Abstract

The present thesis uses *Intersection Types* as powerful tools for describing λ-models and computational properties of λ-terms.

In the first part we prove that many intersection type theories of interest (including those which induce as filter models, Scott's and Park's D_∞ models, the models studied by Barendregt, Coppo and Dezani, and Abramsky and Ong, and Honsell and Ronchi) satisfy an Approximation Theorem with respect to a suitable notion of approximant.

This theorem states that a λ-term has a type if and only if there exists an approximant of that term which has that type. We prove this result uniformly for all the intersection type theories under consideration using a Kripke model analogue of stable sets where bases correspond to worlds.

In the second part we show how to characterise compositionally a number of evaluation properties of λ-terms using intersection type assignment systems. In particular, we focus on termination properties, such as strong normalisation, normalisation, head normalisation, and weak head normalisation. We consider also the persistent versions of such notions. By way of example, we consider also another evaluation property, unrelated to termination, namely reducibility to a closed term.

Many of these characterisation results are new to our knowledge, or else they strengthen or generalise earlier results in the literature.

The completeness parts of the characterisations are proved uniformly for all properties using a set-theoretical semantics of intersection types over suitable kinds of stable sets. This technique generalises Krivine's and Mitchell's methods for strong normalisation to other evaluation properties.

Acknowledgments

I would like to thank the University of Udine, especially Furio Honsell, not only for the financial support of my PhD and guidance in research, but also for giving me an opportunity to complete my PhD, the University of Torino and people in Lambda group at the Computer Science Department for financial support and research collaboration, JAIST where I studied the scientific basis in Computer Science, the University of Leicester, in particular Fer-Jan de Vries and Paula Severi, for their hospitality, and Mariangiola Dezani-Ciancaglini for her full support to my research.

Contents

Introduction

Intersection types were introduced in the late 70's by Dezani and Coppo [CDC80, CDCV80, BCDC83], to overcome the limitations of Curry's type discipline where many interesting terms cannot be typed. The language of Intersection types is very expressive and allows one to describe and capture various properties of λ-terms.

Intersection types have a very significant semantics based on *duality*, which is related to Abramsky's *Domain Theory in Logical Form* [Abr91]. This semantics amounts to the application of that paradigm to the special case of ω-algebraic complete lattice models of pure lambda calculus [CHDCL84]. Namely, types correspond to *compact elements*: the type Ω denoting the least element, intersections denoting *joins* of compact elements, and arrow types denoting *step functions* of compact elements. According to this semantics, a typing judgment can be interpreted as saying that a given term belongs to a pointed compact open set in a ω-algebraic complete lattice model of λ-calculus. Hence, by duality, type theories give rise to *filter λ-models*. Intersection Type Assignment Systems can then be viewed as *finitary logical* descriptions of the interpretation of λ-terms in such models, hence the meaning of a λ-term is the set of types which are deducible for it.

This duality lies at the heart of the success of intersection types as a powerful tool both for analysis and synthesis of λ-models, see for example [AO93, BCDC83, CHDCL84, CDCZ87, Ale91, EHRDR92, HRDR92, DGH93, Plo93, HL99, PRDRR99].

As proved in [Ale91, CHDCL84, Bar] there is a very large class of ω-algebraic models which have natural readings as filter models. This class includes for example all D_∞ inverse limit models, such as models of Scott [Sco72] and Park [Par76], the model in [AO93] which has been studied in connection to the lazy λ-calculus, and the model of [BCDC83] which realizes equality of Böhm trees [RdR82].

A crucial result in the study of the *fine structure*, that is the λ-theory, of ω-algebraic λ-models is the, so called, *Approximation Theorem*, see for example [Wad76, Lon87, Bar84, RDR98, HRDR92]. An Approximation Theorem allows one to express the interpretation of any λ-term, even a non-terminating one, as the supremum of the interpretations of suitable *normal forms*, called the *approximants* of the term, in an appropriate *extended language*. Approximation Theorems are very useful in proving, for instance, *Computational Adequacy* of models with respect to *operational semantics*, see for example [Bar84, HRDR92]. There are other possible methods of showing computational adequacy, both semantical and syntactical, see for example [AO93, Wad76, HRDR92, Pit94], but the method based on Approximation Theorems is usually the most straightforward. Yet, proving an Approximation Theorem for a given model theory can be difficult. Most of the proofs in the literature are based on the technique of *indexed reduction* [Wad76, HRDR92].

However, when the model in question is a filter model, by applying duality, the

Approximation Theorem can be rephrased as follows: the types of a given term are all and only the types of its approximants. This change in perspective opens the way to proving Approximation Theorems via the syntactical route of proof theory, such as *logical predicates* and *computability* techniques.

The aim of Chapter 3 is to show in a uniform way that all the type assignment systems which induce filter models isomorphic to the models in [AO93, BCDC83, RDR98, HRDR92, CDCZ87] satisfy the Approximation Theorem. To this end we use a technique which can be construed as a version of stable sets over a Kripke applicative structure [Mit96].

Intersection type theories are also a very expressive tool for giving *compositional* characterisations, that is a characterisations based on properties of proper subterms, of *evaluation* properties of λ-terms. The seminal result in this respect is that the Ω-free fragment of intersection-types allows one to type *all* and *only* the *strongly normalising terms.* This is largely a folklore result; the first published proof appears in [Pot80].

Since then, the number of intersection type theories, used for studying the fine structure of the denotational semantics of untyped λ-calculus, has increased considerably: for example [CHDCL84, CDCZ87, HRDR92, EHRDR92, AO93, Plo93, HL99]. In all these cases the corresponding intersection type assignment systems are used to provide *finite logical presentations* of particular domain models, which can thereby be viewed also as *filter models.* And hence, intersection type theories provide characterisations of particular semantical properties.

In Chapter 4 we address the problem of investigating *uniformly* the use of intersection type theories, and corresponding type assignment systems, for giving a *compositional* characterisation of *evaluation* properties of λ-terms.

In particular we discuss *termination* properties such as *strong normalisation, normalisation, head normalisation, weak head normalisation.* We consider also the *persistent* versions of such notions, that is when these properties are stable under application of terms; see Definition 4.1.2. By way of example we consider also another evaluation property, unrelated to termination, namely reducibility to a *closed term.*

Many of the characterisation results that we give are indeed inspired by earlier semantical work on filter models of the untyped λ-calculus, but they are rather novel in spirit. We focus, in fact, on *proof-theoretic* properties of intersection type assignment systems *per se.* Most of our characterisations are therefore new, to our knowledge, or else they streamline, strengthen, or generalise earlier results in the literature.

The *completeness* part of the characterisations is proved uniformly for all the properties. We use a very elementary presentation of the technique of *logical relations* phrased in terms of a *set-theoretical* semantics of intersection types over suitable kinds of *stable* sets. This technique generalises Krivine's [Kri90] and Mitchell's [Mit96] proof methods for *strong normalisation* to other evaluation properties.

The present thesis is organised as follows: in Chapter 1 we introduce Intersection Type Assignment System and in Chapter 2 we characterise their properties. By using Intersection types, Approximation Theorems and characterisations of λ-terms properties are given in Chapter 3, which is based on [DCHM01], and Chapter 4, which is based on [DCHM], respectively.

1

Intersection Type Systems

There are various alternative, but ultimately equivalent, presentations of Intersection Type Assignment Systems. We shall focus on a natural deduction style version, because this is most commonly used.

Intersection types are syntactic objects forming an algebra $\mathbf{T}$ freely generated from a set of atoms $\mathbf{C}$ by means of the operators $\rightarrow$ and $\cap$. Using some set ∇ of axioms and rules a preorder $\leq_\nabla$ is defined on $\mathbf{T}$. Working modulo the corresponding equivalence relation $\sim_\nabla$ one obtains a meet semi-lattice (poset with meet operator) $\langle \mathbf{T}/{\sim_\nabla}, \leq, \cap \rangle$. For almost all ∇ the operator $\rightarrow$ preserves $\sim_\nabla$ and one obtains a type structure $\langle \mathbf{T}/{\sim_\nabla}, \rightarrow, \leq, \cap \rangle$.

Given ∇ (or just the type structure) one defines a type assignment system $\lambda\cap^\nabla$. For these type assignment systems the type reconstruction problem is in general undecidable. Using ∇ one can semantically characterise several sets of lambda terms, e.g. the (strongly) normalizing ones. Using ∇ one also defines corresponding *filter λ-models* $\mathcal{F}^\nabla$. It turns out that several classical and new models for untyped λ-calculus can be obtained in this way. Using approximation theorems for the type assignment systems the local structure of the models will be studied.

We introduce the system and its basic properties in this chapter. More precisely, intersection type languages and intersection type theories are introduced in Section 1.1. In Section 1.2 the corresponding type assignment systems are defined.

1.1 Intersection type languages and theories

Intersection types are syntactical objects built by closing a given countable set $\mathbf{C}$ of *atomic types* (type constants) under the *function type* constructor $\rightarrow$ and the *intersection type* constructor $\cap$.

Definition 1.1.1 (Intersection Type Language) *The* intersection type language *over* $\mathbf{C}$*, denoted by* $\mathbf{T} = \mathbf{T}(\mathbf{C})$ *is defined by the following abstract syntax:*

$$\mathbf{T} = \mathbf{C} \mid \mathbf{T}\rightarrow\mathbf{T} \mid \mathbf{T} \cap \mathbf{T}.$$

Notation 1.1.2 *Arbitrary atomic types will be denoted by* $a, b, \ldots$ *and arbitrary types by* $A, B, \ldots$. *When writing intersection types we shall use the following convention: the constructor* $\cap$ *takes precedence over the constructor* $\rightarrow$ *and it associates to the right. For example*

$$(A\rightarrow B\rightarrow C) \cap A\rightarrow B\rightarrow C \equiv ((A\rightarrow(B\rightarrow C)) \cap A)\rightarrow(B\rightarrow C).$$

Remark 1.1.3 *In the literature [CDCV81, BCDC83] intersection types are usually built starting also from type variables that can be replaced by an arbitrary type. This makes types which contain variables akin to* type schemes*. This is a syntactic tool which has been extensively used for studying principal types of* λ*-terms (see e.g. [CDCV80, RDRV84, Bak93]).*

We shall be concerned with several different intersection type languages arising from taking different sets of atomic types, depending on which properties we want to capture. Two typical choices for the set of atomic types will be $\mathbf{C}_\infty$ or a given countable set of constants or finite sets like $\{\Omega, \varphi, \omega\}$ or $\{\nu\}$.

The expressive power of intersection type languages is remarkable. This will become apparent when we will use them as a tool for characterising properties of λ-terms. Much of this expressive power comes from the fact that they are endowed with a *preorder relation*, $\leq$, which induces, on the set of types, the structure of a meet semi-lattice with respect to $\cap$. This appears natural when we think of types as sets of denotations and interpret $\cap$ as set-theoretic intersection, and $\leq$ as set inclusion.

Definition 1.1.4 (Intersection Type Preorder) *Let* $\mathbf{T} = \mathbf{T}(\mathbf{C})$ *be an intersection type language. An* intersection type preorder *over* $\mathbf{T}$ *is a binary relation* $\leq$ *on* $\mathbf{T}$ *satisfying the following set* ∇^0 *("nabla-zero") of axioms and rules:*

(refl)	$A \leq A$
(idem)	$A \leq A \cap A$
(incl$_L$)	$A \cap B \leq A$
(incl$_R$)	$A \cap B \leq B$
(mon)	$\dfrac{A \leq A' \quad B \leq B'}{A \cap B \leq A' \cap B'}$
(trans)	$\dfrac{A \leq B \quad B \leq C}{A \leq C}$

Notation 1.1.5 *We will write $A \sim B$ for $A \leq B$ and $B \leq A$.*

Notice that associativity and commutativity of $\cap$ (modulo $\sim$) follow easily from the above axioms and rules. For instance, commutativity is immediate by applying rules (idem), (incl$_L$), (incl$_R$) and (mon) as follows:

$$A \cap B \leq (A \cap B) \cap (A \cap B) \leq B \cap A.$$

Notation 1.1.6 *Being $\cap$ commutative and associative, we will write $\bigcap_{i \leq n} A_i$ for $A_1 \cap \ldots \cap A_n$. Similarly we will write $\bigcap_{i \in I} A_i$. We convene that I, J, K etc, when referred to as sets of indexes for types, always denote* finite *sets. Moreover $A^n \to B$ will be short for $\underbrace{A \to \cdots \to A}_{n} \to B$.*

Remark 1.1.7 *It is not required that the constructor $\to$ is compatible with $\sim$, but for many type theories (see below) this will be implied by the axiom (η) or ($\eta^\sim$) below.*

In order to have syntactical, possibly effective, presentations of intersection type preorders we introduce the following notion of *intersection type theory*, axiomatized by the basic set ∇^0 for $\leq$ and adding a set of special purpose axioms and rules.

