}_\eta \in \mathcal{F} \qquad \text{and} \qquad \Phi([\![M]\!]^{\mathcal{F}}_\eta) = [\![M]\!]^{D^\circ}_{\Phi \circ \eta}.$$

4.6.2. THEOREM. *For $\eta \in Env^{\mathcal{F}}$ and any basis Γ, define:*

$$\Gamma \models \eta \Leftrightarrow \forall \chi \in \mathsf{Vn} \cup \mathsf{Vv}.\ \Gamma(\chi) \in \eta(\chi).$$

Then for all $M \in \Lambda_{+\|}$:

$$[\![M]\!]^{\mathcal{F}}_\eta = \{\sigma \mid \exists \Gamma.\ \Gamma \models \eta\ \&\ \Gamma \vdash M : \sigma\}.$$

Proof. By induction on M. The case in which M is a variable is trivial.

$M \equiv \lambda x.L$: let $F = [\![\lambda x.L]\!]^{\mathcal{F}}_\eta$ and $f = \lambda d \in D^\circ.\Phi([\![L]\!]^{\mathcal{F}}_{\eta[\Phi^{-1}(d)/x]})$. Then

$$\Phi(F) = [\![\lambda x.L]\!]^{D^\circ}_{\Phi \circ \eta} = \{\!|\lambda d \in D^\circ.[\![L]\!]^{D^\circ}_{(\Phi \circ \eta)[d/x]}|\!\} = \{\!|f|\!\}.$$

$\sigma \in F$ if and only if $\uparrow \sigma \subseteq F$, i.e. by the isomorphism between $\mathcal{F}$ and D°, if and only if $(\sigma)^+ \in K(\{\!|f|\!\})$. If $(\sigma)^+ \in K(\{\!|f|\!\})$, then $(\sigma)^+ = \bigsqcup_{i \in I}(e_i \Rightarrow d_i)$ for I finite and $e_i, d_i \in K(D^\circ)$. By 4.3.14, there are σ_i, τ_i such that $e_i = (\sigma_i)^+$ and $d_i = (\tau_i)^+$, so that we deduce that, for all σ, $\Phi^{-1}(g) = \uparrow\sigma \subseteq F$ if and only if $\sigma = \bigwedge_{i\in I}\sigma_i \to \tau_i$ for some I and σ_i, τ_i such that $f((\sigma_i)^+) \sqsupseteq (\tau_i)^+$.

Since $f((\sigma_i)^+) = \Phi([\![L]\!]^{\mathcal{F}}_{\eta[\uparrow\sigma_i/x]})$, we have $f((\sigma_i)^+) \sqsupseteq (\tau_i)^+$ if and only if $\uparrow\tau_i \subseteq [\![L]\!]^{\mathcal{F}}_{\eta[\uparrow\sigma_i/x]}$, which implies $\tau_i \in [\![L]\!]^{\mathcal{F}}_{\eta[\uparrow\sigma_i/x]}$.

By induction hypothesis $\tau_i \in [\![L]\!]^{\mathcal{F}}_{\eta[\uparrow\sigma_i/x]}$ if and only if there is Γ_i such that $\Gamma_i \models \eta[\uparrow\sigma_i/x]$ and $\Gamma_i \vdash L : \tau_i$. Consider the equivalences:

$$\begin{array}{rcl} \exists\Gamma_i.\ \Gamma_i \models \eta[\uparrow\sigma_i/x]\ \&\ \Gamma_i \vdash L : \tau_i & \Leftrightarrow & \exists\Gamma'_i.\ \Gamma'_i \models \eta\ \&\ \Gamma'_i, x : \sigma_i \vdash L : \tau_i \\ & \Leftrightarrow & \exists\Gamma'_i.\ \Gamma'_i \models \eta\ \&\ \Gamma'_i \vdash \lambda x.L : \sigma_i \to \tau_i \end{array}$$

using 4.5.3*(iii)*. Then, defining Γ as $\Gamma(\chi) = \bigwedge_{i\in I}\Gamma'_i(\chi)$, we have $\Gamma \models \eta$ and $\Gamma \vdash \lambda x.L : \bigwedge_{i\in I}\sigma_i \to \tau_i = \sigma$, so that we conclude, as desired,

$$\sigma \in F \ \Leftrightarrow\ \exists\Gamma.\ \Gamma \models \eta\ \&\ \Gamma \vdash \lambda x.L : \sigma.$$

$M \equiv \lambda v.L$: let $F = [\![\lambda v.L]\!]^{\mathcal{F}}_\eta$, $g = \lambda c \in C^\circ(D^\circ).\Phi([\![L]\!]^{\mathcal{F}}_{\eta[\Phi^{-1}(c)/v]})$ and $f = \lambda d \in D^\circ.\bigsqcup_{C\in C^\circ(d)}$ (if $\bot \notin C$ then $*_{c\in C}\, g(c)$ else $\bot$). Then $\Phi(F) = \{\!|f|\!\}$.

Let $\sigma, e_i, d_i, \sigma_i, \tau_i$ be as in previous case. For all $i \in I$ we have again the condition $d_i \sqsubseteq f(e_i) = \bigsqcup_{C\in C^\circ(e_i)}($ if $\bot \notin C$ then $*_{c\in C}\ g(c)$ else $\bot)$. As this is a directed join and d_i is compact, we know that for all $i \in I$ there exists $C^{(i)} \in C^\circ(e_i)$ such that $d_i \sqsubseteq *_{c\in C^{(i)}}\, g(c)$ or $d_i \sqsubseteq \bot$.

Let $\bot \notin C^{(i)}$. Suppose that $C^{(i)} = \{c^{(i)}_1, \ldots, c^{(i)}_{k_i}\}$. Then for each $j \le k_i$ there exists $\mu^{(i)}_j \in \mathit{IType}$ such that $c^{(i)}_j = (\mu^{(i)}_j)^+$. This implies that $*_{j\le k_i}\, c^{(i)}_j \sqsubseteq e_i$ if and only if $\sigma_i \le \bigoplus_{j\le k_i}\mu^{(i)}_j$. It follows that

$$\begin{array}{rcl} d_i \sqsubseteq *_{c\in C^{(i)}}\, g(c) & \Leftrightarrow & \tau_i \in *_{j\le k_i}[\![L]\!]^{\mathcal{F}}_{\eta[\uparrow\mu^{(i)}_j/v]} \\ & \Leftrightarrow & \exists\nu^{(i)}_1, \ldots, \nu^{(i)}_{k_i}.\forall j \le k_i.\ \nu^{(i)}_j \in [\![L]\!]^{\mathcal{F}}_{\eta[\uparrow\mu^{(i)}_j/v]}\ \&\ \bigoplus_{j\le k_i}\nu^{(i)}_j \le \tau_i. \end{array}$$

By induction hypothesis, for all $j \le k_i$ we have

$$\exists\Gamma^{(i)}_j.\ \Gamma^{(i)}_j \models \eta[\uparrow\mu^{(i)}_j/v]\ \&\ \Gamma^{(i)}_j \vdash L : \nu^{(i)}_j,$$

which is equivalent to

$$\exists\bar\Gamma^{(i)}_j.\ \bar\Gamma^{(i)}_j \models \eta\ \&\ \bar\Gamma^{(i)}_j, v : \mu^{(i)}_j \vdash L : \nu^{(i)}_j.$$

This implies that, for all $i \in I$,

$$\exists\Gamma^{(i)}.\ \Gamma^{(i)} \models \eta\ \&\ \forall j \le k_i.\ \Gamma^{(i)}, v : \mu^{(i)}_j \vdash L : \nu^{(i)}_j.$$

which, since $\forall j \le k_j.\nu^{(i)}_j \le \omega_1$, is equivalent by 4.5.3*(iv)* to

$$\exists\Gamma^{(i)}.\ \Gamma^{(i)} \models \eta\ \&\ \Gamma^{(i)} \vdash \lambda v.L : \bigoplus_{j\le k_i}\mu^{(i)}_j \to \bigoplus_{j\le k_i}\nu^{(i)}_j.$$

The case $\bot \in C^{(i)}$ is easier. In fact we have $f = \{\!|\bot|\!\} \Rightarrow \{\!|\bot|\!\}$, i.e. $F = \uparrow\{\omega \to \omega\}$, and $\vdash \lambda v.L : \omega \to \omega$.

$M \equiv LN$: then, using the fact that Φ is a morphism of applicative structures, we have

$$\begin{array}{rcl} [\![LN]\!]^{\mathcal{F}}_{\eta} & = & \Phi^{-1}([\![LN]\!]^{D^\circ}_{\Phi\circ\eta}) \\ & = & \Phi^{-1}([\![L]\!]^{D^\circ}_{\Phi\circ\eta} \cdot [\![N]\!]^{D^\circ}_{\Phi\circ\eta}) \\ & = & [\![L]\!]^{\mathcal{F}}_{\eta} \cdot [\![N]\!]^{\mathcal{F}}_{\eta} \end{array}$$

where $\cdot$ is defined in 4.3.17*(i)*. The statement follows by the induction hypothesis, 4.5.3*(vi)* and 4.5.3*(vii)*.

$M \equiv L + N$: reasoning as in the previous cases, we have that

$$[\![L+N]\!]^{\mathcal{F}}_{\eta} = [\![L]\!]^{\mathcal{F}}_{\eta} * [\![N]\!]^{\mathcal{F}}_{\eta},$$

where $*$ is defined in 4.3.15. The statement follows by the induction hypothesis and 4.5.3*(viii)*.

$M \equiv L \| N$:

$$\begin{array}{rcl} [\![L\|N]\!]^{\mathcal{F}}_{\eta} & = & \Phi^{-1}([\![L\|N]\!]^{D^\circ}_{\Phi\circ\eta}) \\ & = & \Phi^{-1}([\![L]\!]^{D^\circ}_{\Phi\circ\eta} \vee [\![N]\!]^{D^\circ}_{\Phi\circ\eta}) \\ & = & [\![L]\!]^{\mathcal{F}}_{\eta} \mathbin{\bar{\cup}} [\![N]\!]^{\mathcal{F}}_{\eta} \end{array}$$

where $\bar{\cup}$ is defined in 4.3.12. The statement follows by the induction hypothesis and 4.5.3*(ix)*.

$M \equiv \coprod L$: again we have

$$[\![\coprod L]\!]^{\mathcal{F}}_{\eta} = \coprod [\![L]\!]^{\mathcal{F}}_{\eta},$$

where $\coprod$ is defined in 4.3.17*(ii)*. The statement follows by the induction hypothesis, and 4.5.3*(x)*. □

The interpretation of the parallel and non-deterministic features of our language is clarified by the following equalities, which are shown in the proof of previous Theorem.

$$[\![L+N]\!]^{\mathcal{F}}_{\eta} = [\![L]\!]^{\mathcal{F}}_{\eta} * [\![N]\!]^{\mathcal{F}}_{\eta}$$

$$[\![L\|N]\!]^{\mathcal{F}}_{\eta} = [\![L]\!]^{\mathcal{F}}_{\eta} \mathbin{\bar{\cup}} [\![N]\!]^{\mathcal{F}}_{\eta}.$$

As an immediate consequence of Theorem 4.6.2 we have the completeness of our type assignment system, i.e. that

$$\Gamma \vdash M : \sigma \text{ if and only if } \Gamma \models M : \sigma$$

where $\Gamma \models M : \sigma$ is defined in the standard way.