Definition 1.1.8 (Intersection Type Theory) *Let $\mathbf{T} = \mathbf{T}(\mathbf{C})$ be an intersection type language, and let ∇ be a collection of axioms and rules for judgements of the shape $A \leq B$, with $A, B \in \mathbf{T}$. The* intersection type theory *$\Sigma(\mathbf{C}, \nabla)$ is the set of all judgments $A \leq B$ derivable from the axioms and rules in $\nabla^0 \cup \nabla$.*

Notation 1.1.9 *When we consider the intersection type theory $\Sigma(\mathbf{C}, \nabla)$, we will write*

$$\begin{array}{lll} \mathbf{C}^\nabla & \textit{for} & \mathbf{C}, \\ \mathbf{T}^\nabla & \textit{for} & \mathbf{T}(\mathbf{C}), \\ \Sigma^\nabla & \textit{for} & \Sigma(\mathbf{C}, \nabla). \end{array}$$

Moreover $A \leq_\nabla B$ will be short for $(A \leq B) \in \Sigma^\nabla$. Finally we define $A \sim_\nabla B \iff A \leq_\nabla B \leq_\nabla A$.

(Ω)	$A \leq \Omega$
(ν)	$A{\to}B \leq \nu$
$(\Omega\text{-}\eta)$	$\Omega \leq \Omega{\to}\Omega$
$(\Omega\text{-}lazy)$	$A{\to}B \leq \Omega{\to}\Omega$
$({\to}\text{-}\cap)$	$(A{\to}B) \cap (A{\to}C) \leq A{\to}B \cap C$
$({\to}\text{-}\cap^{\sim})$	$(A{\to}B) \cap (A{\to}C) \sim A{\to}B \cap C$
(η)	$\dfrac{A' \leq A \quad B \leq B'}{A{\to}B \leq A'{\to}B'}$
$(\eta^{\sim})$	$\dfrac{A' \sim A \quad B \sim B'}{A{\to}B \sim A'{\to}B'}$
$(\omega\text{-}Scott)$	$\Omega{\to}\omega \sim \omega$
$(\omega\text{-}Park)$	$\omega{\to}\omega \sim \omega$
$(\omega\varphi)$	$\omega \leq \varphi$
$(\varphi{\to}\omega)$	$\varphi{\to}\omega \sim \omega$
$(\omega{\to}\varphi)$	$\omega{\to}\varphi \sim \varphi$
(I)	$(\varphi{\to}\varphi) \cap (\omega{\to}\omega) \sim \varphi$

Figure 1.1: Possible Axioms and Rules concerning $\leq$.

Definition 1.1.10 *The structure $\langle \mathbf{T}^{\nabla}/{\sim_{\nabla}}, \leq, \cap\rangle$ is obtained by defining in the obvious way $\leq$ and $\cap$, i.e. $[A] \leq [B] \iff A \leq_{\nabla} B$ and $[A] \cap [B] = [A \cap B]$. We often write A for $[A]$ and reason with it modulo $\sim_{\nabla}$.*

Remark 1.1.11 *It is easy to prove that the structure $\langle \mathbf{T}^{\nabla}/{\sim_{\nabla}}, \leq, \cap\rangle$ is a* meet semilattice.

In Figure 1.1 a list appears of special purpose axioms and rules which have been considered in the literature, and which we shall discuss in this part.

The axiom (Ω) states that the resulting type theory has a maximal element. Axiom (Ω) is particularly meaningful when used in combination with the Ω-type assignment system, which essentially treats Ω as the universal type of all λ-terms (see Definition 1.2.5).

The meaning of the other axioms and rules can be grasped if we take types to denote subsets of a domain of discourse and we view $\to$ as the function space constructor in the light of Curry-Scott semantics (see [Sco75]). Thus the type $A{\to}B$ denotes the set of *total* functions which map each element of A into an element of B.

The axiom (ν) states that ν is above any arrow type. This axiom agrees with the ν-type assignment system, which treats ν as the universal type of all λ-abstractions (see Definition 1.2.7). Notice that the role of ν as universal type for λ-abstractions may be played by the type $\Omega\to\Omega$, when Ω is in $\mathbf{C}^\nabla$, since $\Omega \to \Omega$ can be assigned to any λ-abstraction (see Definition 1.2.5). For this reason it is of no use to have at the same time ν and Ω, hence we impose that the two constants do not occur together in any $\mathbf{C}^\nabla$. The elements Ω and ν play a very crucial role in the development of the theory: for this reason we take them always with their associated axioms, (Ω) and (ν) respectively. To sum up, we assume the following important blanket assumptions:

$$\begin{array}{lcl}\mathbf{A}_1 & : & \text{only one between } \Omega \text{ and } \nu \text{ may belong to a given } \mathbf{C}^\nabla;\\ \mathbf{A}_2 & : & \text{if } \Omega \in \mathbf{C}^\nabla \text{ then } (\Omega) \in \nabla;\\ \mathbf{A}_3 & : & \text{if } \nu \in \mathbf{C}^\nabla \text{ then } (\nu) \in \nabla.\end{array}$$

In the same spirit, if Ω is the maximal element, i.e. the whole universe, then $\Omega\to\Omega$ is the set of functions which applied to an arbitrary element return again an arbitrary element. Thus, axiom $(\Omega\text{-}\eta)$ expresses the fact that all the objects in our domain of discourse are total functions, i.e. that Ω is equal to $\Omega\to\Omega$ (see [BCDC83]). If now we want to capture only those terms which truly represent functions, as we do for example in the lazy λ-calculus, we cannot assume axiom $(\Omega\text{-}\eta)$. One still may postulate the weaker property (Ω-*lazy*) to make all functions total (see [AO93]). It simply says that an element which is a function, because it maps A into B, maps also the whole universe into itself.

The intended interpretation of arrow types motivates axiom $(\to\text{-}\cap)$, which implies that if a function maps A into B, and the same function maps also A into C, then, actually, it maps the whole A into the intersection between B and C, i.e. $B \cap C$ (see [BCDC83]).

Rule (η) is also very natural in view of the set-theoretic interpretation. It implies that the arrow constructor is contravariant in the first argument and covariant in the second one. It is clear that if a function maps A into B, and we take a subset A' of A and a superset B' of B, then this function will map also A' into B' (see [BCDC83]).

The rules $(\to\text{-}\cap^\sim)$ and $(\eta^\sim)$ are similar to the rules $(\to\text{-}\cap)$ and (η). They capture properties of the graph models for the untyped lambda calculus (see [Plo75, Eng81]).

The remaining axioms express peculiar properties of D_∞-like inverse limit models (see [BCDC83, CHDCL84, CDCZ87, HRDR92, HL93]).

We can introduce now a list of significant intersection type theories which have been extensively considered in the literature. The order is logical, rather than historical, and some references define the models, others deal with the corresponding filter models (see [CDC80, CDCV81, HL99, EHRDR92, AO93, BCDC83, Sco72, Par76, CDCZ87, HRDR92, DCHM00, Plo93, Eng81]).

We shall denote such theories as Σ^∇, with various different names ∇ corresponding to the initials of the authors who have first considered the λ-model induced by such a theory. For each such ∇ we specify in Figure 1.2 the type theory $\Sigma^\nabla = \Sigma(\mathbf{C}, \nabla)$ by giving the set of constants $\mathbf{C}^\nabla$ and the set ∇ of extra axioms and rules taken from Figure 1.1. Here $\mathbf{C}_\infty$ is an infinite set of fresh (i.e. different from $\Omega, \nu, \phi, \varphi, \omega$) constants.

$\mathbf{C}^{\mathcal{CD}}$	$=$	$\mathbf{C}_\infty$	$\mathcal{CD}$	$=$	$\emptyset$	[CDC80]
$\mathbf{C}^{\mathcal{CDV}}$	$=$	$\mathbf{C}_\infty$	$\mathcal{CDV}$	$=$	$\{(\to\text{-}\cap), (\eta)\}$	[CDCV81]
$\mathbf{C}^{\mathcal{HL}}$	$=$	$\{\varphi, \omega\}$	$\mathcal{HL}$	$=$	$\mathcal{CDV} \cup \{(\omega\varphi), (\varphi{\to}\omega), (\omega{\to}\varphi)\}$	[HL93]
$\mathbf{C}^{\mathcal{EHR}}$	$=$	$\{\nu\}$	$\mathcal{EHR}$	$=$	$\mathcal{CDV} \cup \{(\nu)\}$	[EHRDR92]
$\mathbf{C}^{\mathcal{AO}}$	$=$	$\{\Omega\}$	$\mathcal{AO}$	$=$	$\mathcal{CDV} \cup \{(\Omega), (\Omega\text{-}lazy)\}$	[AO93]
$\mathbf{C}^{\mathcal{BCD}}$	$=$	$\{\Omega\} \cup \mathbf{C}_\infty$	$\mathcal{BCD}$	$=$	$\mathcal{CDV} \cup \{(\Omega), (\Omega\text{-}\eta)\}$	[BCDC83]
$\mathbf{C}^{\mathcal{S}c}$	$=$	$\{\Omega, \omega\}$	$\mathcal{S}c$	$=$	$\mathcal{BCD} \cup \{(\omega\text{-}Scott)\}$	[Sco72]
$\mathbf{C}^{\mathcal{P}a}$	$=$	$\{\Omega, \omega\}$	$\mathcal{P}a$	$=$	$\mathcal{BCD} \cup \{(\omega\text{-}Park)\}$	[Par76]
$\mathbf{C}^{\mathcal{CDZ}}$	$=$	$\{\Omega, \varphi, \omega\}$	$\mathcal{CDZ}$	$=$	$\mathcal{HL} \cup \mathcal{BCD}$	[CDCZ87]
$\mathbf{C}^{\mathcal{HR}}$	$=$	$\{\Omega, \varphi, \omega\}$	$\mathcal{HR}$	$=$	$\mathcal{BCD} \cup \{(\omega\varphi), (I), (\omega{\to}\varphi)\}$	[HRDR92]
$\mathbf{C}^{\mathcal{DHM}}$	$=$	$\{\Omega, \varphi, \omega\}$	$\mathcal{DHM}$	$=$	$\mathcal{BCD} \cup \{(\omega\varphi), (\omega\text{-}Scott), (\omega{\to}\varphi)\}$	[DCHM00]
$\mathbf{C}^{\mathcal{P}l}$	$=$	$\{\Omega, \phi\}$	$\mathcal{P}l$	$=$	$\{(\Omega), (\eta^{\sim})\}$	[Plo93]
$\mathbf{C}^{\mathcal{E}n}$	$=$	$\{\Omega\} \cup \mathbf{C}_\infty$	$\mathcal{E}n$	$=$	$\mathcal{P}l \cup \{(\to\text{-}\cap^{\sim}),\ (\Omega\text{-}\eta)\}$	[Eng81]

Figure 1.2: Type Theories: constants, axioms and rules, and references

1.2 Intersection Type Assignment Systems

Now that we have introduced intersection type theories we have to explain how to capitalize effectively on their expressive power. This is achieved via the crucial notion of *Intersection Type Assignment System*. This is a natural extension of Curry's type (see [Cur34, Cur36, CF58, CHS72, Bar92]) to intersection types. First we need some preliminary definitions and notations.

Definition 1.2.1 (λ-terms) *Assume that there is an infinite sequence Var of distinct symbols $x, y, \ldots$ called* variables*. The set of* λ-terms *is defined inductively as follows:*

i) All variables $x, y, \ldots \in Var$ are λ-terms.

ii) If M and N are any λ-terms then MN is a λ-terms.

iii) If M is any λ-terms and $x \in Var$ then $(\lambda x.M)$ is a λ-term.

An occurrence of a variable x in a λ-term M is free*, if it does not occur in a subterm of M of the shape $\lambda x.N$. If x has at least one free occurrence in M, it is called a* free variable of M*, denoted by $x \in FV(M)$.*

Definition 1.2.2 *i) A* ∇-basis *is a (possibly infinite) set of statements of the shape $x{:}B$, where $B \in \mathbf{T}^{\nabla}$, with all variables distinct.*

ii) An Intersection Type Assignment System *relative to Σ^{∇}, notation $\lambda\cap^{\nabla}$, is a formal system for deriving judgements of the form $\Gamma \vdash^{\nabla} M : A$, where the* subject *$M$ is an untyped λ-term, the* predicate *A is in $\mathbf{T}^{\nabla}$, and Γ is a ∇-basis.*

iii) We say that a term M is typable *in $\lambda\cap^{\nabla}$, for a given ∇-basis Γ, if there is a type $A \in \mathbf{T}^{\nabla}$ such that the judgement $\Gamma \vdash^{\nabla} M : A$ is derivable.*

Actually, according to the set of constants belonging to $\mathbf{C}^{\nabla}$, various type assignment systems can be defined for the type theory Σ^{∇}.

The simplest one is given in the following definition.

Definition 1.2.3 (Basic Type Assignment System)
Let Σ^{∇} be a type theory. The axioms and rules of the basic system type assignment*, denoted by $\lambda\cap^{\nabla}_{\mathcal{B}}$, for deriving judgments $\Gamma \vdash^{\nabla}_{\mathcal{B}} M : A$, are the following:*

(Ax)	$\Gamma \vdash^{\nabla}_{\mathcal{B}} x{:}A$	*if* $(x{:}A \in \Gamma)$
$(\to\mathrm{I})$	$\dfrac{\Gamma, x{:}A \vdash^{\nabla}_{\mathcal{B}} M : B}{\Gamma \vdash^{\nabla}_{\mathcal{B}} \lambda x.M : A{\to}B}$	
$(\to\mathrm{E})$	$\dfrac{\Gamma \vdash^{\nabla}_{\mathcal{B}} M : A \to B \quad \Gamma \vdash^{\nabla}_{\mathcal{B}} N : A}{\Gamma \vdash^{\nabla}_{\mathcal{B}} MN : B}$	
$(\cap\mathrm{I})$	$\dfrac{\Gamma \vdash^{\nabla}_{\mathcal{B}} M : A \quad \Gamma \vdash^{\nabla}_{\mathcal{B}} M : B}{\Gamma \vdash^{\nabla}_{\mathcal{B}} M : A \cap B}$	
$(\leq_{\nabla})$	$\dfrac{\Gamma \vdash^{\nabla}_{\mathcal{B}} M : A \quad A \leq_{\nabla} B}{\Gamma \vdash^{\nabla}_{\mathcal{B}} M : B}$	

Example 1.2.4 *Self-application can be easily typed in* $\lambda\cap^{\nabla}_{\mathcal{B}}$*, as follows.*

$$\dfrac{\dfrac{\dfrac{x{:}(A{\to}B)\cap A \vdash^{\nabla}_{\mathcal{B}} x{:}(A{\to}B)\cap A}{x{:}(A{\to}B)\cap A \vdash^{\nabla}_{\mathcal{B}} x{:}A{\to}B}\,(\leq_{\nabla}) \quad \dfrac{x{:}(A{\to}B)\cap A \vdash^{\nabla}_{\mathcal{B}} x{:}(A{\to}B)\cap A}{x{:}(A{\to}B)\cap A \vdash^{\nabla}_{\mathcal{B}} x{:}A}\,(\leq_{\nabla})}{x{:}(A{\to}B)\cap A \vdash^{\nabla}_{\mathcal{B}} xx : B}\,(\to\mathrm{E})}{\vdash^{\nabla}_{\mathcal{B}} \lambda x.xx : (A{\to}B)\cap A{\to}B}\,(\to\mathrm{I})$$

Any Intersection Type Assignment System $\lambda\cap^{\nabla}$ is clearly an extension of Curry's type assignment system defined by the axiom (Ax) and the rules $(\to\mathrm{I})$ and $(\to\mathrm{E})$.

Note that rule $(\leq_{\nabla})$ is the standard type-subsumption rule in the context of the type assignment system $\lambda\cap^{\nabla}$.

If $\Omega \in \mathbf{C}^{\nabla}$, a natural choice is to set Ω as the universal type of all λ-terms. This amounts to modifying the basic type assignment system by adding a suitable axiom for Ω.

Definition 1.2.5 (Ω-type Assignment System)
Let Σ^{∇} *be a type theory with* $\Omega \in \mathbf{C}^{\nabla}$*. The axioms and rules of the* Ω-type assignment system *(denoted* $\lambda\cap^{\nabla}_{\Omega}$*), for deriving judgments of the form* $\Gamma \vdash^{\nabla}_{\Omega} M : A$*, are those of the basic one, plus the further axiom*

$$(Ax\text{-}\Omega) \quad \Gamma \vdash^{\nabla}_{\Omega} M : \Omega.$$

Example 1.2.6 *Also non-strongly normalizing terms can be typed in* $\lambda\cap^{\nabla}_{\Omega}$ *with a type which is not eqivalent to* Ω*. Note the usage of the axiom* $(Ax\text{-}\Omega)$*. Let* $\Delta \equiv \lambda x.xx$*.*

$$\dfrac{\dfrac{\dfrac{x{:}A, y{:}\Omega \vdash^{\nabla}_{\Omega} x : A}{y{:}\Omega \vdash^{\nabla}_{\Omega} \lambda x.x : A{\to}A}\,(\to\mathrm{I})}{\vdash^{\nabla}_{\Omega} \lambda yx.x : \Omega{\to}A{\to}A}\,(\to\mathrm{I}) \quad \Delta\Delta : \Omega}{\vdash^{\nabla}_{\Omega} (\lambda yx.x)(\Delta\Delta) : A{\to}A}\,(\to\mathrm{E})$$

An interesting example is that the Fixedpoint Combinator $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ *can be typed in* $\lambda\cap^{\nabla}_{\Omega}$ *as follows.*

$$\dfrac{\dfrac{\dfrac{\dfrac{f{:}\Omega \to A \vdash^{\nabla}_{\Omega} f{:}\Omega \to A \quad f{:}\Omega \to A \vdash^{\nabla}_{\Omega} xx : \Omega}{f{:}\Omega \to A \vdash^{\nabla}_{\Omega} f(xx) : A}\,(\to\mathrm{E})}{f{:}\Omega \to A \vdash^{\nabla}_{\Omega} \lambda x.f(xx) : \Omega \to A}\,(\to\mathrm{I}) \qquad f{:}\Omega \to A \vdash^{\nabla}_{\Omega} \lambda x.f(xx) : \Omega}{f{:}\Omega \to A \vdash^{\nabla}_{\Omega} (\lambda x.f(xx))(\lambda x.f(xx)) : A}\,(\to\mathrm{E})}{\vdash^{\nabla}_{\Omega} \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)) : (\Omega \to A) \to A}\,(\to\mathrm{I})$$

Analogously to the case of Ω, when $\nu \in \mathbf{C}^{\nabla}$, it is natural to consider it as the universal type for abstractions, hence modifying the basic system by the addition of a special axiom for ν.

Definition 1.2.7 (ν-type Assignment System)
Let Σ^{∇} *be a type theory with* $\nu \in \mathbf{C}^{\nabla}$*. The axioms and rules of the* ν-type assignment system *(denoted* $\lambda\cap^{\nabla}_{\nu}$*), for deriving judgments of the form* $\Gamma \vdash^{\nabla}_{\nu} M : A$*, are those of the basic one, plus the further axiom*

$$(Ax\text{-}\nu) \quad \Gamma \vdash^{\nabla}_{\nu} \lambda x.M : \nu.$$

Example 1.2.8 *Using axiom* $(Ax\text{-}\nu)$ *we can again type non-strongly normalizing terms, but not the term of Example 1.2.6, as proved in [EHRDR92].*

$$\dfrac{\dfrac{\dfrac{x{:}A, y{:}\nu \vdash^{\nabla}_{\nu} x : A}{y{:}\nu \vdash^{\nabla}_{\nu} \lambda x.x : A{\to}A}\,(\to\mathrm{I})}{\vdash^{\nabla}_{\nu} \lambda yx.x : \nu{\to}A{\to}A}\,(\to\mathrm{I}) \qquad \lambda z.\Delta\Delta : \nu}{\vdash^{\nabla}_{\nu} (\lambda yx.x)(\lambda z.\Delta\Delta) : A{\to}A}\,(\to\mathrm{E})$$

For simplicity we convene that the symbols Ω and ν are reserved to the universal type constants respectively used in the systems $\lambda\cap^{\nabla}_{\Omega}$ and $\lambda\cap^{\nabla}_{\nu}$. I.e. we forbid $\Omega \in \mathbf{C}^{\nabla}$ or $\nu \in \mathbf{C}^{\nabla}$ when we deal with $\lambda\cap^{\nabla}_{\mathcal{B}}$.

Notation 1.2.9 *In the following,* $\lambda\cap^{\nabla}$ *will range over* $\lambda\cap^{\nabla}_{\mathcal{B}}$*,* $\lambda\cap^{\nabla}_{\Omega}$ *and* $\lambda\cap^{\nabla}_{\nu}$*. More precisely we assume that* $\lambda\cap^{\nabla}$ *stands for* $\lambda\cap^{\nabla}_{\Omega}$ *whenever* $\Omega \in \mathbf{C}^{\nabla}$*, for* $\lambda\cap^{\nabla}_{\nu}$ *whenever* $\nu \in \mathbf{C}^{\nabla}$*, and for* $\lambda\cap^{\nabla}_{\mathcal{B}}$ *otherwise. Similarly for* $\vdash^{\nabla}$*. We call* $\lambda\cap^{\nabla}$ Intersection Type Assignment System.

Axiom (Ax-Ω), introduced in [BCDC83], implies that Ω is the universal type, i.e. it may be assigned to any term without restriction. Axiom (Ax-ν), defined in [EHRDR92], implies that ν is the universal type of functions, i.e. it can be assigned to all λ-abstractions. If $\Omega \in \mathbf{C}^{\nabla}$, we could achieve the same effect using $\Omega{\to}\Omega$ in place of ν. However, there are many situations, where it is not convenient to have a universal type, but still we want to be able to type all λ-abstractions, for instance when discussing call-by-value λ-calculus. Therefore the type ν can come in handy. This gives further motivations to our choice of never considering systems containing both ν and Ω.

Notice that the subterm property does not hold in general for $\lambda\cap_\nu^\nabla$. In fact $\lambda x.M$ is typable also when M is not typable. Moreover, in $\lambda\cap_\Omega^\nabla$ and $\lambda\cap_\nu^\nabla$, a judgment $\Gamma \vdash^\nabla M : A$ does not imply $\mathrm{FV}(M) \subseteq \Gamma$.

Notice that intersection elimination rules

$$(\cap\mathrm{E}) \quad \frac{\Gamma \vdash^\nabla M : A \cap B}{\Gamma \vdash^\nabla M : A} \quad \frac{\Gamma \vdash^\nabla M : A \cap B}{\Gamma \vdash^\nabla M : B}.$$

can be immediately proved to be derivable in all $\lambda\cap^\nabla$.

Remark 1.2.10 *In the above definitions, λ-terms are considered modulo α-conversion. Alternatively (i.e. to avoid the use of α-conversion) one could have added the weakening rule to the type assignment systems:*

$$(\mathit{weakening}) \quad \frac{\Gamma \vdash M : A \quad x \notin \Gamma}{\Gamma, x{:}B \vdash M : A}.$$

Remark 1.2.11 *An (intersection)* type structure *is of the form*

$$\mathcal{S} = \langle \mathcal{S}, \to, \leq, \cap \rangle$$

where:

- $\to, \cap$ *are total functions from* $\mathcal{S} \times \mathcal{S}$ *to* $\mathcal{S}$*;*
- $\leq$ *is a partial order on* $\mathcal{S}$*;*
- $\to, \cap$ *are compatible with* $\sim$ *(the equivalence induced by* $\leq$*).*

For type theories Σ^∇ *such that* $\to$ *is compatible with* $\sim_\nabla$ *one has that*

$$\mathcal{S}^\nabla = \langle \mathbf{T}^\nabla/{\sim_\nabla}, \to, \leq, \cap \rangle$$

is a type structure. Again $\to, \leq, \cap$ *are defined in the obvious way. The definition of Intersection Type Assignment System can also be given for an intersection type structure* $\mathcal{S}$*. In that case one obtains the system* $\lambda\cap^{\mathcal{S}}$ *and writes* $\vdash^{\mathcal{S}}$ *for its type assignment. It is easy to observe that in the case* $\mathcal{S} = \mathcal{S}^\nabla$ *the resulting* $\vdash$ *is the same as* $\vdash^\nabla$ *after the identification of* A *and* $[A]$*. The reason for introducing* $\vdash^\nabla$ *in a more general way (resulting in an* $\to$ *not compatible with* $\sim_\nabla$*) is to capture Krivine's systems* $\mathcal{D}(\Omega)$ *not having* $\leq$ *relation on types, but only the (formal)* $\cap$ *(see [Kri90]).*

From the view point of the proposition-as-types, λ-terms-as-proofs paradigm and the Intersection Type Assignment System is rather non-standard. First of all, viewing types as propositions, both $\to$ and $\leq_\nabla$ can be interpreted as the *relevant* implication (see [Mey95, Ven94]), while $\to$ corresponds to the ordinary implicational connective, along the lines of the Curry-Howard isomorphism ([CF58, How80]). Furthermore, as shown in [Hin82], rule ($\cap$I) does not correspond to the ordinary conjunction introduction. The behaviour of $\cap$ is rather that of a proof-theoretic connective, i.e. a connective which is sensitive to the way the premises have been established (see [LE85, AB91, BM94]). Rule ($\cap$I) requires, in fact, that both proofs of the propositions A and B have the same structure, i.e. that they are the same λ-term. Alternatively, as shown in [Mey95, Ven94], intersection can be seen to correspond to *relevant* conjunction.