4.6.3. Corollary. *The structure $\mathcal{F} \simeq D^\circ$ is a model of the equivalence defined in 4.4.4 over $\Lambda^0_{+\|}$ via the interpretations $[\![\cdot]\!]^{\mathcal{F}}$ (or $[\![\cdot]\!]^{D^\circ}$), i.e.*

$$\forall M, N \in \Lambda^0_{+\|}.\ M = N \Rightarrow [\![M]\!]^{\mathcal{F}} = [\![N]\!]^{\mathcal{F}}$$

where the environments are omitted since they do not affect the meaning of closed terms.

Proof. Immediate by 4.5.6 and 4.6.2. □

Being $\mathcal{F}$ ordered by subset inclusion, a partial order $\sqsubseteq^{\mathcal{F}}$ (hence an equivalence $\simeq^{\mathcal{F}}$) is induced over $\Lambda_{+\|}$, namely

$$M \sqsubseteq^{\mathcal{F}} N \ \Leftrightarrow\ \forall \eta \in Env^{\mathcal{F}}.\ [\![M]\!]^{\mathcal{F}}_{\eta} \subseteq [\![N]\!]^{\mathcal{F}}_{\eta}.$$

In the following section we will prove that $\sqsubseteq^{\mathcal{F}}$ coincides with the operational preorder.

4.7. Adequacy and Completeness

In this section we will prove that the filter model exactly mirrors the operational semantics, i.e. that it is fully abstract. This means that:

- the filter model is adequate, that is it does not equate operationally distinct programs:
$$M \sqsubseteq^{\mathcal{F}} N \text{ implies } M \sqsubseteq^{\mathcal{O}} N \;;$$
- the filter model is complete, that is it reflects the operational distinctions:
$$M \sqsubseteq^{\mathcal{O}} N \text{ implies } M \sqsubseteq^{\mathcal{F}} N.$$

Key to this result is the converse of 4.5.7*(i)*, i.e.

$$\vdash M : \omega_1 \text{ implies } M\Downarrow^{must} . \tag{4.57}$$

4.7.1. Characteristic Terms

As in previous chapter, the key property on which the proof of full abstraction relies is that any compact element of $\mathcal{F}$, which is of the shape $\uparrow\sigma$, is λ-definable, since for all types σ there exists a characteristic (closed) term R_σ such that

$$\vdash \mathsf{R}_\sigma : \tau \quad \text{if and only if} \quad \sigma \leq \tau, \tag{4.58}$$

that is $[\![\mathsf{R}_\sigma]\!]^{\mathcal{F}} = \uparrow\sigma$. Such terms are constructed inductively together with test terms: to each type σ we associate a test term T_σ such that for all closed terms M:

$$\mathsf{T}_\sigma M\Downarrow^{must} \quad \text{if and only if} \quad \vdash M : \sigma. \tag{4.59}$$

The definition of characteristic terms and test terms finely reflects the duality between $\|$ and $+$, and their correspondence with $\wedge$ and $\oplus$, respectively. It is interesting to compare them with the characteristic terms of subsection 3.5.4.5.3 in chapter 3.

Let Ω be any unsolvable of degree 0 (typically $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$) and $\mathbf{I_V} \equiv \lambda v.v$ the call-by-value identity combinator. Then we define two families of terms $\{\mathsf{R}_\sigma\}_{\sigma\in Type}$ and $\{\mathsf{T}_\sigma\}_{\sigma\in Type}$ as follows.

4.7.1. Definition. The *characteristic terms* R_σ and the *test terms* T_σ are defined by induction on the number of symbols of σ.

$$\mathsf{R}_\sigma \equiv \begin{cases} \Omega & \text{if } \Theta(\sigma) = \{\omega\} \\ \lambda x.(\mathsf{T}_\mu x)\,\mathsf{R}_\nu & \text{if } \Theta(\sigma) = \{\mu \to \nu\} \\ \mathsf{R}_\mu \|\, \mathsf{R}_\nu & \text{if } \Theta(\sigma) = \{\mu \wedge \nu\} \\ \sum_{\sigma'\in\Theta(\sigma)} \mathsf{R}_{\sigma'} & \text{otherwise.} \end{cases}$$

$$\mathsf{T}_\sigma \equiv \begin{cases} \lambda x.\mathbf{I} & \text{if } \Theta(\sigma) = \{\omega\} \\ \lambda x.\mathbf{I_V}((\lambda v.\mathsf{T}_\nu(v\,\mathsf{R}_\mu))x) & \text{if } \Theta(\sigma) = \{\mu \to \nu\} \\ \lambda x.\mathbf{I_V}(\mathsf{T}_\mu\, x + \mathsf{T}_\nu\, x) & \text{if } \Theta(\sigma) = \{\mu \wedge \nu\} \\ \lambda x.\mathbf{I_V}(\sum_{\sigma'\in\Theta(\sigma)} \mathsf{T}_{\sigma'}(\coprod x)) & \text{if } \omega \in \Theta(\sigma) \neq \{\omega\} \\ \lambda x.\mathbf{I_V}((\lambda v.\|_{\sigma'\in\Theta(\sigma)}\mathsf{T}_{\sigma'} v)x + \sum_{\sigma'\in\Theta(\sigma)} \mathsf{T}_{\sigma'}(\coprod x)) & \text{otherwise.} \end{cases}$$

To see that this inductive definition is correct, let $|\sigma|$ be the number of symbols occurring in σ. A simple induction on σ shows that, if $\sigma' \in \Theta(\sigma)$ then $|\sigma'| \leq |\sigma|$, and that, if the cardinality of $\Theta(\sigma)$ is greater than 1, then this inequality is strict.

In all cases but $\sigma \equiv \omega$, T_σ has the shape $\lambda x.\mathbf{I_V} M$, for some M: so, for any N, either $\mathsf{T}_\sigma N$ must converge or it must diverge (i.e. it cannot be both may divergent and may convergent).

$\mathsf{T}_{\mu\to\nu}$ has an internal call-by-value abstraction besides that one in $\mathbf{I_V}$. Indeed, in view of (4.57) and (4.59), if the argument of $\mathsf{T}_{\mu\to\nu}$ has an arrow type, then it has also type ω_1, so that it is expected to be must convergent.

Concerning the definition of T_σ when $\Theta(\sigma)$ has more than one element, observe that, if x has type σ, then $\coprod x$ has type $\bigwedge_{\sigma'\in\Theta(\sigma)} \sigma'$. So for example, if $\sigma \equiv \mu \oplus \nu \oplus \omega$ and μ, ν are arrow types, we have

$$\mathsf{T}_{\mu\oplus\nu\oplus\omega}(\mathsf{R}_\mu + \mathsf{R}_\nu + \mathsf{R}_\omega) \xrightarrow{*} \mathbf{I_V}(\mathsf{T}_\mu(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\omega) + \mathsf{T}_\nu(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\omega) + \mathsf{T}_\omega(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\omega)),$$

which is a convergent term assuming (4.58) and (4.59). Now let us consider $\sigma \equiv (\mu \wedge \nu) \oplus \tau$, where μ, ν, τ are arrow types. In this case we have

$$\mathsf{T}_{(\mu\wedge\nu)\oplus\tau}((\mathsf{R}_\mu \| \mathsf{R}_\nu) + \mathsf{R}_\tau) \xrightarrow{*}$$

$$\begin{aligned}&\mathbf{I_V}(\mathsf{T}_{\mu\wedge\nu}(\mathsf{R}_\mu \| \mathsf{R}_\nu) \| \mathsf{T}_\tau(\mathsf{R}_\mu \| \mathsf{R}_\nu) + \mathsf{T}_{\mu\wedge\nu}\mathsf{R}_\tau \| \mathsf{T}_\tau \mathsf{R}_\tau + \mathsf{T}_{\mu\wedge\nu}(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\tau) \| \mathsf{T}_\tau(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\tau) + \\ &+ \mathsf{T}_{\mu\wedge\nu}(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\tau) + \mathsf{T}_\tau(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\tau)),\end{aligned}$$

which also converges. To understand the use of the subterm $(\lambda v.\|_{\sigma'\in\Theta(\sigma)} \mathsf{T}_{\sigma'} v)x$, consider that by dropping it we would obtain:

$$\mathsf{T}_{(\mu\wedge\nu)\oplus\tau}(\mathsf{R}_\mu + \mathsf{R}_\nu + \mathsf{R}_\tau) \xrightarrow{*} \mathbf{I_V}(\mathsf{T}_{\mu\wedge\nu}(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\tau) + \mathsf{T}_\tau(\mathsf{R}_\mu \| \mathsf{R}_\nu \| \mathsf{R}_\tau)),$$

which converges, in spite of the fact that $\mathsf{R}_\mu + \mathsf{R}_\nu + \mathsf{R}_\tau$ does not have in general the type $(\mu\wedge\nu)\oplus\tau$. Lastly, to justify the subterms with the coproduct, notice that

$$(\lambda v.(\mathsf{T}_\mu v \| \mathsf{T}_\nu v))(\mathsf{R}_\mu + \mathsf{R}_\mu) \xrightarrow{*} \mathsf{T}_\mu \mathsf{R}_\mu \| \mathsf{T}_\nu \mathsf{R}_\mu,$$

which converges, while $\mathsf{R}_\mu + \mathsf{R}_\mu$ cannot be typed by $\mu \oplus \nu$ whenever $\mu \not\leq \nu$.

For the following proof it is useful to consider the sets of types of $\mathbf{I}$ and $\mathbf{I_V}$. Let us define $\mathcal{I}$ as the least filter including all types of the shape $\sigma \to \sigma$.

4.7.2. Proposition.

(i) $\vdash \mathbf{I} : \tau$ if and only if $\tau \in \mathcal{I}$:

(ii) $\sigma, \tau \in \mathcal{I}$ if and only if $\sigma \oplus \tau \in \mathcal{I}$;

(iii) $\sigma \to \tau \in \mathcal{I}$ if and only if $\sigma \leq \tau$;

(iv) $\vdash \mathbf{I_V} : \sigma \to \tau$ and $\tau \neq \omega$ imply $\sigma \leq \tau \wedge \omega_1$;

(v) $\Gamma \vdash \mathbf{I_V} M : \tau$ and $\tau \neq \omega$ imply $\Gamma \vdash M : \tau \wedge \omega_1$.

Proof.

(i) follows from 4.5.3*(i)*.

(ii). ($\Rightarrow$). Notice that $\sigma \wedge \tau \leq \sigma \oplus \tau$. ($\Leftarrow$) follows from *(i)* and 4.5.4*(ii)*.

(iii). Easy.

(iv) follows from 4.5.5*(i)* and 4.5.3*(i)*.

(v). By 4.5.4*(iii)* $\vdash \mathbf{I_V} : \sigma \to \tau$ and $\vdash M : \sigma$ for some σ. So we are done by *(iv)* and rule $(\leq)$. $\square$

4.7.3. LEMMA.

(i) $\vdash \mathsf{T}_\sigma : \tau \to \rho \oplus \omega$, *and* $\rho \leq \omega_1$ *imply* $\vdash \mathsf{T}_\sigma : \tau \to \rho \oplus \omega_1$*;*
(ii) $\vdash \mathsf{R}_\sigma : \tau$ *if and only if* $\sigma \leq \tau$*;*
(iii) *for all* $\tau \in Type$, $\vdash \mathsf{T}_\sigma : \sigma \to \tau \to \tau$*;*
(iv) $\vdash \mathsf{T}_\sigma : \tau \to \rho$, *and* $\rho \leq \omega_1$ *imply* $\tau \leq \sigma$, *and* $\rho \in \mathcal{I}$.