As we remarked earlier, there are various equivalent alternative presentations of Intersection Type Assignment Systems. We have chosen a natural deduction presentation, where ∇-bases are additive. We could have taken, just as well, a sequent style presentation and replace rule ($\to$E) with the three rules ($\to L$), ($\cap L$) and (cut) occuring in Proposition 1.2.13 (see [BDCL95, BG00]). Next to this we could have formulated the rules so that ∇-bases "multiply". Notice that because of the presence of the type constructor $\cap$, a special notion of *multiplication of ∇-bases* can be given.

Definition 1.2.12 (Multiplication of ∇-bases)

$$\begin{array}{rcl} \Gamma \uplus \Gamma' & = & \{x{:}\, A \cap B \mid x{:}\, A \in \Gamma \ \textit{and}\ x{:}\, B \in \Gamma'\} \\ & \cup & \{x{:}\, A \mid x{:}\, A \in \Gamma \ \textit{and}\ x \notin \Gamma'\} \\ & \cup & \{x{:}\, B \mid x{:}\, B \in \Gamma' \ \textit{and}\ x \notin \Gamma\}. \end{array}$$

Accordingly we define:

$$\Gamma \sqsubseteq_{\uplus} \Gamma' \iff \exists \Gamma''.\, \Gamma \uplus \Gamma'' = \Gamma'.$$

For example, $\{x{:}A, y{:}B\} \uplus \{x{:}C, z{:}D\} = \{x{:}A \cap C, y{:}B, z{:}D\}$.

Using the above definition the following rules can be easily shown to be admissible in the system $\lambda\cap^\nabla$.

$$(\textit{multiple weakening}) \quad \frac{\Gamma_1 \vdash M : A}{\Gamma_1 \uplus \Gamma_2 \vdash M : A}$$

$$(\textit{relevant} \to \mathrm{E}) \quad \frac{\Gamma_1 \vdash M : A \to B \quad \Gamma_2 \vdash N : A}{\Gamma_1 \uplus \Gamma_2 \vdash MN : B}$$

$$(\textit{relevant}\ \cap \mathrm{I}) \quad \frac{\Gamma_1 \vdash M : A \quad \Gamma_2 \vdash M : B}{\Gamma_1 \uplus \Gamma_2 \vdash M : A \cap B}$$

More interestingly, one can try to replace rule ($\leq_\nabla$) with other more perspicuous rules. This is possible for some simple theories. We will see some important examples of this approach in Proposition 2.1.10, as soon as we will have proved appropriate "inversion" theorems for $\lambda\cap^\nabla$. For some very special theories, one can even remove altogether with rule ($\leq_\nabla$), provided the remaining rules are reformulated "multiplicatively" with respect to ∇-bases (see e.g. [DGH93]). We shall not follow up this line of investigation.

In $\lambda\cap^{\nabla}$, assumptions are allowed to appear in the ∇-basis without any restriction. Alternatively, we might introduce a *relevant* Intersection Type Assignment System, where only "minimal-base" judgments are derivable (see [HRDR92]). Rules like (*relevant* $\rightarrow$E) and (*relevant* $\cap$I), which exploit the above notion of multiplication of ∇-bases, are essential for this purpose. Relevant systems are necessary, for example, for giving finitary logical descriptions of qualitative domains as defined in [GLT89]. We will not, however, follow up this line of research (see [HRDR92]).

A first simple proposition, which can be proved straightforwardly by induction on the structure of derivations is the following.

Proposition 1.2.13 *For arbitrary intersection type theories Σ^{∇} the following rules are admissible in the Intersection Type assignment System $\lambda\cap^{\nabla}$.* [1]

$$
\begin{array}{ll}
(\textit{weakening}) & \dfrac{\Gamma \vdash^{\nabla} M : A \quad x \notin \Gamma}{\Gamma, x{:}B \vdash^{\nabla} M : A}; \\
(\textit{strengthening}) & \dfrac{\Gamma, x{:}B \vdash^{\nabla} M : A \quad x \notin \mathrm{FV}(M)}{\Gamma \vdash^{\nabla} M : A}; \\
(\textit{cut}) & \dfrac{\Gamma, x{:}B \vdash^{\nabla} M : A \quad \Gamma \vdash^{\nabla} N : B}{\Gamma \vdash^{\nabla} M[x := N] : A}; \\
(\leq_{\nabla} L) & \dfrac{\Gamma, x{:}B \vdash^{\nabla} M : A \quad C \leq_{\nabla} B}{\Gamma, x{:}C \vdash^{\nabla} M : A}; \\
(\rightarrow L) & \dfrac{\Gamma, y{:}B \vdash^{\nabla} M : A \quad \Gamma \vdash^{\nabla} N : C \quad x \notin \Gamma}{\Gamma, x{:}C \rightarrow B \vdash^{\nabla} M[y := xN] : A}; \\
(\cap L) & \dfrac{\Gamma, x{:}A \vdash^{\nabla} M : B}{\Gamma, x{:}A \cap C \vdash^{\nabla} M : B}.
\end{array}
$$

The proofs in the following chapters will freely use the rules of the above Proposition.

[1] This means that all implications like $\Gamma \vdash^{\nabla} M : A \quad x \notin \Gamma \Rightarrow \Gamma, x{:}B \vdash^{\nabla} M : A$ are valid.

2
Properties

In this chapter we focus on properties of the Intersection Type Assignment Systems.

In Section 2.1 we discuss Inversion (Generation) Theorems. In Section 2.2 we characterize type theories validating various subject conversion properties. Together with the classical β and η reductions and expansions, we consider also meaningful restrictions, such as Plotkin's call-by-value reduction, etc.

2.1 Generation Theorems

Intersection type theories can be rather wild structures. Throughout this part, in the style of [CHDCL84] we shall isolate special syntactical properties of type theories which have significant consequences for the induced type assignment systems. As an important instance of this, we introduce properties of intersection type theories which allow to "reverse" some of the rules of the type assignment system $\lambda\cap^\nabla$, so as to achieve some form of Generation (or Inversion) Theorems (see Theorems 2.1.3 and 2.1.9). In order to show that this is a non-trivial issue, we start by a simple example which shows that rule $(\rightarrow\mathrm{E})$ cannot be reversed always, i.e. that if $\Gamma \vdash^\nabla MN : B$ it is not in general true that there exists A such that $\Gamma \vdash^\nabla M : A{\rightarrow}B$ and $\Gamma \vdash^\nabla N : A$.

Example 2.1.1 *Consider $\nabla = \mathcal{CD}$ and let $\Gamma = \{x{:}E, y{:}F\}$, where $E = (A{\rightarrow}B) \cap (C{\rightarrow}D)$ and $F = A \cap C$. Then*

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} x : E}{G \vdash^{\mathcal{CD}}_{\mathcal{B}} x : A{\rightarrow}B}(\leq_{\mathcal{CD}}) \quad \dfrac{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} y : F}{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} y : A}(\leq_{\mathcal{CD}})}{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} xy : B}(\rightarrow\mathrm{E}) \quad \dfrac{\dfrac{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} x : E}{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} x : C{\rightarrow}D}(\leq_{\mathcal{CD}}) \quad \dfrac{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} y : F}{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} y : C}(\leq_{\mathcal{CD}})}{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} xy : D}(\rightarrow\mathrm{E})}{\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} xy : B \cap D}(\cap\mathrm{I})$$

Nevertheless it is not possible to get a type A' such that $\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} x : A'{\rightarrow}(B \cap D)$ and $\Gamma \vdash^{\mathcal{CD}}_{\mathcal{B}} y : A'$.

In the general case we can only say that when $\Gamma \vdash^\nabla MN : A$, then there are a finite set I and types B_i, C_i, such that for each $i \in I$, $\Gamma \vdash^\nabla M : B_i{\rightarrow}C_i$, $\Gamma \vdash^\nabla N : B_i$, and moreover $\bigcap_{i\in I} C_i \leq_\nabla A$. Reasoning similarly on the rule $(\rightarrow\mathrm{I})$, one can conclude again that it cannot be reversed. More formally, we get the following theorem.

Notation 2.1.2 *When we write "... assume $A \not\sim_\nabla \Omega$ ..." we mean that this condition is always true when we deal with $\vdash^\nabla_{\mathcal{B}}$ and $\vdash^\nabla_\nu$, while it must be checked for $\vdash^\nabla_\Omega$. Similarly, the condition $\nu \not\leq_\nabla A$ must be checked just for $\vdash^\nabla_\nu$.*
In the following "iff ... for some I and $A_i ... \in \mathbf{T}^\nabla$" will be short for "iff there exists a finite set I and, for all $i \in I$, types $A_i ... \in \mathbf{T}^\nabla$ such that, for all $i \in I$ ...".

Theorem 2.1.3 (Generation Theorem I) *Let Σ^∇ be a type theory.*

i) Assume $A \not\sim_\nabla \Omega$. Then $\Gamma \vdash^\nabla MN : A$ iff $\Gamma \vdash^\nabla M : B_i{\rightarrow}C_i$, $\Gamma \vdash^\nabla N : B_i$, and $\bigcap_{i\in I} C_i \leq_\nabla A$, for some I and $B_i, C_i \in \mathbf{T}^\nabla$.

ii) Assume $\nu \not\leq_\nabla A$. Then $\Gamma \vdash^\nabla \lambda x.M : A$ iff $\Gamma, x{:}B_i \vdash^\nabla M : C_i$ and $\bigcap_{i\in I}(B_i{\rightarrow}C_i) \leq_\nabla A$, for some I and $B_i, C_i \in \mathbf{T}^\nabla$.

Proof. The proof of each (⇐) is easy. So we only treat (⇒). (i) By induction on derivations. The only interesting case is when $A \equiv A_1 \cap A_2$ and the last applied rule is ($\cap$I):

$$(\cap\mathrm{I}) \quad \frac{\Gamma \vdash^{\nabla} MN : A_1 \quad \Gamma \vdash^{\nabla} MN : A_2}{\Gamma \vdash^{\nabla} MN : A_1 \cap A_2}.$$

The condition $A \not\sim_{\nabla} \Omega$ implies that we cannot have both $A_1 \sim_{\nabla} \Omega$ and $A_2 \sim_{\nabla} \Omega$. We do the proof for $A_1 \not\sim_{\nabla} \Omega$ and $A_2 \not\sim_{\nabla} \Omega$, the other cases can be treated similarly. By induction there are I, B_i, C_i, J, D_j, E_j such that

$$\begin{array}{lll} \forall i \in I. & \Gamma \vdash^{\nabla} M : B_i \to C_i, & \Gamma \vdash^{\nabla} N : B_i, \\ \forall j \in J. & \Gamma \vdash^{\nabla} M : D_j \to E_j, & \Gamma \vdash^{\nabla} N : D_j, \\ \bigcap_{i \in I} C_i \leq_{\nabla} A_1 & \quad\& & \bigcap_{j \in J} E_j \leq_{\nabla} A_2. \end{array}$$

So we are done since $(\bigcap_{i \in I} C_i) \cap (\bigcap_{j \in J} E_j) \leq_{\nabla} A$.

(ii) The proof is very similar to the proof of (i). It is again by induction on derivations and again the only interesting case is when the last applied rule is ($\cap$I):

$$(\cap\mathrm{I}) \quad \frac{\Gamma \vdash^{\nabla} \lambda x.M : A_1 \quad \Gamma \vdash^{\nabla} \lambda x.M : A_2}{\Gamma \vdash^{\nabla} \lambda x.M : A_1 \cap A_2}.$$

The condition $\nu \not\leq_{\nabla} A$ implies that we cannot have both $\nu \leq_{\nabla} A_1$ and $\nu \leq_{\nabla} A_2$. We do the proof for $\nu \not\leq_{\nabla} A_1$ and $\nu \not\leq_{\nabla} A_2$. By induction there are I, B_i, C_i, J, D_j, E_j such that

$$\begin{array}{l} \forall i \in I.\ \Gamma, x{:}B_i \vdash^{\nabla} M : C_i, \forall j \in J.\ \Gamma, x{:}D_j \vdash^{\nabla} M : E_j, \\ \bigcap_{i \in I}(B_i \to C_i) \leq_{\nabla} A_1 \quad \& \quad \bigcap_{j \in J}(D_j \to E_j) \leq_{\nabla} A_2. \end{array}$$

So we are done since $(\bigcap_{i \in I}(B_i \to C_i)) \cap (\bigcap_{j \in J}(D_j \to E_j)) \leq_{\nabla} A$. The other two cases are trivial. For instance, if if $\nu \not\leq_{\nabla} A_1$ and $\nu \sim_{\nabla} A_2$, it is sufficient to take $\bigcap_{i \in I}(B_i \to C_i)$ above to conclude. □

Notice that as a consequence of previous theorem the *subformula property* (i.e. that if a term is typable then all its subterms are typable) holds with the only exception of an abstraction typed by the constant ν.

We can give sufficient conditions on type theories in order to get a standard inversion of rules (→I) and (→E) (see Theorem 2.1.9). First we need a definition.

Definition 2.1.4 *Let Σ^{∇} be a type theory.*

i) Σ^{∇} is beta *iff for all I, $A_i, B_i, C, D \in \mathbf{T}^{\nabla}$:*

$$\begin{array}{c} \bigcap_{i \in I}(A_i \to B_i) \leq_{\nabla} C \to D \ \&\ D \not\sim_{\nabla} \Omega \Rightarrow \\ \exists J \subseteq I,\ C \leq_{\nabla} \bigcap_{i \in J} A_i \ \&\ \bigcap_{i \in J} B_i \leq_{\nabla} D. \end{array}$$

ii) Σ^{∇} is ν-sound *iff*

$$\nu \not\sim_{\nabla} A \to B \text{ for all } A,\ B \in \mathbf{T}^{\nabla}.$$

We make a few comments on the previous definition. If we look at $\cap$ as representing $\sqcup$ and arrow types as representing step functions, then the condition for a type theory of being beta, is exactly the relation which holds between sups of step functions [GHK+80].The condition of being ν-sound is used to prevent both ν from being a redundant type and from assigning too many types to a λ-abstraction (assigning ν amounts exactly to discriminating an abstraction and nothing more). Notice that Σ^{∇} is trivially ν-sound when $\nu \notin \mathbf{C}^{\nabla}$.

Now we give sufficient conditions on type theories for being beta (Definition 2.1.5 and Lemma 2.1.6). Lemma 2.1.6 will allow us to conclude that all the intersection type theories of Figure 1.2 are beta (Theorem 2.1.7).

Definition 2.1.5 *A type theory* Σ^{∇} *is* strong beta *iff:*

i) each axiom or rule of ∇ *either belongs to* $\mathcal{BCD}$ *or has one of the following two shapes:*

- $a \leq a'$,
- $a \sim \bigcap_{i\in I}(a_i^{(1)} \to a_i^{(2)})$,

where $a, a', a_i^{(1)}, a_i^{(2)} \in \mathbf{C}^{\nabla}$*, and* $a, a', a_i^{(2)} \not\sim \Omega$ *for all* $i \in I$*;*

ii) for each $a \not\sim \Omega \in \mathbf{C}^{\nabla}$*, there is exactly one axiom in* ∇ *of the shape* $a \sim \bigcap_{i\in I}(a_i^{(1)} \to a_i^{(2)})$*;*

iii) ∇ *contains* $a \leq a'$ *iff:*

(a) $a \sim \bigcap_{i\in I}(a_i^{(1)} \to a_i^{(2)}) \in \nabla$,

(b) $a' \sim \bigcap_{j\in J}(a_j'^{(1)} \to a_j'^{(2)}) \in \nabla$*, and*

(c) for all $j \in J$ *there exists* $i \in I$ *such that* $a_j'^{(1)} \leq a_i^{(1)} \in \nabla$ *and* $a_i^{(2)} \leq a_j'^{(2)} \in \nabla$*.*

For example the theories $\Sigma^{\mathcal{HL}}$, $\Sigma^{\mathcal{S}c}$, $\Sigma^{\mathcal{P}a}$, $\Sigma^{\mathcal{CDZ}}$, $\Sigma^{\mathcal{HR}}$, and $\Sigma^{\mathcal{DHM}}$ are strong beta.

Lemma 2.1.6 *Each strong beta type theory is beta.*

Proof. By assumption for each constant $a \in \mathbf{C}^{\nabla}$ there is exactly one axiom stating that a is equivalent to an intersection of arrow types. We denote by $\bigcap_{l\in L^{(a)}}(b^{(a)} \to d^{(a)})$ such an intersection. Moreover notice that the most general form of an intersection type is a finite intersection of arrow types and type constants. We can prove by simultaneous induction on the definition of $\leq_{\nabla}$ two statements, the first of which implies the beta condition:

- if $(\bigcap_{i\in I}(A_i\to B_i))\cap(\bigcap_{h\in H} a_h) \leq_\nabla (\bigcap_{j\in J}(C_j\to D_j))\cap(\bigcap_{k\in K} c_k)$, and $D_j\not\sim_\nabla\Omega$, then there exist $I'\subseteq I$, $H'\subseteq H$ and, for all $h\in H'$, $L^{(a_h)'}\subseteq L^{(a_h)}$ such that $C_j \leq_\nabla (\bigcap_{i\in I'} A_i)\cap(\bigcap_{h\in H'}(\bigcap_{l\in L^{(a_h)'}} b_l^{(a_h)}))$ and $(\bigcap_{i\in I'} B_i)\cap(\bigcap_{h\in H'}(\bigcap_{l\in L^{(a_h)'}} d_l^{(a_h)})) \leq_\nabla D_j$;
- if $(\bigcap_{i\in I}(A_i\to B_i))\cap(\bigcap_{h\in H} a_h) \leq_\nabla (\bigcap_{j\in J}(C_j\to D_j))\cap(\bigcap_{k\in K} c_k)$, and $c_k\not\sim_\nabla\Omega$, then for each $m\in L^{(b_k)}$ there exist $I'\subseteq I$, $H'\subseteq H$ and, for all $h\in H'$, $L^{(a_h)'}\subseteq L^{(a_h)}$ such that $b_m^{(c_k)} \leq_\nabla (\bigcap_{i\in I'} A_i)\cap(\bigcap_{h\in H'}(\bigcap_{l\in L^{(a_h)'}} b_l^{(a_h)}))$ and $(\bigcap_{i\in I'} B_i)\cap(\bigcap_{h\in H'}(\bigcap_{l\in L^{(a_h)'}} d_l^{(a_h)})) \leq_\nabla d_m^{(c_k)}$.

□

Theorem 2.1.7

i) All the type theories of Figure 1.2 are beta.

ii) All the type theories of Figure 1.2 are ν-sound.

Proof. (i) For $\nabla\in\{\mathcal{CD},\mathcal{CDV},\mathcal{EHR},\mathcal{AO},\mathcal{BCD},\mathcal{P}l,\mathcal{E}n\}$ we can prove by induction on the definition of $\leq_\nabla$, that if $(\bigcap_{i\in I}(A_i\to B_i))\cap(\bigcap_{h\in H} a_h) \leq_\nabla (\bigcap_{j\in J}(C_j\to D_j))\cap(\bigcap_{k\in K} b_k)$ then $\forall j\in J$ if $D_j\not\sim_\nabla\Omega$ then $\exists I'\subseteq I$ such that $C_j\leq_\nabla\bigcap_{i\in I'} A_i$, and $\bigcap_{i\in I'} B_i\leq_\nabla D_j$.
The theories Σ^∇ for $\nabla\in\{\mathcal{HL},\mathcal{S}c,\mathcal{P}a,\mathcal{CDZ},\mathcal{HR},\mathcal{DHM}\}$ are beta by Lemma 2.1.6.
(ii) For $\Sigma^{\mathcal{EHR}}$ one can easily show, by induction on $\leq_{\mathcal{EHR}}$, that $\nu\leq_{\mathcal{EHR}} A$ implies that A is an intersection of ν. □

Using the notations introduced in Definition 2.1.4, we can give now a rather powerful version of a Generation Theorem for $\lambda\cap^\nabla$. Special cases of this theorem have been previously proved in [BCDC83, CHDCL84, CDCZ87, HRDR92, EHRDR92].

Notation 2.1.8 *We write "the type theory Σ^∇ validates ∇'" to mean that all axioms and rules of ∇' are admissible in Σ^∇.*

Theorem 2.1.9 (Generation Theorem II) *Let Σ^∇ be a type theory.*

i) Assume $A\not\sim_\nabla\Omega$. Then $\Gamma\vdash^\nabla x:A$ iff $(x{:}B)\in\Gamma$ and $B\leq_\nabla A$ for some $B\in\mathbf{T}^\nabla$.

ii) Assume $A\not\sim_\nabla\Omega$ and let Σ^∇ validate $\mathcal{CDV}$. Then $\Gamma\vdash^\nabla MN:A$ iff $\Gamma\vdash^\nabla M:B\to A$, and $\Gamma\vdash^\nabla N:B$ for some $B\in\mathbf{T}^\nabla$.

iii) Let Σ^∇ be ν-sound and beta. Then $\Gamma\vdash^\nabla \lambda x.M:B\to C$ iff $\Gamma,x{:}B\vdash^\nabla M:C$.

PROOF. The proof of each ($\Leftarrow$) is easy. So we only treat ($\Rightarrow$).

(i) Easy by induction on derivations, since only the axioms (Ax), (Ax-Ω), and the rules ($\cap$I), ($\leq_\nabla$) can be applied. Notice that the condition $A \not\sim_\nabla \Omega$ implies that $\Gamma \vdash^\nabla x : A$ cannot be obtained just using axioms (Ax-Ω).

(ii) Let I, B_i, C_i be as in Theorem 2.1.3(i). Applying rule ($\cap$I) to $\Gamma \vdash^\nabla M : B_i \to C_i$ we can derive $\Gamma \vdash^\nabla M : \bigcap_{i\in I}(B_i \to C_i)$, so by ($\leq_\nabla$) we have $\Gamma \vdash^\nabla M : \bigcap_{i\in I} B_i \to \bigcap_{i\in I} C_i$, since

$$\bigcap_{i\in I}(B_i \to C_i) \leq_\nabla \bigcap_{i\in I}(\bigcap_{i\in I} B_i \to C_i) \leq_\nabla \bigcap_{i\in I} B_i \to \bigcap_{i\in I} C_i$$

by rule (η) and axiom ($\to$-$\cap$). We can choose $B = \bigcap_{i\in I} B_i$ and conclude $\Gamma \vdash^\nabla M : B \to A$ since $\bigcap_{i\in I} C_i \leq_\nabla A$.

(iii) The case $C \sim_\nabla \Omega$ is trivial for $\lambda\cap_\Omega^\nabla$. Otherwise, by the ν-soundness of Σ^∇ we cannot have $\nu \sim_\nabla B \to C$. Let I, B_i, C_i be as in Theorem 2.1.3(ii), where $A \equiv B \to C$. Then, $\bigcap_{i\in I}(B_i \to C_i) \leq_\nabla B \to C$ implies that there exists a $J \subseteq I$ such that $B \leq_\nabla \bigcap_{i\in J} B_i$ and $\bigcap_{i\in J} C_i \leq_\nabla C$, since Σ^∇ is beta. From $\Gamma, x{:}B_i \vdash^\nabla M : C_i$ we can derive $\Gamma, x{:}B \vdash^\nabla M : C_i$ using rule ($\leq_\nabla L$), so by ($\cap$I) we have $\Gamma, x{:}B \vdash^\nabla M : \bigcap_{i\in J} C_i$. Finally applying rule ($\leq_\nabla$) we can conclude $\Gamma, x{:}B \vdash^\nabla M : C$. □

We are now in the position of proving that, for the theories $\Sigma^{\mathcal{CD}}$ and $\Sigma^{\mathcal{CDV}}$, we can replace rule ($\leq_\nabla$) (for $\nabla = \mathcal{CD}$ and $\nabla = \mathcal{CDV}$, respectively) by more perspicuous rules, still obtaining basic type assignment systems equivalent to the original ones.