Proof. *(i)* If $\sigma \equiv \omega$ then $\mathsf{T}_\omega \equiv \lambda x.\mathbf{I}$, which has type $\tau \to \rho \oplus \omega$ if and only if $\vdash \mathbf{I} : \rho \oplus \omega$ by 4.5.3*(iii)*. This implies that $\vdash \mathbf{I} : \rho$ by 4.5.4*(ii)*. By hypothesis $\rho \leq \omega_1$, so that $\rho \leq \rho \oplus \omega_1$, and the statement follows.

In all other cases, using the fact that T_σ has the shape $\lambda x.\mathbf{I_V} M$, we have

$$\begin{array}{rcll}\vdash \mathsf{T}_\sigma : \tau \to \rho \oplus \omega & \Rightarrow & x : \tau \vdash \mathbf{I_V} M : \rho \oplus \omega & \text{by 4.5.3}\textit{(iii)} \\ & \Rightarrow & x : \tau \vdash M : (\rho \oplus \omega) \wedge \omega_1 & \text{by 4.7.2}\textit{(v)} \text{ and } \rho \leq \omega_1 \\ & \Rightarrow & x : \tau \vdash M : \rho \oplus \omega_1 & \text{by } (\leq) \\ & \Rightarrow & \vdash \mathsf{T}_\sigma : \tau \to \rho \oplus \omega_1 & \text{by } (\to \text{E}) \text{ and } (\to \text{I}_n), \\ & & & \text{since } \vdash \mathbf{I_V} : \rho \oplus \omega_1 \to \rho \oplus \omega_1.\end{array}$$

We prove *(ii)*-*(iv)* by a simultaneous induction on σ. Notice that all test terms are terms in $\mathbf{W}^0$. Therefore we will freely use the properties stated in 4.5.4. We consider only the most interesting cases.

$\Theta(\sigma) = \mu \to \nu$.

- (ii) $\mathsf{R}_{\mu\to\nu} \equiv \lambda x.(\mathsf{T}_\mu x)\,\mathsf{R}_\nu$ and suppose $\vdash \mathsf{R}_{\mu\to\nu} : \tau$. Then we proceed by a subordinate induction on the structure of τ, the base case being that $\tau \equiv \alpha \to \beta$, since $\mathsf{R}_{\mu\to\nu}$ is an abstraction (see 4.5.3*(ii)*).
  We consider two cases, $\beta \neq \beta \oplus \omega$ and $\beta = \beta \oplus \omega$.
  If $\beta \neq \beta \oplus \omega$ then

$$\begin{array}{lcll}\vdash \lambda x.(\mathsf{T}_\mu x)\,\mathsf{R}_\nu : \alpha \to \beta & & & \\ & \Rightarrow & x : \alpha \vdash (\mathsf{T}_\mu x)\,\mathsf{R}_\nu : \beta & \text{by 4.5.3}\textit{(iii)} \\ & \Rightarrow & \exists I, \beta_i, \delta.\ x : \alpha \vdash \mathsf{T}_\mu x : \bigoplus_{i\in I}(\delta \to \beta_i) & \\ & & \ \&\ \vdash \mathsf{R}_\nu : \delta\ \&\ \beta = \bigoplus_{i\in I}\beta_i & \text{by 4.5.3}\textit{(vi)} \\ & \Rightarrow & \vdash \mathsf{T}_\mu : \alpha \to \bigoplus_{i\in I}(\delta \to \beta_i)\ \&\ \nu \leq \delta & \text{by induction and 4.5.4}\textit{(iv)} \\ & \Rightarrow & \vdash \mathsf{T}_\mu : \alpha \to \bigoplus_{i\in I}(\nu \to \beta_i) & \text{by } (\leq) \\ & \Rightarrow & \alpha \leq \mu\ \&\ \bigoplus_{i\in I}(\nu \to \beta_i) \in \mathcal{I} & \text{by induction} \\ & \Rightarrow & \alpha \leq \mu\ \&\ \forall i \in I.\ \nu \to \beta_i \in \mathcal{I} & \text{by 4.7.2}\textit{(ii)} \\ & \Rightarrow & \alpha \leq \mu\ \&\ \forall i \in I.\ \nu \leq \beta_i & \text{by 4.7.2}\textit{(iii)} \\ & \Rightarrow & \alpha \leq \mu\ \&\ \nu \leq \beta & \text{since } \beta = \bigoplus_{i\in I}\beta_i \\ & \Rightarrow & \mu \to \nu \leq \alpha \to \beta. & \end{array}$$

  When $\beta = \beta \oplus \omega$ we have

$$\begin{array}{ll} & \exists I, \beta_i, \delta.\ x : \alpha \vdash \mathsf{T}_\mu x : \bigoplus_{i\in I}(\delta \to \beta_i) \oplus \omega\ \&\ \vdash \mathsf{R}_\nu : \delta\ \&\ \beta = \bigoplus_{i\in I}\beta_i \oplus \omega \text{ by 4.5.3}\textit{(vii)} \\ \Rightarrow & \vdash \mathsf{T}_\mu : \alpha \to \bigoplus_{i\in I}(\delta \to \beta_i) \oplus \omega\ \&\ \nu \leq \delta \text{ as above} \\ \Rightarrow & \vdash \mathsf{T}_\mu : \alpha \to \bigoplus_{i\in I}(\nu \to \beta_i) \oplus (\nu \to \omega) \text{ by } \textit{(i)} \text{ and } (\leq),\end{array}$$

and then the proof proceeds as in the previous case.
When $\tau \equiv \alpha \wedge \beta$ or $\alpha \oplus \beta$ the statement follows using the subordinate induction hypothesis and, in the last case, using 4.5.4*(ii)* which implies both $\vdash \mathsf{R}_{\mu\to\nu} : \alpha$ and $\vdash \mathsf{R}_{\mu\to\nu} : \beta$, being $\mathsf{R}_{\mu\to\nu} \in \mathbf{W}$.

- (iii) Being $\tau \to \tau \in IType$, we have that for all τ

$$\vdash \mathbf{I_V} : (\tau \to \tau) \to (\tau \to \tau). \tag{4.60}$$

By induction hypothesis, $\vdash \mathsf{T}_\nu : \nu \to \tau \to \tau$ and $\vdash \mathsf{R}_\mu : \mu$; therefore, being $\mu \to \nu \in IType$,

$$\dfrac{\dfrac{v : \mu \to \nu \vdash \mathsf{T}_\nu : \nu \to \tau \to \tau \qquad \dfrac{v : \mu \to \nu \vdash v : \mu \to \nu \quad v : \mu \to \nu \vdash \mathsf{R}_\mu : \mu}{v : \mu \to \nu \vdash v\mathsf{R}_\mu : \nu}\,(\to \mathrm{E})}{v : \mu \to \nu \vdash \mathsf{T}_\nu(v\mathsf{R}_\mu) : \tau \to \tau}\,(\to \mathrm{E})}{\vdash \lambda v.\mathsf{T}_\nu(v\mathsf{R}_\mu) : (\mu \to \nu) \to \tau \to \tau}\,(\to \mathrm{I}_v)$$

By this and rule $(\to\ \mathrm{E})$ we have $x : \mu \to \nu \vdash (\lambda v.\mathsf{T}_\nu(v\mathsf{R}_\mu))x : \tau \to \tau$, and the statement follows by (4.60), rule $(\to\ \mathrm{E})$ and rule $(\to\ \mathrm{I}_n)$.

- (iv) $\mathsf{T}_{\mu\to\nu} \equiv \lambda x.\mathbf{I_V}((\lambda v.\mathsf{T}_\nu(v\mathsf{R}_\mu))x)$ and suppose that $\vdash \mathsf{T}_{\mu\to\nu} : \tau \to \rho$, $\rho \leq \omega_1$. Then we have

$$\begin{array}{lll}
 & x : \tau \vdash \mathbf{I_V}((\lambda v.\mathsf{T}_\nu(v\mathsf{R}_\mu))x) : \rho & \\
\Rightarrow & x : \tau \vdash (\lambda v.\mathsf{T}_\nu(v\mathsf{R}_\mu))x : \rho & \text{by 4.7.2}\mathit{(v)} \\
\Rightarrow & \vdash \lambda v.\mathsf{T}_\nu(v\mathsf{R}_\mu) : \tau \to \rho & \text{by 4.5.4}\mathit{(iv)} \\
\Rightarrow & \exists I, \tau_i, \rho_i.\ \forall i \in I.\ v : \tau_i \vdash \mathsf{T}_\nu(v\mathsf{R}_\mu) : \rho_i\ \& & \\
 & \tau = \bigoplus_{i\in I} \tau_i\ \&\ \rho = \bigoplus_{i\in I} \rho_i & \text{by 4.5.3}\mathit{(iv)} \\
\Rightarrow & \forall i \in I. \exists \alpha_i.\ \vdash \mathsf{T}_\nu : \alpha_i \to \rho_i\ \&\ v : \tau_i \vdash v\mathsf{R}_\mu : \alpha_i & \text{by 4.5.4}\mathit{(iii)} \\
\Rightarrow & \forall i \in I. \exists \alpha_i.\ \alpha_i \leq \nu\ \&\ \tau_i \leq \mu \to \alpha_i\ \&\ \rho_i \in \mathcal{I} & \text{by induction and 4.5.4}\mathit{(v)} \\
\Rightarrow & \forall i \in I.\ \tau_i \leq \mu \to \nu\ \&\ \rho \in \mathcal{I} & \text{by 4.7.2}\mathit{(ii)} \text{ as } \rho = \bigoplus_{i\in I} \rho_i \\
\Rightarrow & \tau \leq \mu \to \nu\ \&\ \rho \in \mathcal{I} & \text{since } \tau = \bigoplus_{i\in I} \tau_i.
\end{array}$$

$\Theta(\sigma) = \{\mu \wedge \nu\}$.

- (iv) $\mathsf{T}_{\mu\wedge\nu} \equiv \lambda x.\mathbf{I_V}(\mathsf{T}_\mu x + \mathsf{T}_\nu x)$ and suppose that $\vdash \mathsf{T}_{\mu\wedge\nu} : \tau \to \rho$, $\rho \leq \omega_1$. Then we have

$$\begin{array}{llll}
 & x : \tau \vdash \mathbf{I_V}(\mathsf{T}_\mu x + \mathsf{T}_\nu x) : \rho & \text{by 4.5.3}\mathit{(iii)} & \\
\Rightarrow & x : \tau \vdash \mathsf{T}_\mu x + \mathsf{T}_\nu x : \rho & \text{by 4.7.2}\mathit{(v)} & \\
\Rightarrow & \exists \alpha, \beta.\ x : \tau \vdash \mathsf{T}_\mu x : \alpha\ \&\ x : \tau \vdash \mathsf{T}_\nu x : \beta\ \&\ \alpha \oplus \beta \leq \rho & \text{by 4.5.3}\mathit{(viii)} & (4.61) \\
\Rightarrow & \exists \alpha, \beta.\ \vdash \mathsf{T}_\mu : \tau \to \alpha\ \&\ \vdash \mathsf{T}_\nu : \tau \to \beta & \text{by 4.5.4}\mathit{(iv)} & \\
\Rightarrow & \exists \alpha, \beta.\ \tau \leq \mu\ \&\ \tau \leq \nu\ \&\ \alpha, \beta \in \mathcal{I} & \text{by induction} & \\
\Rightarrow & \tau \leq \mu \wedge \nu\ \&\ \rho \in \mathcal{I} & \text{by (4.61) and 4.7.2}\mathit{(ii)}. &
\end{array}$$

$\omega \in \Theta(\sigma) \neq \{\omega\}$.

- (iii) By derivability of rule $(\oplus \wedge \coprod)$ we know that $x : \sigma \vdash \coprod x : \sigma'$ for all $\sigma' \in \Theta(\sigma)$; therefore

$$\dfrac{\dfrac{\dfrac{\text{ind. hyp.}}{x:\sigma\vdash \mathsf{T}_{\sigma'}:\sigma'\to\tau\to\tau}\quad x:\sigma\vdash \coprod x:\sigma'\quad \forall\sigma'\in\Theta(\sigma)}{x:\sigma\vdash \mathsf{T}_{\sigma'}(\coprod x):\tau\to\tau\quad \forall\sigma'\in\Theta(\sigma)}\,(\to \mathrm{E})}{x:\sigma\vdash \sum_{\sigma'\in\Theta(\sigma)}\mathsf{T}_{\sigma'}(\coprod x):\tau\to\tau}\,(\oplus\mathrm{I}_+)$$

So we conclude using *(4.60)*, $(\to \mathrm{E})$, and $(\to \mathrm{I}_n)$.

- (iv) Let $\Theta(\sigma)=\{\sigma_i \mid i\in I\}$.

$$\begin{array}{llll}
\vdash \mathsf{T}_\sigma:\tau\to\rho & \Rightarrow & x:\tau\vdash \mathbf{I_V}(\sum_{i\in I}\mathsf{T}_{\sigma_{\mathsf{i}}}(\coprod x)):\rho & \text{by 4.5.3}\textit{(iii)}\\
& \Rightarrow & x:\tau\vdash \sum_{i\in I}\mathsf{T}_{\sigma_{\mathsf{i}}}(\coprod x):\rho & \text{by 4.7.2}\textit{(v)}\text{ since } \rho\le\omega_1\\
& \Rightarrow & \forall i\in I\ \exists\mu_i.\ x:\tau\vdash \mathsf{T}_{\sigma_{\mathsf{i}}}(\coprod x):\mu_i\ \&& \\
& & \qquad\qquad \bigoplus_{i\in I}\mu_i\le\rho & \text{by 4.5.3}\textit{(viii)}\\
& \Rightarrow & \forall i\in I\ \exists\mu_i,\nu_i.\ x:\tau\vdash \mathsf{T}_{\sigma_{\mathsf{i}}}:\nu_i\to\mu_i\ \&& \\
& & \qquad\qquad x:\tau\vdash \coprod x:\nu_i & \text{by 4.5.4}\textit{(iii)}\\
& \Rightarrow & \forall i\in I\ .\nu_i\le\sigma_i\ \&\ \mu_i\in\mathcal{I} & \text{by induction}\\
& \Rightarrow & \forall i\in I\ .x:\tau\vdash \coprod x:\sigma_i\ \&\ \rho\in\mathcal{I} & \text{by 4.7.2}\textit{(ii)}.
\end{array}$$

Now $x:\tau\vdash \coprod x:\sigma_i$ implies by 4.5.3*(x)* that there are $J_i,\tau_j^{(i)}$ such that

$$\tau=\bigoplus_{j\in J_i}\tau_j^{(i)}\quad\text{and}\quad \bigwedge_{j\in J_i}\tau_j^{(i)}\le\sigma_i. \tag{4.62}$$

Observe that $\omega\in\Theta(\sigma)$ implies that $\sigma=\sigma\oplus\omega$, so that

$$\begin{array}{lll}
\tau & = & \bigoplus_{i\in I}(\bigoplus_{j\in J_i}\tau_j^{(i)})\\
& = & [\bigwedge_{i\in I}(\bigwedge_{j\in J_i}\tau_j^{(i)})]\oplus\bigoplus_{i\in I}\bigoplus_{j\in J_i}\tau_j^{(i)} \quad \text{by } \mathbf{(P0)}\\
& \le & [\bigwedge_{i\in I}(\bigwedge_{j\in J_i}\tau_j^{(i)})]\oplus\omega\\
& \le & (\bigwedge_{i\in I}\sigma_i)\oplus\omega \quad \text{by (4.62)}\\
& = & \sigma \quad \text{by } \mathbf{(P3)}.
\end{array}$$

$\omega\notin\Theta(\sigma)$.

- (iii) Because of $\omega\notin\Theta(\sigma)$, for all $\sigma'\in\Theta(\sigma)$, $v:\sigma'$ is a basis. Hence

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{ind. hyp.}}{x:\sigma,v:\sigma'\vdash \mathsf{T}_{\sigma'}:\sigma'\to\tau\to\tau}\quad v:\sigma'\vdash v:\sigma'}{x:\sigma,v:\sigma'\vdash \mathsf{T}_{\sigma'}v:\tau\to\tau\quad\forall\sigma'\in\Theta(\sigma)}\,(\to\mathrm{E})}{x:\sigma,v:\sigma'\vdash \|_{\sigma'\in\Theta(\sigma)}\mathsf{T}_{\sigma'}v:\tau\to\tau\quad\forall\sigma'\in\Theta(\sigma)}\,(\wedge\mathrm{I}_{\|}),(\le)}{x:\sigma\vdash\lambda v.\|_{\sigma'\in\Theta(\sigma)}\mathsf{T}_{\sigma'}v:\sigma\to\tau\to\tau\quad x:\sigma\vdash x:\sigma}\,(\to\mathrm{I}_v)}{x:\sigma\vdash(\lambda v.\|_{\sigma'\in\Theta(\sigma)}\mathsf{T}_{\sigma'}v)x:\tau\to\tau}\,(\to\mathrm{E})$$

Moreover we can derive $x:\sigma\vdash\sum_{\sigma'\in\Theta(\sigma)}\mathsf{T}_{\sigma'}(\coprod x):\tau\to\tau$ as in previous case. Now the statement follows applying rules $(\oplus\mathrm{I}_+)$, $(\to\mathrm{E})$, $(\to\mathrm{I}_n)$, and using (4.60).

- (iv) Let $\Theta(\sigma) = \{\sigma_i \mid i \in I\}$. As in the proof of previous case we get

$$x : \tau \vdash (\lambda v.\|_{i\in I}\mathsf{T}_{\sigma_i} v)x + \sum_{i\in I} \mathsf{T}_{\sigma_i}(\coprod x)) : \rho.$$

By 4.5.3*(viii)*, this judgment implies that there exist μ and μ_i such that

$$x : \tau \vdash (\lambda v.\|_{i\in I}\mathsf{T}_{\sigma_i} v)x : \mu \tag{4.63}$$

$$\forall i \in I.\ x : \tau \vdash \mathsf{T}_{\sigma_i}(\coprod x) : \mu_i \tag{4.64}$$

$$\mu \oplus \bigoplus_{i\in I} \mu_i \leq \rho. \tag{4.65}$$

As above from (4.64) we have $\mu_i \in \mathcal{I}$ and (4.62). Moreover we get

$$\begin{array}{lll} (4.63) \Rightarrow & \vdash \lambda v.\|_{i\in I}\mathsf{T}_{\sigma_i} v : \tau \to \mu & \text{by } 4.5.4\textit{(iv)} \\ \Rightarrow & \exists H, \tau_h, \nu_h.\tau = \bigoplus_{h\in H} \tau_h \ \&\ \mu = \bigoplus_{h\in H} \nu_h \& & (4.66) \\ & \forall h \in H.\ v : \tau_h \vdash \|_{i\in I}\mathsf{T}_{\sigma_i} v : \nu_h & \text{by } 4.5.3\textit{(iv)}. \end{array}$$

$$\tag{4.67}$$

$$\begin{array}{lll} v : \tau_h \vdash \|_{i\in I}\mathsf{T}_{\sigma_i} v : \nu_h & & \\ \Rightarrow & \forall i \in I\ \exists \xi_h^{(i)}.\ v : \tau_h \vdash \mathsf{T}_{\sigma_i} v : \xi_h^{(i)} \ \&\ \bigwedge_{i\in I} \xi_h^{(i)} \leq \nu_h & \text{by } 4.5.3\textit{(ix)} \\ \Rightarrow & \forall i \in I\ \exists \xi_h^{(i)}.\ \vdash \mathsf{T}_{\sigma_i} : \tau_h \to \xi_h^{(i)} & \text{by } 4.5.4\textit{(iv)}. \end{array}$$

Observing that (4.65) and $\rho \leq \omega_1$ imply $\mu \leq \omega_1$, from $\mu = \bigoplus_{h\in H} \nu_h$ and $\bigwedge_{i\in I} \xi_h^{(i)} \leq \nu_h$ we get $\bigwedge_{i\in I} \xi_h^{(i)} \leq \omega_1$. This gives $\exists i_h \in I.\ \xi_h^{(i_h)} \leq \omega_1$. In fact, if we assume $\forall i \in I.\ \xi_h^{(i)} = \xi_h^{(i)} \oplus \omega$, we have $\bigwedge_{i\in I} \xi_h^{(i)} = (\bigwedge_{i\in I} \xi_h^{(i)}) \oplus \omega$ by **(P3)**. Moreover by *(i)* we have that $\xi_h^{(i)} = \xi_h^{(i)} \oplus \omega$ implies $\xi_h^{(i)} = \omega$. So by induction

$$\forall h \in H \exists i_h \in I.\ \tau_h \leq \sigma_{i_h} \ \&\ \xi_h^{(i_h)} \in \mathcal{I}. \tag{4.68}$$

Therefore from $\forall h \in H, \exists i_h \in I.\ \xi_h^{(i_h)} \in \mathcal{I}$, $\bigwedge_{i\in I} \xi_h^{(i)} \leq \nu_h$, $\forall h \in H, \forall i \in I$ either $\xi_h^{(i)} \leq \omega_1$ or $\xi_h^{(i)} = \omega$, $\mu = \bigoplus_{h\in H} \nu_h$, $\forall i \in I.\ \mu_i \in \mathcal{I}$, and *(4.65)*, we conclude $\rho \in \mathcal{I}$. Lastly

$$\begin{array}{rcll} \tau & = & \bigoplus_{i\in I}(\bigoplus_{j\in J_i} \tau_j^{(i_h)}) & \text{by } \textit{(4.62)} \\ & = & [\bigwedge_{i\in I}(\bigwedge_{j\in J_i} \tau_j^{(i_h)})] \oplus \tau & \text{by } \textbf{(PO)} \\ & = & [\bigwedge_{i\in I}(\bigwedge_{j\in J_i} \tau_j^{(i_h)})] \oplus \bigoplus_{h\in H} \tau_h & \text{by } \textit{(4.66)} \\ & \leq & (\bigwedge_{i\in I} \sigma_i) \oplus \bigoplus_{h\in H} \sigma_{i_h} & \text{by } \textit{(4.62)} \text{ and } \textit{(4.68)} \\ & \leq & \sigma & \text{by } \textbf{(P2)}. \end{array}$$

□

4.7.2. Typability implies Convergence

Aim of this subsection is to prove that types completely characterize must and may convergence, i.e.

$$\vdash M : \omega_1 \ \Rightarrow\ M \Downarrow^{must} \qquad\qquad \vdash M : \omega_1 \oplus \omega \ \Rightarrow\ M \Downarrow^{may}$$

for all closed terms M. The $\Leftarrow$ were proved in 4.5.7*(i)*, 4.5.7*(iii)*.