Proposition 2.1.10 *i) Let $\vdash$ denote derivability in the system obtained from $\vdash_{\mathcal{B}}^{\mathcal{CD}}$ by replacing rule ($\leq_{\mathcal{CD}}$) by the rules*

$$(\cap\mathrm{E}) \quad \frac{\Gamma \vdash M : A \cap B}{\Gamma \vdash M : A} \qquad \frac{\Gamma \vdash M : A \cap B}{\Gamma \vdash M : B}.$$

Then $\Gamma \vdash_{\mathcal{B}}^{\mathcal{CD}} M : A$ iff $\Gamma \vdash M : A$.

ii) Let $\vdash$ denote derivability in the system obtained from $\vdash_{\mathcal{B}}^{\mathcal{CDV}}$ by replacing rule ($\leq_{\mathcal{CDV}}$) by the rules ($\cap$E) and

$$(R\text{-}\eta) \quad \frac{\Gamma \vdash \lambda x.Mx : A}{\Gamma \vdash M : A} \quad \textit{if } x \notin \mathrm{FV}(M).$$

Then $\Gamma \vdash_{\mathcal{B}}^{\mathcal{CDV}} M : A$ iff $\Gamma \vdash M : A$.

PROOF. (i) It is immediate to verify that rule ($\cap$E) is derivable in $\vdash_{\mathcal{B}}^{\mathcal{CD}}$ using the axioms (incl$_L$) and (incl$_R$). The converse can be proved by induction on the derivation of a judgement in $\vdash_{\mathcal{B}}^{\mathcal{CD}}$. The only non trivial case arises when the last rule is an application of ($\leq_{\mathcal{CD}}$). This is proved by induction on the definition of $\leq_{\mathcal{CD}}$. The cases of Axioms (idem) (incl$_L$) and (incl$_R$) follow respectively from rules ($\cap$I) and ($\cap$E). The other cases are trivial.

(ii) ($\Leftarrow$) By (i) rule ($\cap$E) is derivable. Therefore, we are left to prove that (R-η) is admissible. Notice that $\Omega \notin \mathbf{C}^{\mathcal{CDV}}$ and $\nu \notin \mathbf{C}^{\mathcal{CDV}}$. By Theorem 2.1.3(ii) we have $\Gamma, x{:}B_i \vdash^{\mathcal{CDV}}_{\mathcal{B}} Mx : C_i$, and $\bigcap_{i\in I}(B_i{\to}C_i) \leq_{\mathcal{CDV}} A$ for some I and $B_i, C_i \in \mathbf{T}^{\mathcal{CDV}}$. Moreover, since $\Sigma^{\mathcal{CDV}}$ validates $\mathcal{CDV}$, we get $\Gamma, x{:}B_i \vdash^{\mathcal{CDV}}_{\mathcal{B}} x : D_i$ and $\Gamma, x{:}B_i \vdash^{\mathcal{CDV}}_{\mathcal{B}} M : D_i{\to}C_i$ for some $D_i \in \mathbf{T}^{\mathcal{CDV}}$, by Lemma 2.1.9(ii). Now from Lemma 2.1.9(i) $B_i \leq_{\mathcal{CDV}} D_i$, which implies $D_i{\to}C_i \leq_{\mathcal{CDV}} B_i{\to}C_i$ by (η). So we can derive $\Gamma, x{:}B_i \vdash^{\mathcal{CDV}}_{\mathcal{B}} M : \bigcap_{i\in I}(B_i{\to}C_i)$ by rules ($\leq_{\mathcal{CDV}}$) and ($\cap$I). Lastly by rules ($\leq_{\mathcal{CDV}}$) and (*strengthening*) (recall that by hypothesis $x \notin \mathrm{FV}(M)$) we conclude $\Gamma \vdash^{\mathcal{CDV}}_{\mathcal{B}} M : A$.

($\Rightarrow$) We follow the same line of reasoning as in (i). The only extra cases we have to consider are those corresponding to applications of the axiom ($\to$-$\cap$) and rule (η) in the derivation of the second premise of rule ($\leq_{\mathcal{CDV}}$). In the first case we have

$$\begin{array}{lll}
\Gamma \vdash M : (A{\to}B) \cap (A{\to}C) & \Rightarrow & \Gamma \vdash M : A{\to}B \text{ and } \Gamma \vdash M : A{\to}C \\
 & & \quad \text{by rule } (\cap\mathrm{E}) \\
 & \Rightarrow & \Gamma, x{:}A \vdash Mx : B \text{ and } \Gamma, x{:}A \vdash Mx : C \\
 & & \quad \text{by rule } (\to\mathrm{E}) \text{ where } x \text{ is fresh,} \\
 & \Rightarrow & \Gamma, x{:}A \vdash Mx : B \cap C \\
 & & \quad \text{by rule } (\cap\mathrm{I}) \\
 & \Rightarrow & \Gamma \vdash \lambda x.Mx : A{\to}B \cap C \\
 & & \quad \text{by rule } (\to\mathrm{I}) \\
 & \Rightarrow & \Gamma \vdash M : A{\to}B \cap C \\
 & & \quad \text{by rule } (R\text{-}\eta).
\end{array}$$

For the second case, assuming $A' \leq_{\mathcal{CDV}} A$ and $B \leq_{\mathcal{CDV}} B'$, we have

$$\begin{array}{lll}
\Gamma \vdash M : A{\to}B & \Rightarrow & \Gamma, x{:}A \vdash Mx : B \\
 & & \quad \text{by rule } (\to\mathrm{E}) \text{ where } x \text{ is fresh,} \\
 & \Rightarrow & \Gamma, x{:}A \vdash Mx : B' \\
 & & \quad \text{by induction} \\
 & \Rightarrow & \Gamma, x{:}A' \vdash Mx : B' \\
 & & \quad \text{by rule } (\leq_{\mathcal{CDV}}\ \mathrm{L}) \\
 & \Rightarrow & \Gamma \vdash \lambda x.Mx : A'{\to}B' \\
 & & \quad \text{by rule } (\to\mathrm{I}) \\
 & \Rightarrow & \Gamma \vdash M : A'{\to}B' \\
 & & \quad \text{by rule } (R\text{-}\eta).
\end{array}$$

□

2.2 Subject reduction and expansion

We address now the important issue type preservation under conversion. There are various notions of conversion worthwhile considering: β-expansion, β-reduction, η-expansion, η-reduction, together with some of their *restrictions*. Interesting examples of restrictions are the set of Plotkin's β_v-redexes, the set of β-I-redexes, the set of β-N-redexes. Let us recall first the definitions of these redexes.

Definition 2.2.1 *i) A redex $(\lambda x.M)N$ is a β_v-*redex *if N is a variable or an abstraction ([Plo75]).*

*ii) A redex $(\lambda x.M)N$ is a β-*I-redex *if $x \in \mathrm{FV}(M)$ ([CF58]).*

*iii) A redex $(\lambda x.M)N$ is a β-*N-redex *if $x \notin \mathrm{FV}(M)$ and N is either a variable or a closed strongly normalizing term ([HL99]).*

We introduce rules of the form

$$(\text{R-}exp)\quad \frac{M \to_R N \qquad \Gamma \vdash^{\nabla} N : A}{\Gamma \vdash^{\nabla} M : A}$$

$$(\text{R-}red)\quad \frac{M \to_R N \qquad \Gamma \vdash^{\nabla} M : A}{\Gamma \vdash^{\nabla} N : A}$$

where $\to_R$ denotes the reduction relation obtained by restricting the contraction to the set of R-redexes.

In particular we characterise those type assignment systems for which types are preserved under various notions of reduction and expansion. These properties is crucial in chapters where λ-models are induced by type theories.

Theorem 2.2.2 *(Characterisation of β-conversion)*

i) Assume $\Gamma \vdash^{\nabla} M[x := N] : A$. Then $\Gamma \vdash^{\nabla} (\lambda x.M)N : A$ iff N is typable in the context Γ.

ii) (β-expansion) Rule (β-exp) is admissible in $\lambda\cap^{\nabla}$ iff (i) holds for all pairs of β-redexes and corresponding β-contracta.

iii) (β-reduction) Rule (β-red) is admissible in $\lambda\cap^{\nabla}$ iff rule ($\to$I) can be reversed, i.e. for all Γ, M, A, B:

$$\Gamma \vdash^{\nabla} \lambda x.M : B \to A \;\Rightarrow\; \Gamma, x{:}B \vdash^{\nabla} M : A.$$

Proof. (i) ($\Rightarrow$) Clearly if N is not typable in the context Γ then also $(\lambda x.M)N$ has no type in Γ by Theorem 2.1.3(i).

($\Leftarrow$) Let D be a deduction of $\Gamma \vdash^{\nabla} M[x := N] : A$ and let $\Gamma_i \vdash^{\nabla} N : B_i$ for $i \in I$ be all the statements in D whose subject is N. Without loss of generality we can assume that x does not occur in Γ.

If I is nonempty, notice that $\Gamma \subseteq \Gamma_i$ but $\Gamma \restriction \mathrm{FV}(N) = \Gamma_i \restriction \mathrm{FV}(N)$ (recall that (*weakening*) and (*strengthening*) are not rules of the system, they only are admissible rules, so ∇-bases in a derivation can only be modified by rule ($\to$I)). So using rules (*strengthening*) and ($\cap$I), we have that $\Gamma \vdash^{\nabla} N : \bigcap_{i\in I} B_i$. Moreover, one can easily

see, by induction on M, that $\Gamma, x{:}\bigcap_{i\in I} B_i \vdash^\nabla M : A$. Thus, by rule $(\to\mathrm{I})$, we have $\Gamma \vdash^\nabla \lambda x.M : \bigcap_{i\in I} B_i \to A$. Hence, by $(\to\mathrm{E})$ we can conclude $\Gamma \vdash^\nabla (\lambda x.M)N : A$.
If I is empty, we get from D a derivation of $\Gamma \vdash^\nabla M : A$ by replacing each N by x. By assumption there exists a B such that $\Gamma \vdash^\nabla N : B$. By rule (*weakening*) we get $\Gamma, x{:}B \vdash^\nabla M : A$ and we can conclude as in the previous case.

(ii) The proof by a double induction on $\to_\beta$ and on derivations is straigthforward.

(iii) $(\Rightarrow)$ Assume $\Gamma \vdash^\nabla \lambda x.M : B \to A$, which implies $\Gamma, y{:}B \vdash^\nabla (\lambda x.M)y : A$ by rule $(\to\mathrm{E})$ for a fresh y. The admissibility of rule (β-*red*) gives us $\Gamma, y{:}B \vdash^\nabla M[x := y] : A$. Hence $\Gamma, x{:}B \vdash^\nabla M : A$.

$(\Leftarrow)$ It suffices to show that $\Gamma \vdash^\nabla (\lambda x.M)N : A$ implies $\Gamma \vdash^\nabla M[x := N] : A$. The case $A \sim_\nabla \Omega$ is trivial for $\lambda\cap^\nabla_\Omega$. Otherwise by Theorem 2.1.3(ii), $\Gamma \vdash^\nabla \lambda x.M : B_i \to C_i$, $\Gamma \vdash^\nabla N : B_i$ and $\bigcap_{i\in I} C_i \leq_\nabla A$, for some I, B_i and C_i. By hypothesis we get $\Gamma, x{:}B_i \vdash^\nabla M : C_i$. Then $\Gamma \vdash^\nabla M[x := N] : C_i$ follows by an application of rule (*cut*), and so we can conclude $\Gamma \vdash^\nabla M[x := N] : A$ using rules $(\cap\mathrm{I})$ and $(\leq_\nabla)$. □

As an immediate consequence of Theorem 2.1.9(iii) and of Theorem 2.2.2(iii) we get

Corollary 2.2.3 *If Σ^∇ is ν-sound and beta then rule (β-red) is admissible in $\lambda\cap^\nabla$.*

The rather contrived statement given in Theorem 2.2.2(ii) above is immediately met in $\lambda\cap^\nabla_\Omega$, in $\lambda\cap^\nabla_{\mathcal{B}}$ when $x \in \mathrm{FV}(M)$, and in $\lambda\cap^\nabla_\nu$ when N is an abstraction. For restricted β-expansions we can give the following simple conditions on type theories.

Corollary 2.2.4 *i) Rule (β-I-exp) is admissible in all $\lambda\cap^\nabla_{\mathcal{B}}$ and $\lambda\cap^\nabla_\Omega$, but in no $\lambda\cap^\nabla_\nu$.*

ii) Rule (β-N-exp) is admissible in all $\lambda\cap^\nabla$, provided that we consider only ∇-bases which assign types to all variables, i.e. ∇-bases Γ such that $x \in \Gamma$ for all $x \in Var$.

iii) Rule (β_v-exp) is admissible in all $\lambda\cap^\nabla_\Omega$ and $\lambda\cap^\nabla_\nu$, provided that in this last case we consider only ∇-bases Γ such that $\Gamma \supseteq \{x{:}\nu \mid x \in Var\}$. It is admissible in no $\lambda\cap^\nabla_{\mathcal{B}}$.

iv) Rule (β-exp) is admissible in all $\lambda\cap^\nabla_\Omega$, but in no $\lambda\cap^\nabla_{\mathcal{B}}$ and in no $\lambda\cap^\nabla_\nu$.