The proof of $\Rightarrow$ requires a double induction on types and deductions. Following a standard methodology, we split this induction by introducing a "realizability interpretation" of types as sets of closed terms.

4.7.4. Definition (Realizability). We define the mapping $[\![\cdot]\!] : Type \to \mathcal{P}(\Lambda^0_{+\|})$ by induction:

(i) $[\![\omega]\!] = \Lambda^0_{+\|}$;

(ii) $M \in [\![\sigma \to \tau]\!]$ iff exists $V \approx U_1 + \cdots + U_n$, for some n and $U_1, \ldots, U_n \in \mathbf{U}^0$ such that $M \xrightarrow{*} V$, and for all $N \in [\![\sigma]\!]$ and $i \leq n :\ m(U_i)N \in [\![\tau]\!]$;

(iii) $M \in [\![\sigma \wedge \tau]\!]$ iff $M \in [\![\sigma]\!]$ and $M \in [\![\tau]\!]$;

(iv) $M \in [\![\sigma \oplus \tau]\!]$ iff for some $P, Q \in \Lambda^0_{+\|}$, $M \xrightarrow{*} P + Q$, and $P \in [\![\sigma]\!]$ & $Q \in [\![\tau]\!]$.

If M is open, let s be a map from term variables to $\Lambda^0_{+\|}$. Now define:

(i) $s \models_r \Gamma$ iff for all $\chi : \sigma \in \Gamma :\ s(\chi) \in [\![\sigma]\!]$;

(ii) $\Gamma \models_r M : \sigma$ iff $s(M) \in [\![\sigma]\!]$ for all s such that $s \models_r \Gamma$.

The correctness of this definition is due to the fact that all types are inhabited by some closed term; in fact we have $\models_r \mathsf{R}_\sigma : \sigma$ for all types σ.

The following Lemma states some key properties of our realizability interpretation.

4.7.5. Lemma.

(i) $M \in [\![\sigma]\!]$ *and* $N \xrightarrow{*} M$ *imply* $N \in [\![\sigma]\!]$*;*

(ii) $M, N \in [\![\sigma]\!]$ *implies* $M + N \in [\![\sigma]\!]$*;*

(iii) $M + N \in [\![\sigma]\!]$ *and* σ *irreducible imply* $M, N \in [\![\sigma]\!]$*;*

(iv) $M \in [\![\sigma]\!]$ *implies that for all closed* $N :\ M\|N \in [\![\sigma]\!]$*;*

(v) if $U \in \mathbf{U}^0$ *and* $m(U)N \in [\![\sigma]\!]$ *for some closed* N*, then* $UN \in [\![\sigma]\!]$*;*

(vi) $\sigma \leq \tau$ *implies* $[\![\sigma]\!] \subseteq [\![\tau]\!]$*;*

(vii) $M \in [\![\sigma]\!]$ *implies* $\coprod M \in [\![\sigma]\!]$ *;*

(viii) $M \in [\![\sigma]\!]$ *and* $\sigma \neq \omega$*, irreducible imply that for some* $V \approx \sum_{i \leq n} U_i$*, where* $U_1, \ldots, U_n \in \mathbf{U}^0$*,* $M \xrightarrow{*} V$ *and* $m(U_i) \in [\![\sigma]\!]$ *for all* $i \leq n$*.*

Proof. All proofs but that of *(vi)* are by induction on σ.

(i) is straightforward from Definition 4.7.4.

(ii) The only interesting case is:

$\sigma \equiv \tau \oplus \rho$**.** If $M, N \in [\![\tau \oplus \rho]\!]$ then for some $M_0, N_0 \in [\![\tau]\!]$ and $M_1, N_1 \in [\![\rho]\!]$ it is the case that $M \xrightarrow{*} M_0 + M_1$ and $N \xrightarrow{*} N_0 + N_1$. By induction hypothesis $M_0 + N_0 \in [\![\tau]\!]$ and $M_1 + N_1 \in [\![\rho]\!]$; therefore

$$M + N \xrightarrow{*} (M_0 + N_0) + (M_1 + N_1) \in [\![\tau \oplus \rho]\!].$$

(iii) If $\sigma \equiv \tau \to \rho$ then $M + N \xrightarrow{*} V$ where $V \approx U_1 + \cdots + U_m + U_{m+1} + \cdots + U_{m+k}$, and $M \xrightarrow{*} U_1 + \cdots + U_m$ and $N \xrightarrow{*} U_{m+1} + \cdots + U_{m+k}$, say. Then $M, N \in [\![\sigma]\!]$ follows immediately by definition.

(iv) We consider the cases in which σ is an arrow or a sum.

$M \in \llbracket\tau \to \rho\rrbracket$ implies by definition that there are n and $U_1, \cdots, U_n \in \mathbf{U}^0$ such that $M \overset{*}{\longrightarrow} \sum_{i \le n} U_i$ and $\forall L \in \llbracket\tau\rrbracket$ and $i \le n$, $m(U_i)L \in \llbracket\rho\rrbracket$. We have $M\|N \overset{*}{\longrightarrow} \sum_{i \le n}(U_i\|N)$ by rule $(+\|)$ and $m(U_i\|N) \equiv m(U_i)\|m(N)$ or $m(U_i\|N) \equiv m(U_i)$ according to $N \in \mathbf{U}$ or not. In the first case $m(U_i)L \in \llbracket\rho\rrbracket$ implies $m(U_i)L\|m(N)L \in \llbracket\rho\rrbracket$ by induction, and therefore $m(U_i\|N)L \in \llbracket\rho\rrbracket$ by *(i)* since $m(U_i\|N)L \equiv (m(U_i)\|m(N))L \longrightarrow m(U_i)L\|m(N)L$. The second case is immediate.

$M \in \llbracket\tau \oplus \rho\rrbracket$ implies by definition that there are $P, Q \in \Lambda^0_{+\|}$ such that $M \overset{*}{\longrightarrow} P + Q$, $P \in \llbracket\tau\rrbracket$ and $Q \in \llbracket\rho\rrbracket$. We have $M\|N \overset{*}{\longrightarrow} P\|N + Q\|N$ by distributivity and $P\|N \in \llbracket\tau\rrbracket$, $Q\|N \in \llbracket\rho\rrbracket$ by induction.

(v) By Remark 4.4.1 $m(U)$ is either U itself, when $U \in \mathbf{W}^0$, or there exist $W \in \mathbf{W}^0$ and a closed $M \notin \mathbf{U}$ such that $U \approx W\|M$ and $m(U) \equiv W$. In the first case the statement is immediate; in the second one, if $WN \in \llbracket\sigma\rrbracket$ then $WN\|MN \in \llbracket\sigma\rrbracket$ by (iii), so that $(W\|M)N \in \llbracket\sigma\rrbracket$ by *(i)* because $(W\|M)N \longrightarrow WN\|MN$.

(vi) By induction on the axiomatic presentation of $\le$. We use *(ii)* for the case $\sigma \oplus \sigma \le \sigma$. The most interesting case is $(\nu \oplus \rho) \wedge \mu \le (\nu \wedge \mu) \oplus (\rho \wedge \mu)$. Now $M \in \llbracket\sigma\rrbracket$ implies $M \in \llbracket\nu \oplus \rho\rrbracket$ and $M \in \llbracket\mu\rrbracket$. From $M \in \llbracket\nu \oplus \rho\rrbracket$ we have $M \overset{*}{\longrightarrow} P + Q$ for some $P \in \llbracket\nu\rrbracket$ and $Q \in \llbracket\rho\rrbracket$. Therefore by *(iv)* we have $P\|M \in \llbracket\nu \wedge \mu\rrbracket$ and $Q\|M \in \llbracket\rho \wedge \mu\rrbracket$. So we can conclude $M \approx M\|M \overset{*}{\longrightarrow} (P + Q)\|M \longrightarrow P\|M + Q\|M \in \llbracket(\nu \wedge \mu) \oplus (\rho \wedge \mu)\rrbracket$.

(vii) We consider the case in which σ is a sum. $M \in \llbracket\tau \oplus \rho\rrbracket$ implies by definition that there are $P, Q \in \Lambda^0_{+\|}$ such that $M \overset{*}{\longrightarrow} P + Q$, $P \in \llbracket\tau\rrbracket$ and $Q \in \llbracket\rho\rrbracket$. Then $\coprod M \overset{*}{\longrightarrow} \coprod P \| \coprod Q$ by rule $(\coprod_+)$ and by induction $\coprod P \in \llbracket\tau\rrbracket$ and $\coprod Q \in \llbracket\rho\rrbracket$. Therefore we have $\coprod M \in \llbracket\tau \wedge \rho\rrbracket$ by *(iv)*, so we can conclude using *(vi)*: $\coprod M \in \llbracket\tau \oplus \rho\rrbracket$.

(viii) If $\sigma \equiv \tau \wedge \rho$ we have by induction $M \overset{*}{\longrightarrow} \sum_{i \in I} U_i$ and $M \overset{*}{\longrightarrow} \sum_{j \in J} U'_j$ for some some finite sets of indexes I, J and $m(U_i) \in \llbracket\tau\rrbracket$ and $m(U'_j) \in \llbracket\rho\rrbracket$. Now $M \approx M\|M \overset{*}{\longrightarrow} (\sum_{i \in I} U_i)\|(\sum_{j \in J} U'_j) \overset{*}{\longrightarrow} \sum_{i \in I}\sum_{j \in J}(U_i\|U'_j)$ and $m(U_i\|U'_j) \equiv m(U_i)\|m(U'_j) \in \llbracket\tau \wedge \rho\rrbracket$. $\square$

As expected realizability coincides with derivability in $\vdash$ and this implies that we can show that all closed terms typeable by ω_1 must converge.

4.7.6. THEOREM. *The assignment system is sound and complete with respect to the realizability interpretation, namely*

$$\Gamma \vdash M : \sigma \textit{ iff } \Gamma \models_r M : \sigma.$$

Proof. Soundness (that is $\Rightarrow$) is by induction on derivations. We consider just the more complex cases (the other being standard or similar).

Suppose that the derivation ends by

$$\frac{\Gamma, v : \sigma_i \vdash M : \tau_i \quad i = 1, \ldots, n}{\Gamma \vdash \lambda v.M : \bigoplus_{i \le n} \sigma_i \to \bigoplus_{i \le n} \tau_i}$$

where each σ_i is irreducible and different from ω. Since $\lambda v.M \in \mathbf{W}$ and it is irreducible, we have to show that, if $s \models_r \Gamma$, then $s(\lambda v.M)N \in \llbracket\bigoplus_{i \le n} \tau_i\rrbracket$ for all $N \in \llbracket\bigoplus_{i \le n} \sigma_i\rrbracket$. If $N \in \llbracket\bigoplus_{i \le n} \sigma_i\rrbracket$ then by 4.7.4 $N \overset{*}{\longrightarrow} \sum_{i \le n} N_i$ for some $N_1, \ldots, N_n$ such that $N_i \in \llbracket\sigma_i\rrbracket$ for all i. By Lemma

4.7.5*(viii)* for each i there exist $n_i \in \mathbb{N}$ and $U_{i,1}, \ldots, U_{i,n_i} \in \mathbf{U}$ such that $N_i = \sum_{j \leq n_i} U_{i,j}$ and $\forall j \leq n_i.\ m(U_{i,j}) \in [\![\sigma_i]\!]$ (1). Now:

$$(\lambda v.M)N \xrightarrow{*} (\lambda v.M)(\sum_{i\leq n}\sum_{j\leq n_i} U_{i,j}) \xrightarrow{*} \sum_{i\leq n}\sum_{j\leq n_i}((\lambda v.M)U_{i,j}) + Q \xrightarrow{*} \sum_{i\leq n}\sum_{j\leq n_i} P_{i,j} + Q'$$

where $Q \equiv (\lambda v.M)(\|_{i\leq n, j\leq n_i} U_{i,j})$, $Q' \equiv M[m(\|_{i\leq n,j\leq n_i} U_{i,j})/v]\|(\lambda v.M)V$ for some V such that $\|_{i\leq n,j\leq n_i} U_{i,j} \longrightarrow V$, and

(a) $P_{i,j} \equiv M[U_{i,j}/v]$ if $U_{i,j} \in \mathbf{W}^0$ (in this case $m(U_{i,j}) \equiv U_{i,j}$)

(b) $P_{i,j} \equiv M[m(U_{i,j})/v]\|(\lambda v.M)V_{i,j}$ if $U_{i,j} \longrightarrow V_{i,j}$.

Let $s_{i,j} = s[m(U_{i,j})/v]$, then by (1) and the induction hypothesis $s_{i,j}(M) \in [\![\tau_i]\!]$. In case (a) $s(P_{i,j}) \equiv s_{i,j}(M)$, so $s(P_{i,j}) \in [\![\tau_i]\!]$. In case (b) $s(P_{i,j}) \equiv s_{i,j}(M)\|s((\lambda v.M)V_{i,j})$, and again $s(P_{i,j}) \in [\![\tau_i]\!]$ by 4.7.5*(iv)*. For Q', notice that $m(\|_{i\leq n,j\leq n_i} U_{i,j}) \equiv \|_{i\leq n,j\leq n_i} m(U_{i,j})$, and therefore by (1) and 4.7.5*(iv)* we have $m(\|_{i\leq n,j\leq n_i} U_{i,j}) \in [\![\sigma_i]\!]$ for all $i \leq n$. Let $s' = s[m(\|_{i\leq n,j\leq n_i} U_{i,j})/v]$, then, again by induction hypothesis $s'(M) \in [\![\tau_i]\!]$, so that, by 4.7.5*(iv)* $s(Q') \equiv s'(M)\|s((\lambda v.M)V) \in [\![\tau_i]\!]$ for all $i \leq n$. We finally conclude, by 4.7.5*(ii)* and 4.7.5*(i)* that $(\lambda v.M)N \in [\![\bigoplus_{i\leq n} \tau_i]\!]$ as desired.

If the derivation ends by

$$\frac{\Gamma \vdash M : \bigoplus_{i\leq n}(\sigma \to \tau_i) \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \bigoplus_{i\leq n} \tau_i}$$

choose any s such that $s \models_r \Gamma$. By induction $s(M) \in [\![\bigoplus_{i\leq n}(\sigma \to \tau_i)]\!]$, i.e. $s(M) \xrightarrow{*} \sum_{i\leq n} M_i$ for some $M_1, \ldots, M_n$ such that $M_i \in [\![\sigma \to \tau_i]\!]$. Again by induction $s(N) \in [\![\sigma]\!]$. Now, by definition, each $M_i \xrightarrow{*} \sum_{j_i \in J_i} U_{j_i}$ such that, for all $j_i \in J_i$, $m(U_{j_i})N \in [\![\tau_i]\!]$. By 4.7.5*(v)* this implies that $U_{j_i}N \in [\![\tau_i]\!]$ for all i and $j_i \in J_i$, so that $M_i s(N) \in [\![\tau_i]\!]$, i.e. $s(MN) \in [\![\bigoplus_{i\leq n} \tau_i]\!]$ by 4.7.5*(i)* since $s(MN) \xrightarrow{*} \sum_{i\leq n} M_i s(N)$.

To prove completeness (namely $\Leftarrow$) first we observe that, given Γ, Γ', such that $FV(\Gamma) \cap FV(\Gamma') = \emptyset$ the map $s_{\Gamma,\Gamma'}$ defined by

$$s_{\Gamma,\Gamma'}(\chi) = \begin{cases} \chi & \text{if } \chi \in FV(\Gamma') \\ \mathsf{R}_\tau & \text{if } \chi : \tau \in \Gamma \\ \mathsf{R}_{\omega\to\omega} & \text{if } \chi \notin FV(\Gamma \cup \Gamma') \text{ and } \chi \text{ is call-by-value} \\ \mathsf{R}_\omega & \text{otherwise} \end{cases}$$

is such that $s_{\Gamma,\emptyset} \models_r \Gamma$, hence by hypothesis $s_{\Gamma,\emptyset}(M) \in [\![\sigma]\!]$. It is easy to prove by induction on M that for all Γ, Γ', such that $FV(\Gamma) \cap FV(\Gamma') = \emptyset$:

$$\forall M, \sigma.\ \Gamma' \vdash s_{\Gamma,\Gamma'}(M) : \sigma \Rightarrow \Gamma \cup \Gamma' \vdash M : \sigma.$$

Now we prove:

$$\Gamma \models_r M : \sigma \Rightarrow \Gamma \vdash M : \sigma$$

by induction on σ. All cases are immediate but that of the arrow types. Let $\sigma \equiv \tau \to \rho$.

By the soundness part of this theorem we know that $\mathsf{R}_\tau \in [\![\tau]\!]$. By definition $s_{\Gamma,\emptyset}(M) \equiv M' \xrightarrow{*} \sum_{i\leq n} U_i$ for some n and $U_1, \ldots, U_n \in \mathbf{U}^0$ such that, for all $i \leq n$, $m(U_i)\mathsf{R}_\tau \in [\![\rho]\!]$. By induction hypothesis we have $\Gamma \vdash m(U_i)\mathsf{R}_\tau : \rho$. Since $m(U) \in \mathbf{W}^0$, by 4.5.4*(iii)* there exists τ' such that $\vdash m(U_i) : \tau' \to \rho$ and $\vdash \mathsf{R}_\tau : \tau'$. Now 4.7.3*(ii)* implies that $\tau \leq \tau'$, so that $\tau' \to \rho \leq \tau \to \rho$ and therefore, by rule $(\leq)$, we know that $\vdash m(U_i) : \tau \to \rho$ for all $i \leq n$. Using 4.5.4*(vi)* we also know that $\vdash U_i : \tau \to \rho$ for all i, so that, by rule (+I), $\vdash \sum_{i \leq n} U_i : \tau \to \rho$. Therefore $\vdash M' : \tau \to \rho$ by 4.5.6, which immediately implies by the above claim that $\Gamma \vdash M : \tau \to \rho$. □

The main result we obtain is the converse of 4.5.7, which easily follows from soundness, namely:

4.7.7. Corollary.
(i) $\vdash M : \omega_1$ *implies* $M \Downarrow^{must}$*;*
(ii) $\vdash M : \omega_1 \oplus \omega$ *implies* $M \Downarrow^{may}$*.*

As a byproduct of these results we get a proof of Proposition 4.4.6, namely that, if $M \Downarrow^{must}$ and $M \xrightarrow{*} N$ implies $N \Downarrow^{must}$ for closed M, N. Indeed, if $M \Downarrow^{must}$ then $\vdash M : \omega_1$ by 4.5.7(i). Since types are preserved by reduction (actually by conversion) by Theorem 4.5.6, we know that $\vdash N : \omega_1$. Therefore, by Corollary 4.7.7.*(i)*, we conclude that $N \Downarrow^{must}$ as desired. A similar argument, using 4.5.7(iii) and 4.7.7.*(ii)*, shows that if $M \Downarrow^{may}$ and $M \xrightarrow{*} N$ then $N \Downarrow^{may}$.

4.7.3. Full Abstraction Theorem

To establish the main result of this chapter we use the discriminability power of test terms.

4.7.8. Theorem. *Let M be a closed term.*

$$\mathsf{T}_\sigma M \Downarrow^{must} \quad \textit{iff} \ \vdash M : \sigma.$$

Proof.
$(\Rightarrow)$.

$$\begin{array}{lll} \mathsf{T}_\sigma M \Downarrow^{must} & \Rightarrow \ \vdash \mathsf{T}_\sigma M : \omega_1 & \text{by 4.5.7}\textit{(i)} \\ & \Rightarrow \ \exists \rho.\ \vdash_{\mathcal{L}} \mathsf{T}_\sigma : \rho \to \omega_1 \ \ \& \ \vdash M : \rho & \text{by 4.5.4.}\textit{(iii)} \text{ since } \mathsf{T}_\sigma \in \mathsf{TVal}^0 \\ & \Rightarrow \ \exists \rho.\ \rho \leq \sigma \ \ \& \ \vdash M : \rho & \text{by 4.7.3}\textit{(iv)} \\ & \Rightarrow \ \vdash M : \sigma & \text{by rule } (\leq). \end{array}$$

$(\Leftarrow)$.

$$\begin{array}{lll} \vdash M : \sigma & \Rightarrow \ \vdash \mathsf{T}_\sigma M : \omega_1 & \text{by 4.7.3}\textit{(iii)} \\ & \Rightarrow \ \mathsf{T}_\sigma M \Downarrow^{must} & \text{by 4.7.7.}\textit{(i)}. \end{array}$$

□

Adequacy is a consequence of 4.7.7 and of the fact that $\simeq^{\mathcal{F}}$ is a congruence. For the converse it suffices to observe that test terms discriminate internally, that is with respect to the convergence predicate, terms having different interpretations in the filter model.

4.7.9. Theorem (Full Abstraction Theorem).

$$M \sqsubseteq^{\mathcal{F}} N \ \textit{iff} \ M \sqsubseteq^{\mathcal{O}} N.$$

Proof. $(\Rightarrow)$ (Adequacy) Since $\sqsubseteq^{\mathcal{F}}$ is a precongruence, we immediately have that

$$M \sqsubseteq^{\mathcal{F}} N \Rightarrow \forall C[\ ].\ \vdash C[M] : \omega_1 \Rightarrow \Gamma \vdash C[N] : \omega_1.$$

It follows that

$$C[M]\Downarrow^{must} \Rightarrow \vdash C[M] : \omega_1 \text{ by } 4.5.7\textit{(i)} \Rightarrow \vdash C[N] : \omega_1 \Rightarrow C[N]\Downarrow^{must} \text{ by } 4.7.7.\textit{(i)}.$$

($\Leftarrow$) (Completeness)

$$M \not\sqsubseteq^{\mathcal{F}} N \;\Leftrightarrow\; \exists \Gamma, \sigma.\, \Gamma \vdash M : \sigma \;\&\; \Gamma \not\vdash N : \sigma.$$

Let $FV(MN) = \{\chi_i \mid 1 \leq i \leq n\}$, $\Gamma = \{\chi_i : \tau_i \mid 1 \leq i \leq n\}$ and $\tau = \tau_1 \to \cdots \to \tau_n \to \sigma$, then $\vdash \lambda\chi_1 \ldots \chi_n.M : \tau$ and $\not\vdash \lambda\chi_1 \ldots \chi_n.N : \tau$ by 4.5.3.*(iii)* and *(iv)*. Therefore choosing $C[\,] = \mathsf{T}_\tau(\lambda\chi_1 \ldots \chi_n.[\,])$ we have that $C[M]\Downarrow^{must}$ and $C[N]\Uparrow^{may}$ which imply $M \not\sqsubseteq^{\mathcal{O}} N$. $\square$

The full abstraction theorem shows that the calculus we have defined is a complete description of the powerdomain structure devised in section 4.2, being the match between the two provided by the logical system.