Proof. The positive statements of the four items but (ii) follow from (the proof of) Theorem 2.2.2. Item (ii) depends also on Theorem 4.2.2(i) where we will prove that each strongly normalizing term is typable in all type theories from a suitable ∇-basis. So all closed strongly normalizing terms are typable in all type theories starting from the empty ∇-basis. Moreover, by (Ax), all term variables are are typeable in each ∇-basis which assigns types to all term variables.
An example showing that (β-I-*exp*) is not admissible in $\lambda\cap^\nabla_\nu$ is $\vdash^\nabla_\nu \lambda x.z : \nu$ and $\not\vdash^\nabla_\nu (\lambda yx.y)z : \nu$.

For the non admissibility of (β_v-*exp*) in $\lambda\cap^{\nabla}_{\mathcal{B}}$ and of (β-*exp*) in $\lambda\cap^{\nabla}_{\mathcal{B}}$ and $\lambda\cap^{\nabla}_{\nu}$, notice that we can always derive $\vdash^{\nabla} \lambda x.x : A{\rightarrow}A$, but by the Generation Theorems I and II (Theorems 2.1.3(i) and 2.1.9(i)) we cannot derive the same type for $(\lambda yx.x)z$ from the empty ∇-basis without using (Ax-Ω). □

Notice that there are β-redexes that, without being β-I-redexes or β-N-redexes, are typable whenever their contracta are. As an example take $(\lambda x.(yz))(yz)$.

In order to characterise the admissibility of rule (η-*exp*), we need to introduce a further condition on type theories. This condition is necessary and sufficient to derive from the ∇-basis $x{:}a$ the type a for $\lambda y.xy$, as we will show in the proof of Theorem 2.2.7.

Definition 2.2.5 *A type theory Σ^{∇} is* eta *iff for all $a \in \mathbf{C}^{\nabla}$ either*

i) $\bigcap_{i\in I}(A_i{\rightarrow}B_i)\leq_{\nabla}a$ *for some $I, A_i, B_i \in \mathbf{T}^{\nabla}$ such that $B_i\sim_{\nabla}\Omega$ for all $i \in I$, or*

ii) $\nu\leq_{\nabla}a$*, or*

iii) there exist non empty families of types $\{A_i, B_i\}_{i\in I}$, $\{D_{i,j}, E_{i,j}\}_{j\in J_i}$ in $\mathbf{T}^{\nabla}$ such that

$$\begin{array}{c}\bigcap_{i\in I}(A_i{\rightarrow}B_i)\leq_{\nabla}a \leq_{\nabla} \bigcap_{i\in I}(\bigcap_{j\in J_i}(D_{i,j}{\rightarrow}E_{i,j}))\\ \forall i \in I.B_i\not\sim_{\nabla}\Omega \;\Rightarrow\; A_i \leq_{\nabla} \bigcap_{j\in J_i} D_{i,j} \;\&\; \bigcap_{j\in J_i} E_{i,j} \leq_{\nabla} B_i.\end{array}$$

It is easy to verify that if Σ^{∇} validates $\mathcal{CDV}$ then the condition of the above definition simplifies to the requirement that all atomic types are either bigger than $\Omega \rightarrow \Omega$ or than ν or they are equivalent to a suitable intersection of arrow types, namely

$$\forall a \in \mathbf{C}^{\nabla}.\Omega \rightarrow \Omega \leq_{\nabla} a \text{ or } \nu \leq_{\nabla} a \text{ or } \exists I, \{A_i, B_i\}_{i\in I}.\ \bigcap_{i\in I}(A_i{\rightarrow}B_i)\sim_{\nabla}a.$$

The next proposition singles out the intersection type theories Σ^{∇} of Figure 1.2 which are eta.

Proposition 2.2.6 *If $\nabla \in \{\mathcal{HL}, \mathcal{EHR}, \mathcal{AO}, \mathcal{S}c, \mathcal{P}a, \mathcal{CDZ}, \mathcal{HR}\}$, then Σ^{∇} is an eta theory.*

The characterisation of η-conversion can be given directly on the type theories.

Theorem 2.2.7 *(Characterisation of η-conversion)*

i) Rule (η-exp) is admissible in $\lambda\cap^{\nabla}$ iff Σ^{∇} is eta.

ii) Rule (η-red) is admissible in $\lambda\cap^{\nabla}_{\mathcal{B}}$ iff Σ^{∇} validates $\mathcal{CDV}$, in $\lambda\cap^{\nabla}_{\Omega}$ iff Σ^{∇} validates $\mathcal{BCD}$, and it is admissible in no $\lambda\cap^{\nabla}_{\nu}$.

Proof. (i) ($\Rightarrow$) Let $a \in \mathbf{C}^\nabla$ be a constant that does not satisfy the first two conditions in Definition 2.2.5. We can derive $\{x{:}a\} \vdash^\nabla x : a$. To derive $\{x{:}a\} \vdash^\nabla \lambda y.xy : a$ by Theorem 2.1.3(ii) we have $\{x{:}a, y{:}A_i\} \vdash^\nabla xy : B_i$ and $\bigcap_{i\in I}(A_i{\to}B_i) \leq_\nabla a$ for some I, $A_i, B_i \in \mathbf{T}^\nabla$. Let now $B_i \not\sim_\nabla \Omega$ for some $i \in I$. Then, by Theorem 2.1.3(i), we have that there exists J_i and $D_{i,j}, E_{i,j}$, such that for each $j \in J_i$ $\{x{:}a, y{:}A_i\} \vdash^\nabla x : D_{i,j}{\to}E_{i,j}$, $\{x{:}a, y{:}A_i\} \vdash^\nabla y : D_{i,j}$, and $\cap_{j\in J_i} E_{i,j} \leq_\nabla B_i$ for some $J_i, D_{i,j}, E_{i,j}$. By Theorem 2.1.9(i) we have $a \leq_\nabla D_{i,j}{\to}E_{i,j}$ and $A_i \leq_\nabla D_{i,j}$ for all $i \in I$ and $j \in J_i$. So we conclude

$$\begin{array}{c}\bigcap_{i\in I}(A_i{\to}B_i) \leq_\nabla a \leq_\nabla \bigcap_{i\in I}(\bigcap_{j\in J_i}(D_{i,j}{\to}E_{i,j}))\\ \forall i \in I.\ B_i \not\sim_\nabla \Omega \ \Rightarrow\ A_i \leq_\nabla \bigcap_{j\in J_i} D_{i,j} \ \&\ \bigcap_{j\in J_i} E_{i,j} \leq_\nabla B_i.\end{array}$$

($\Leftarrow$) The proof that $\Gamma \vdash^\nabla M : A$ implies $\Gamma \vdash^\nabla \lambda x.Mx : A$, where x is fresh, is by induction on the structure of A. The only interesting case is if A is a type constant a for some $a \in \mathbf{C}^\nabla$ such that:

$$\begin{array}{c}\bigcap_{i\in I}(A_i{\to}B_i) \leq_\nabla a \leq_\nabla \bigcap_{i\in I}(\bigcap_{j\in J_i}(D_{i,j}{\to}E_{i,j}))\\ \forall i \in I. B_i \not\sim_\nabla \Omega \ \Rightarrow\ A_i \leq_\nabla \bigcap_{j\in J_i} D_{i,j} \ \&\ \bigcap_{j\in J_i} E_{i,j} \leq_\nabla B_i.\end{array}$$

By rule ($\leq_\nabla$) we can derive $\Gamma \vdash^\nabla M : D_{i,j}{\to}E_{i,j}$ for all $i \in I, j \in J_i$, and so $\Gamma, x{:}D_{i,j} \vdash^\nabla Mx : E_{i,j}$ by rule ($\to$E). From ($\leq_\nabla$ L), ($\cap$I) and ($\leq_\nabla$) in the case $B_i \not\sim_\nabla \Omega$, or (Ax-$\Omega$) in the case $B_i \sim_\nabla \Omega$, we get $\Gamma, x{:}A_i \vdash^\nabla Mx : B_i$ and this implies $\Gamma \vdash^\nabla \lambda x.Mx : A_i{\to}B_i$ using rule ($\to$I). So we can conclude by ($\cap$I) and ($\leq_\nabla$) that $\Gamma \vdash^\nabla \lambda x.Mx : a$. The other cases are easy.

(ii) ($\Rightarrow$) Let us assume that Σ^∇ does not validate axiom ($\to$-$\cap$), i.e. that there are types A, B, C such that $(A{\to}B) \cap (A{\to}C) \not\leq_\nabla A{\to}B \cap C$. We can derive $\{x{:}(A{\to}B) \cap (A{\to}C)\} \vdash^\nabla_{\mathcal{B}} \lambda y.xy : A{\to}B \cap C$ using ($\leq_\nabla$), ($\to$E), ($\cap$I), and ($\to$I), but $x : A{\to}B \cap C$ cannot be derived from $x{:}(A{\to}B) \cap (A{\to}C)$ by Theorem 2.1.9(i). Now suppose that Σ^∇ does not validate rule (η), i.e. that there are types A, B, C, D such that $A \leq_\nabla B$ and $C \leq_\nabla D$ but $B{\to}C \not\leq_\nabla A{\to}D$. We can derive $\{x{:}B{\to}C\} \vdash^\nabla_{\mathcal{B}} \lambda y.xy : A{\to}D$ using ($\leq_\nabla$), ($\to$E), and ($\to$I), but $\{x{:}B{\to}C\} \not\vdash^\nabla_{\mathcal{B}} x : A{\to}D$ by Theorem 2.1.9(i).
If $\Omega \in \mathbf{C}^\nabla$ we get $\{x{:}\Omega\} \vdash^\nabla_\Omega \lambda y.xy : \Omega{\to}\Omega$ by axiom (Ax-Ω) and rule ($\to$I), hence $\{x{:}\Omega\} \vdash^\nabla_\Omega x : \Omega{\to}\Omega$. By Theorem 2.1.9(i) it follows $\Omega \leq_\nabla \Omega{\to}\Omega$, i.e. Σ^∇ must validate axiom (Ω-η).
If $\nu \in \mathbf{C}^\nabla$ we get $\vdash^\nabla_\nu \lambda y.xy : \nu$ by axiom (Ax-ν), but we cannot derive $x : \nu$ from the empty ∇-basis by Theorem 2.1.9(i).

($\Leftarrow$) We prove that under the given conditions on type theories $\Gamma \vdash^\nabla \lambda x.Mx : A$ and $x \notin \mathrm{FV}(M)$ imply $\Gamma \vdash^\nabla M : A$. We give the proof for $\lambda\cap^\nabla_\Omega$, that one for $\lambda\cap^\nabla_{\mathcal{B}}$ being similar and simpler. By Theorem 2.1.3(ii) $\Gamma \vdash^\nabla_\Omega \lambda x.Mx : A$ implies that there are I, B_i, C_i such that $\Gamma, x{:}B_i \vdash^\nabla_\Omega Mx : C_i$ and $\bigcap_{i\in I}(B_i{\to}C_i) \leq_\nabla A$. If for some i we get $C_i \sim_\nabla \Omega$, then we can obtain $B_i{\to}C_i \sim_\nabla \Omega$ by axiom (Ω-η) and rule (η). Therefore we can forget those $B_i{\to}C_i$. Otherwise $\Gamma, x{:}B_i \vdash^\nabla_\Omega Mx : C_i$ implies by Theorem 2.1.9(ii) and (*strengthening*) that $\Gamma \vdash^\nabla_\Omega M : D_i{\to}C_i$, and $\Gamma, x{:}B_i \vdash^\nabla_\Omega x : D_i$, for some D_i. By

Theorem 2.1.9(i) we get $B_i \leq_\nabla D_i$, so we can derive $\Gamma \vdash^\nabla_\Omega M : B_i \to C_i$ using rule $(\leq_\nabla)$, since $D_i \to C_i \leq_\nabla B_i \to C_i$ by rule (η). Rule $(\cap\mathrm{I})$ implies $\Gamma \vdash^\nabla_\Omega M : \bigcap_{i\in I}(B_i \to C_i)$. So we can conclude $\Gamma \vdash^\nabla_\Omega M : A$ using rule $(\leq_\nabla)$. $\square$

Generation theorems which were proven for each specific systems, have been given in the general format. The following table summarises our results on admissibility of rules.