Chapter 5

Conclusion

The literature related to the present work has mostly been quoted in the introduction. A research direction, which has not yet been mentioned, is that initiated by Milner in [72], and developed by Sangiorgi in [87]. The idea is to consider the π-calculus as the basic model, and to study lazy λ-calculus via encodings into the π-calculus itself. A comparative study and a survey of these investigations is [62]. Although related, this approach seems orthogonal to that which is developed in the present thesis, where the concurrent λ-calculus is studied with respect to denotational models via a type assignment systems. In spite of this, as remarked at page 84, we conjecture that the theory of this encoding and that of the model of chapter 3 coincide.

It is interesting to compare our full abstraction result with the negative results of [58] and [90]. Both papers deal with typed λ-calculus (actually PCF).
Jim and Meyer show in [58] that any denotational semantics which is adequate for PCF, and in which a certain simple boolean functional exists, cannot be fully abstract for extensions of PCF satisfying the Context Lemma. This boolean functional is not Scott's continuous, but it is stably continuous. So it follows that there is no extension of PCF satisfying the Context Lemma for which the stable domains are fully abstract. Actually we consider Scott's continuous functions and moreover our calculus does not satisfy the Context Lemma. For example we distinguish between $\mathbf{I} + \Delta$ and $\lambda x.(x + xx)$, which clearly have the same applicative behavior.
Sieber [90] adds call-by-value and non-determinism to PCF. He explains why the non-deterministic extensions of call-by-name λ-calculus studied in [10, 11] need the power-domain functors only at ground types. Instead call-by-value and non-determinism require also power-domains for function types. Moreover Sieber shows that no fully abstract model of demonic non-determinism and call-by-value can be given in the typed case. The counterexample he considers, however, has no counterpart in the type-free setting (see the discussion at the end of section 3.5).

Finally in [36], [35] ideas coming from the present thesis are used to define a calculus of higher-order processes including communication primitives. For this calculus a program logic in the form of a type assignment system is introduced. The induced filter model turns out to be fully abstract with respect to the operational preorder. This is closely related, and should be compared, to [49].

Summarizing, the main achievement of the present study is the definition of three operational semantics which assess in a correct and effective way features like parallelism, non-determinism, higher-order functions, and call-by-value which have not received, to the knowledge of the author, a convincing treatment within a unique comprehensive system.

Moreover the abstract descriptions of the operational semantics by means of the logic equivalence, which are guaranteed by the full abstraction theorem, show that an elegant correspondence between the operators in the term syntax and the logical connectives is at the basis of the whole construction.

The logical models are simple distributive lattices enriched with an operation which interprets functional application. The simplicity of these structures is also remarkable.

References

[1] M. ABADI, "A Semantics for Static Type Inference in a Nondeterministic Language", *Info. and Comp* 109, 1994, 300-306.

[2] S. ABRAMSKY, "On Semantic Foundations for Applicative Multiprogramming" *LNCS* 154, Springer-Verlag, Berlin 1983, 1-14.

[3] ———, "Observation Equivalence as Testing Equivalence", *Theor. Comp. Sci.* 53, 1987, 225-241.

[4] ———, "The Lazy Lambda Calculus", *Research Topics in Functional Programming*, D.Turner ed., Addison-Wesley, Reading 1989, 65-116.

[5] ———, "Domain Theory in Logical Form", *Ann. of Pure and Appl. Logics*, 51, 1991, 1-77.

[6] S. ABRAMSKY, C.-H.L. ONG, "Full Abstraction in the Lazy Lambda Calculus", *Info. and Comp.* 105, 1993, 159-267.

[7] F. ALESSI, "Type Preorders", *LNCS* 787, Springer-Verlag, Berlin, 1994, 37-51.

[8] F. ALESSI, M. DEZANI-CIANCAGLINI, U. DE'LIGUORO, "May and Must Convergency in Concurrent λ-calculus", *LNCS* 841, Springer-Verlag, Berlin 1994, 211-220.

[9] ———, "A Convex Powerdomain over Lattices: its Logic and λ-calculi", Internal Report, Universitá di Torino 1996.

[10] E.A. ASHCROFT, M.C.B. HENNESSY, "A Mathematical Semantics for a Non-deterministic Typed Lambda Calculus", *Theor. Comp. Sci.* 11, 1980, 227-245.

[11] E. ASTESIANO, G. COSTA, "Non-Determinism and Fully Abstract Models", *R.A.I.R.O. Theor. Inf.* 14, 1980, 323-347.

[12] E. ASTESIANO, E. ZUCCA, "Parametric Channels via Label Expressions in CCS", *Theor. Comp. Sci.* 33, 1984, 45-63.

[13] S. VAN BAKEL, "Complete Restrictions of the Intersection Type Discipline", *Theor. Comp. Sci.* 102, 1992, 135-163.

[14] ———, "Principal Type Schemes for the Strict Type Assignment System", *J. Logic and Comp.* 3(6), 1993, 643-670.

[15] ——–, "Intersection Type Assignment Systems", *Theor. Comp. Sci.* 151 (2), 1995, 385-436.

[16] F. Barbanera, M. Dezani-Ciancaglini, U. de'Liguoro, "Intersection and Union Types: Syntax and Semantics", *Info. and Comp.* 119, 1995, 202-230.

[17] H.P. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*, 2nd ed., North-Holland, Amsterdam 1984.

[18] H.P. Barendregt, M. Coppo, M. Dezani-Ciancaglini, "A Filter Lambda Model and the Completeness of Type Assignment", *J. of Symbolic Logic* 48, 1983, 931-940.

[19] E.W. Beth, *The Foundation of Mathematics*, 2nd ed., North-Holland, Amsterdam 1965.

[20] G. Boudol, "Semantique Operationelle at Algebrique des Programmes Recursifs Non-Deterministes", Thèse d'Etat, Université de Paris VII, 1980.

[21] ———, "Toward a Lambda-calculus for Concurrent and Communicating Systems", TAP-SOFT'89, *LNCS* 351, Springer-Verlag, Berlin, 1989, 149-161.

[22] ———, "A Lambda Calculus for (Strict) Parallel Functions", *Info. and Comp.* 108, 1994, 51-127.

[23] G. Boudol, C.Laneve, "The Discriminating Power of Multiplicities in the λ-calculus", *Info. and Comp.* 126(1), 1996, 83-102.

[24] M. Coppo, M. Dezani-Ciancaglini, F. Honsell, G. Longo, "Extended Type Structures and Filter λ-models", *Logic Colloquium '82*, G. Lolli et al. eds., North-Holland, Amsterdam 1984, 241-262.

[25] M. Coppo, M. Dezani-Ciancaglini, G. Longo, "Applicative Information Systems", *LNCS* 159, Springer-Verlag, Berlin 1983, 35-64.

[26] M.Coppo, M. Dezani-Ciancaglini, B.Venneri, "Principal Type Schemes and λ-calculus Semantics", *To H.B.Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, New York, 1980, 535-560.

[27] M. Coppo, M. Dezani-Ciancaglini, G .Longo, "Applicative Information Systems", CAAP '83, *LNCS* 159, Springer-Verlag, Berlin 1983, 35-64.

[28] M.Coppo, A.Ferrari, "Type Inference, Abstract Interpretation and Strictness Analysis", *Theor. Comp. Sci.* 121, 1993, 113-144.

[29] B.A. Davey, H.A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, Cambridge 1990.

[30] R.De Nicola, M.Hennessy, "Testing Equivalence for Proceses", *Theor. Comp. Sci.* 34, 1984, 82-113.

[31] M. Dezani-Ciancaglini, U. de'Liguoro, A. Piperno, "Filter Models for a Parallel and Non-Deterministic λ-calculus", *LNCS* 711, Springer-Verlag, Berlin 1993, 403-412.

[32] ———, "Fully Abstract Semantics for Concurrent λ-calculus", *LNCS* 789, Springer-Verlag, Berlin 1994, 16-35.

[33] ———, "Filter Models for Conjunctive-Disjunctive λ-calculi", *Theor. Comp. Sci.* to appear.

[34] ———, "A Filter Model for Concurrent λ-calculi", *SIAM J. of Comp.* to appear.

[35] F. Damiani, M. Dezani-Ciancaglini, P. Giannini, "A Filter Model for Mobile Processes", Internal Report, Universitá di Torino 1996.

[36] M. Dezani-Ciancaglini, U. de'Liguoro, P. van Rossum, "A Type Assignment System for Higher-order Communicating Processes", Internal Report, Università di Torino, 1994.

[37] A. Giacalone, P. Mishra, S. Prasad, "FACILE: A Symmetric Integration of Concurrent and Functional Programming", *LNCS* 352, Springer-Verlag, Berlin 1989, 184-201.

[38] ———, "Operational and Algebraic Semantics for FACILE: A Symmetric Integration of Concurrent and Functional Programming", *LNCS* 443, Springer-Verlag, Berlin 1990, 765-779.

[39] G.K. Gierz, K.H. Hoffmann, K. Keimel, J.D. Mislove, D.S. Scott, *A Compendium of Continuous Lattices*, Springer-Verlag, Berlin 1980.

[40] G.Gonthier, J.J. Lévy, P.A.Melliés, "An Abstract Standardization Theorem", *LICS '92*, IEEE Computer Soc. Press, Silver Spring, 1992, 72-81.

[41] C. Gunter, "Relating partial and total correctness interpretations of non-deterministic programs", Proceedings of POPL'90, ACM, New York 1990, 306-319.

[42] C. Gunter, "The mixed powerdomain", *Theor. Comp. Sci.* 103, 1992, 311-334.

[43] C. Gunter, D. Scott, "Semantic Domains", in J. van Leeuwen ed., *Handbook of Theoretical Computer Sceince*, North-Holland, Amsterdam 1990, vol.B, 633-674.

[44] R. Heckmann, "Set Domains", *LNCS* 432, Springer-Verlag, Berlin 1990, 177-196.

[45] M.C.B. Hennessy, "The Semantics of Call-by-value and Call-by-name in a Non-Deterministic Environment", *SIAM J. of Comp.* 9, 1980, 67-84.

[46] ———, "Powerdomains and Non-Deterministic Recursive Definitions", *LNCS* 137, Springer-Verlag, Berlin 1982, 178-193.

[47] ———, *Algebraic Theory of Processes*, MIT Press, Cambridge Massachussets 1988.

[48] ———, "A Fully Abstract Denotational Model for Higher-Order Processes", *Info. and Comp.* 112, 1994, 55-95.

[49] ———, "Higher-order Processes and their Models", *LNCS* 820, Springer-Verlag, Berlin 1994, 286-303.

[50] M. Hennessy, G. Plotkin, "Full abstraction for a simple parallel programming language", MFCS '79, *LNCS* 74, Springer-Verlag, Berlin 1979, 108-120.

[51] R. Hindley, "The Principal Type Scheme of an Object in Combinatory Logic", *Trans. Amer. Math. Soc.* 146, 1969, 29-60.

[52] R. Hindley, "The Completeness Theorem for Typing λ-terms", *Theor. Comp. Sci.* 22, 1983, 1-17.

[53] ———,"The Simple Semantics for Coppo-Dezani-Sallé Type Assignment", *LNCS* 137, Springer-Verlag, Berlin 1982, 212-226.

[54] J.R. HINDLEY, G. LONGO, "Lambda-calculus Models and Extensionality", *Zeit. für Math. Logik* 26, 1980, 289-310.

[55] C.A.R. HOARE, *Communicating Sequential Processes*, Prentice-Hall, Englewood Cliffs, 1985.

[56] J.M.E.HYLAND"A Syntactic Characterization of the Equality in Some Models of the λ-calculus", *J.London Math.Soc.* (2), 12, 1976, 361-370.

[57] T.P.JENSEN, "Disjunctive Strictness Analysis", *LICS '92*, IEEE Comp. Soc. Press, Silver Spring, 1992, 174-185.

[58] T. JIM, A.R. MEYER, "Full Abstraction and the Context Lemma", *SIAM J. of Comp.*, to appear.

[59] P.T. JOHNSTONE, *Stone Spaces*, Cambridge University Press, Cambridge 1982.

[60] P.J. LANDIN, "The Mechanical Evaluation of Expressions", *The Computer Journal* 6, 1964, 308-320.

[61] K.G.LARSEN, G.WINSKELL, "Using Information Systems to Solve Recursive Domain Equations Effectively", *LNCS* 173, Springer-Verlag, Berlin 1984, 109-130.

[62] C. LAVATELLI, "Non-Deterministic Lazy λ-calculus versus π-calculus", LIENS Internal Report n. 15, Paris 1993.

[63] J.J. LÉVY, "An Algebraic Interpretation of the $\lambda\beta K$-calculus and a Labelled λ-calculus", λ-calculus and Computer Science Theory, *LNCS* 37, 1975,147-165.

[64] U. DE' LIGUORO, "Non-deterministic Untyped λ-calculus", PhD Thesis, Un. di Roma I, 1991.

[65] U. DE' LIGUORO, A. PIPERNO, "Non-Deterministic Extensions of Untyped λ-calculus", *Info. and Comp.* 122(2), 1995, 149-177.

[66] G. LONGO "Set Theoretical Models of Lambda Calculus: Theory, Expansions and Isomorphisms", *Ann. of Pure and Appl. Logics* 24, 1983, 153-188.

[67] M.G. MAIN, "A Powerdomain Primer", *Bull. of EATCS* 33, 1987, 115-147.

[68] J. MCCARTHY, "A Basis for a Mathematical Theory of Computation", *Computer Programming and Formal Systems*, P. Brafford, D. Hirshberg eds., North-Holland, Amsterdam, 1963, 33-70.

[69] S. MAC LANE, "Categories for the Working Mathematician", Graduate Texts in Mathematics, Springer Verlag, 1971.

[70] A. MEYER, "What is a Model of the Lambda Calculus?", *Info. and Comp.* 52, 1982, 87-122.

[71] R. MILNER, "Fully Abstract Models of Typed λ-calculi", *Theor. Comp. Sci.* 4, 1977, 1-22.

[72] ———, "Function as Processes", *J. of Math. Struct. in Comp. Sci.* 2, 1992, 119-141.

[73] R. Milner, J.G. Parrow, D.J. Walker, "A Calculus of Mobile Processes, Parts I and II", *Info. and Comp.* 100, 1992, 1-40, 41-77.

[74] E. Moggi, "Notions of Computation and Monads", Information and Computation, 93 (1991), 55-92.

[75] J.H. Morris, *Lambda Calculus Models of Programming Languages*, Dissertation, M.I.T. 1968.

[76] F. Nielson, "The Typed λ-Calculus with First-Class Processes", *LNCS* 366, Springer-Verlag, Berlin 1989, 357-369.

[77] C.-H.L. Ong, "The Lazy λ-Calculus : an Investigation into the Foundations of Functional Programming", Ph.D. Thesis , University of London, 1988. Also Prize Fellowship Dissertation, Trinity College, Cambridge.

[78] ———, "Concurrent Lambda Calculus and a General Precongruence Theorem for Applicative Bisimulations", Draft, Cambridge Un., 1992.

[79] ———, "Non-Determinism in a Functional Setting", *LICS '93*, IEEE Comp. Soc. Press, Los Alamitos 1993, 275-286.

[80] R.Plasmeijer, M. van Eekelen, *Functional Programming and Parallel Graph Rewriting*, Addison-Wesley, New York 1993.

[81] G.D. Plotkin, "Call-by-name, Call-by-value and the λ-calculus", *Theor. Comp. Sci.* 1, 1975, 125-159.

[82] ———, "A Powerdomain Construction", *SIAM J. of Comp.* 5, 1976, 457-487.

[83] ———, "A Semantics for Static Type Inference", *Info. and Comp.* 109, 1994, 256-299.

[84] ———, "LCF Considered as a Programming Language", *Theor. Comp. Sci.*, 5, 1977, 223-256.

[85] J.H. Reppy, "Concurrent ML: Design, Application and Semantics", *LNCS* 693, Springer-Verlag, Berlin 1993, 165-198.

[86] S.Ronchi della Rocca, B.Venneri, "Principal Type Schemes for an Extended Type Theory", *Theor. Comp. Sci.* 28, 1984, 151-169.

[87] D. Sangiorgi, "The Lazy λ-calculus in a Concurrency Scenario", *Info. and Comp.* 111(2), 1994, 120-153.

[88] D. Scott, "A Type Theoretical Alternative to ISWIM, CUCH, OWHY", 1969, unpublished until 1993; *Theor. Comp. Sci.* 121, 1993, 411-440.

[89] ———, "Domains for Denotational Semantics", *LNCS* 140, Springer-Verlag, Berlin 1982, 577-613.

[90] K. Sieber, "Call-by-Value and Non-Determinism", *LNCS* 664, Springer-Verlag, Berlin 1993, 376-390.

[91] H. Soendergaard, P. Sestoft, "Non-Determinism in Functional Languages", *The Computer Journal* 35, 1992, 514-523.

[92] M.B. Smyth, "Power Domains", *J. Comp. Sys. Sci.* 16, 1978, 23-36.

[93] B. Thomsen, "A Calculus of Higher-Order Communicating Systems", *POPL'89*, ACM Press, New York 1989, 142-154.

[94] ———, " Calculi for Higher-Order Communicating Systems", Ph.D. Thesis, Imperial College, 1990.

[95] C.P. Wadsworth, " The Relation between Computational and Denotational Properties for Scott's D_∞-models of the Lambda Calculus", *SIAM J. of Comp.* 5, 1976, 488-521.

Samenvatting

Dit proefschrift behandelt in hoofdstuk 2 een λ-calculus uitgebreid met zowel een operator voor parallellisme als voor niet-deterministische keuze. Een reductie relatie, die de klassieke β-reductie uitbreidt, geeft het disjunctieve karakter van het parallellisme en het conjunctieve karakter van het niet-determinisme weer. Er is een natuurlijke operationele semantiek gebaseerd op een gegeneraliseerde vorm van oplosbaarheid. Voor deze taal wordt een filter-model geconstrueerd; een systeem van type toekenning met doorsneden en vereniging geeft een logische beschrijving van dit model. Dit filter-model blijkt adekwaat te zijn voor de operationele semantiek. Een belangrijk hulpmiddel in het bewijs van dit resultaat is een versie van de approximatie stelling, die inhoud dat een type toegekend kan worden aan een term dan en slechts dan als het toegekend kan worden aan een benadering van deze term.

Een natuurlijke uitbreiding is de toevoeging van *angelic* parallellisme en *demonic* niet-determinisme. Dit gebeurt in hoofdstuk 3. In het kader van de *lazy* λ-calculus is het interessant om zowel *call-by-name* als *call-by-value* variabelen in te voeren. De reductie van *call-by-value* abstractie vereist een nadere verfijning van het begrip “waarde”, dat verkregen wordt door onderscheid te maken tussen partiële en totale waarden. Een kleine verandering in het type toekenningssysteem, bekend van het klassieke geval, geeft een logische beschrijving van een filter-model voor de uitgebreide taal. In dit nieuwe filter-model spelen de verenigingstypen een cruciale rol om de *call-by-name* van de *call-by-value* variabelen te onderscheiden. Het hoofd resultaat luidt dat het filter-model de zogenaamde *fully abstract* eigenschap bezit. Een belangrijke stap in het bewijs bestaat uit de constructie van karakteristieke termen, welke de dualiteit tussen conjunctieve en disjunctieve operatoren op een nauwkeurige manier weergeven.

Men kan de niet-deterministische keuze operator interpreteren met behulp van een partiële ordening die fijner is dan de *demonic* relatie; dit vereist dat deze relatie met betrekking tot convergentie zich gedraagt als de Egli-Milner ordening, dit met behoud van de wederzijdse distributiviteit tussen de operatoren voor parallellisme en niet-deterministische keuze. Dit leidt in hoofdstuk 4 tot een nieuwe constructie van een *powerdomain* in de categorie van volledige algebraische tralies (met continue functies als morfismen), welke aan de bovengenoemde eisen voldoet. Ook in dit geval wordt de constructie uitgevoerd met behulp van een geschikt systeem voor type toekenning met doorsnede en vereninginstypen.

De hoofdstukken kunnen onafhankelijk gelezen worden, omdat alle benodigde begrippen en resultaten behandeld worden. Alleen voor sommige bewijzen en vergelijkingen wordt er naar voorafgaande hoofdstukken verwezen.

Curriculum Vitae

Born in Torino (22/12/46).
Master in Physics at the University of Torino cum laude (5/11/70).
CNR fellowship at the "Istituto di Scienze dell'Informazione" of the University of Torino from 1/1/71 to 31/7/72.
Assistant professor of "Sistemi per l'Elaborazione dell'Informazione" at the University of Torino from 1/8/72 to 31/10/81.
Associated professor of "Teoria degli Algoritmi e Calcolabilità" at the University of Torino from 1/11/74 to 31/10/81.
Full professor of "Teoria e Applicazioni delle Macchine Calcolatrici" at the University of Torino since 31/10/81.
Member of the program committees of International Conferences, teacher of Postdoctoral Schools, invited speaker to International Conferences.
Promotor or Co-promotor of the Ph.D. thesis of Felice Cardone, Fabio Alessi, Steffen van Bakel, Franco Barbanera, Adriana Compagnoni, Luigi Liquori.
Member of IFIP W.G.2.2 on "Formal Description of Programming Concepts" since 1/6/84.
Moved to the "Centro Linceo dell'Accademia Nazionale dei Lincei" from 1/11/84 to 31/10/87.
Member of the Editorial Board of "Information and Computation" since 1991.
Member of the "Academia Europaea" since 1994.