Table 2.1: Admissibility of typing rules in type assignment systems

System	β-exp	β-I-exp	β-N-exp	β_v-exp	η-exp	η-red
$\lambda\cap^\nabla_\mathcal{B}$	No	Yes	(1)	No	Σ^∇ is eta	Σ^∇ validates $\mathcal{CDV}$
$\lambda\cap^\nabla_\Omega$	Yes	Yes	Yes	Yes	Σ^∇ is eta	Σ^∇ validates $\mathcal{BCD}$
$\lambda\cap^\nabla_\nu$	No	No	(1)	(2)	Σ^∇ is eta	No

(1) All ∇-bases Γ are such that $x \in \Gamma$ for all $x \in \mathit{Var}$.

(2) All ∇-bases Γ are such that $\Gamma \supseteq \{x{:}\nu \mid x \in \mathit{Var}\}$.

3

Approximation Theorems

The present chapter is organized as follows. In Section 3.1 we introduce approximante normal forms. In section 3.2 we prove the Approximation Theorem using a Kripke interpretation of intersection types in which the bases play the role of worlds. In Section 3.3 we give some applications of the Approximation Theorems.

3.1 Approximate normal forms

For many of the type theories of Figure 1.2 we introduce appropriate notions of *approximants* which agree with the λ-theories of different models and therefore also with the type theories describing these models. Then we will prove that all types of an approximant of a given term (with respect to the appropriate notion of approximants) are also types of the given term. Finally we show the converse, namely that the types which can be assigned to a term can also be assigned to at least one approximant of that term. Hence a type can be derived for a term *if and only if* it can be derived for an approximant of that term.

In this chapter we consider two extensions of λ-calculus both obtained by adding one constant. The first one is the well known language $\lambda\bot$, see [Bar84]. The other extension is obtained by adding Φ and it is discussed in [HRDR92].

Definition 3.1.1 *i) The set $\Lambda\bot$ of $\lambda\bot$-terms is obtained by adding the constant $\bot$ to the formation rules of λ-terms.*

ii) The set $\Lambda\Phi$ of $\lambda\Phi$-terms is obtained by adding the constant Φ to the formation rules of λ-terms.

Notice that axiom (Ax-Ω) gives $\Gamma \vdash^{\nabla} \bot : \Omega$.

We consider two mappings (α and β) from λ-terms to $\lambda\bot$-terms and one mapping (γ) from λ-terms to $\lambda\Phi$-terms. These mappings differ in the translation of β-redexes. Clearly the values of these mappings are β-irreducible terms, i.e. normal forms for an extended language. As usual we call them *approximate normal forms.*

Definition 3.1.2 *The mappings $\alpha : \Lambda \to \Lambda\bot, \beta : \Lambda \to \Lambda\bot, \gamma : \Lambda \to \Lambda\Phi$ are inductively defined by:*

- $\dagger(\lambda\vec{x}.yM_1 \ldots M_m) = \lambda\vec{x}.y \dagger(M_1) \ldots \dagger(M_m)$ *for* $\dagger \in \{\alpha, \beta, \gamma\}$ *;*
- $\alpha(\lambda\vec{x}.(\lambda y.R)NM_1 \ldots M_m) = \perp$ *;*
- $\beta(\lambda\vec{x}.(\lambda y.R)NM_1 \ldots M_m) = \lambda\vec{x}.\perp$ *;*
- $\gamma(\lambda\vec{x}.(\lambda y.R)NM_1 \ldots M_m) = \lambda\vec{x}.\Phi\gamma(\lambda y.R)\gamma(N)\gamma(M_1)\ldots\gamma(M_m)$

where $m \geq 0$*.*

In order to give the appropriate Approximation Theorem we will use the mapping α for the type assignment systems $\lambda\cap_\Omega^{\mathcal{BCD}}$, $\lambda\cap_\Omega^{\mathcal{S}c}$, $\lambda\cap_\Omega^{\mathcal{CDZ}}$, $\lambda\cap_\Omega^{\mathcal{DHM}}$ the mapping β for the type assignment system $\lambda\cap_\Omega^{\mathcal{AO}}$, and the mapping γ for the type assignment systems $\lambda\cap_\Omega^{\mathcal{P}a}$, $\lambda\cap_\Omega^{\mathcal{HR}}$.

Each one of the above mappings associates a set of approximants to each λ-term in the standard way.

Definition 3.1.3 *Let* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{DHM}, \mathcal{HR}\}$*. The set* $\mathcal{A}_\nabla(M)$ *of* ∇-approximants *of* M *is defined by*

$$\mathcal{A}_\nabla(M) = \{P \mid \exists M'.\ M \twoheadrightarrow_\beta M' \text{ and } P \equiv \dagger(M')\},$$

where $\dagger = \alpha$ *for* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{DHM}\}$*,* $\dagger = \beta$ *for* $\nabla = \mathcal{AO}$*, and* $\dagger = \gamma$ *for* $\nabla \in \{\mathcal{P}a, \mathcal{HR}\}$*.*

We extend the typing to $\lambda\perp$-terms and to $\lambda\Phi$-terms by adding two different axioms for Φ and nothing for $\perp$.

Definition 3.1.4 *i) We extend the type assignment* $\lambda\cap^\nabla$*, where* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{DHM}\}$*, to* $\lambda\perp$*-terms.*

ii) We extend the type assignment $\lambda\cap_\Omega^{\mathcal{P}a}$ *to* $\lambda\Phi$*-terms by adding the axiom*

$$(Ax\text{-}\Phi\text{-}\mathcal{P}a) \quad \Gamma \vdash^{\mathcal{P}a} \Phi : \omega.$$

iii) We extend the type assignment $\lambda\cap_\Omega^{\mathcal{HR}}$ *to* $\lambda\Phi$*-terms by adding the axiom*

$$(Ax\text{-}\Phi\text{-}\mathcal{HR}) \quad \Gamma \vdash^{\mathcal{HR}} \Phi : \varphi.$$

It is easy to verify that the appropriate generalization of the Generation Theorem (Theorem 2.1.9) holds also for these extensions of the type assignment systems. Therefore we do not introduce different notations for these extended type assignment systems and we will freely refer to the Generation Theorem when discussing derivations in these extended systems.

We immediately get:

Proposition 3.1.5 *i) Let* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{DHM}\}$*. Then* $\Gamma \vdash^\nabla \perp : A$ *iff* $A \sim_\nabla \Omega$*;*

ii) $\Gamma \vdash_{\Omega}^{\mathcal{P}a} \Phi : A$ *iff* $\omega \leq_{\mathcal{P}a} A$.

iii) $\Gamma \vdash^{\mathcal{HR}} \Phi : A$ *iff* $\varphi \leq_{\mathcal{HR}} A$.

We introduce a partial order on the set of approximate normal forms which reflects the β-reduction on λ-terms in the following sense.

Definition 3.1.6 *$P \sqsubseteq_{\nabla} P'$ iff there are λ-terms M, M' such that:*

- $M \twoheadrightarrow_{\beta} M'$
- $P \equiv \dagger(M)$
- $P' \equiv \dagger(M')$

where $\dagger$ is as in Definition 3.1.3.

As expected, the order on approximate normal forms agrees with the typing relation.

Lemma 3.1.7 *i) If $\Gamma \vdash^{\nabla} P : A$, and $P \sqsubseteq_{\nabla} P'$ then $\Gamma \vdash^{\nabla} P' : A$.*

ii) If $P, P' \in \mathcal{A}_{\nabla}(M)$, $\Gamma \vdash^{\nabla} P : A$, and $\Gamma \vdash^{\nabla} P' : B$ then there is $P'' \in \mathcal{A}_{\nabla}(M)$ such that $\Gamma \vdash^{\nabla} P'' : A \cap B$.

PROOF. (i). It suffices to consider the case $P \equiv \dagger((\lambda x.M)N)$ and $P' \equiv \dagger(M[x := N])$.
For $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{DHM}\}$ the proof is easy using Proposition 3.1.5(i).
For $\nabla \in \{\mathcal{P}a, \mathcal{HR}\}$ notice that $\gamma(M[x := N])$ is equal to $\gamma(M)$ where the occurrences of x have been replaced by $\Phi\gamma(N)$ if they are functional and N is an abstraction, and by $\gamma(N)$ otherwise. More formally if we define the mapping $\overline{\gamma}(M) : \Lambda \to \Lambda\Phi$:

$$\overline{\gamma}(M) = \begin{cases} \Phi\gamma(M) & \text{if } M \equiv \lambda x.M' \\ \gamma(M) & \text{otherwise} \end{cases}$$

and the mapping $\{\ \}_y^x : \Lambda \to \Lambda$:

$$\begin{array}{lcl} \{z\}_y^x & = & z \\ \{MN\}_y^x & = & \begin{cases} y\{N\}_y^x & \text{if } M \equiv x \\ \{M\}_y^x\{N\}_y^x & \text{otherwise} \end{cases} \\ \{\lambda z.M\}_y^x & = & \lambda z.\{M\}_y^x. \end{array}$$

then it easy to check that $\gamma(M[x := N]) \equiv \gamma(\{M\}_y^x)[x := \gamma(N)][y := \overline{\gamma}(N)]$ (provided that $y \notin FV(M)$).

By Theorem 2.1.9(ii) from $\Gamma \vdash^{\nabla} P : A$ we get $\Gamma \vdash^{\nabla} \Phi : B \to C \to A$, $\Gamma \vdash^{\nabla} \lambda x.\gamma(M) : B$, $\Gamma \vdash^{\nabla} \gamma(N) : C$ for some B, C.

For $\nabla = \mathcal{P}a$ we get $\omega \leq_{\mathcal{P}a} B \to C \to A$ from $\Gamma \vdash_{\Omega}^{\mathcal{P}a} \Phi : B \to C \to A$ by Proposition 3.1.5(ii). This implies $B \leq_{\mathcal{P}a} \omega$, $C \leq_{\mathcal{P}a} \omega$, and $\omega \leq_{\mathcal{P}a} A$ since $\omega \sim_{\mathcal{P}a} \omega \to \omega$ and $\Sigma^{\mathcal{P}a}$ is a beta theory by Theorem 2.1.7. We obtain by rule $(\leq_{\mathcal{P}a})$ $\Gamma \vdash_{\Omega}^{\mathcal{P}a} \lambda x.\gamma(M) : \omega$ and $\Gamma \vdash_{\Omega}^{\mathcal{P}a} \gamma(N) : \omega$. We can derive $\Gamma, x{:}\omega \vdash_{\Omega}^{\mathcal{P}a} \gamma(M) : \omega$ (by Theorem 2.1.9(iii)), which implies $\Gamma, x{:}\omega, y{:}\omega \vdash_{\Omega}^{\mathcal{P}a} \gamma(\{M\}_y^x) : \omega$ by rule (*weakening*). Moreover from $\Gamma \vdash_{\Omega}^{\mathcal{P}a} \gamma(N) : \omega$ we get $\Gamma \vdash_{\Omega}^{\mathcal{P}a} \Phi\gamma(N) : \omega$ using rules $(\leq_{\mathcal{P}a})$ and $(\to\mathrm{E})$ since $\omega \sim_{\mathcal{P}a} \omega \to \omega$. So we conclude $\Gamma \vdash_{\Omega}^{\mathcal{P}a} \gamma(\{M\}_y^x)[x := \gamma(N)][y := \overline{\gamma}(N)] : A$ by rules (*cut*) and $(\leq_{\mathcal{P}a})$.

For $\nabla = \mathcal{HR}$ we get $\varphi \leq_{\mathcal{HR}} B \to C \to A$ from $\Gamma \vdash^{\mathcal{HR}} \Phi : B \to C \to A$ by Theorem 3.1.5. This implies either ($B \leq_{\mathcal{HR}} \varphi$ and $\varphi \leq_{\mathcal{HR}} C \to A$) or ($B \leq_{\mathcal{HR}} \omega$ and $\omega \leq_{\mathcal{HR}} C \to A$) since $\varphi \sim_{\mathcal{HR}} (\varphi \to \varphi) \cap (\omega \to \omega)$ and $\Sigma^{\mathcal{HR}}$ is a beta theory by Theorem 2.1.7 (notice that $\varphi \cap \omega \sim_{\mathcal{HR}} \omega$). Similarly in the first case from $\varphi \leq_{\mathcal{HR}} C \to A$ we get either $C \leq_{\mathcal{HR}} \varphi$ and $\varphi \leq_{\mathcal{HR}} A$ or $C \leq_{\mathcal{HR}} \omega$ and $\omega \leq_{\mathcal{HR}} A$. In the second case from $\omega \leq_{\mathcal{HR}} C \to A$ we get $C \leq_{\mathcal{HR}} \varphi$ and $\omega \leq_{\mathcal{HR}} A$ since $\omega \sim_{\mathcal{HR}} \varphi \to \omega$. To sum up using rule $(\leq_{\mathcal{HR}})$ we have the following alternative cases:

- $\Gamma \vdash^{\mathcal{HR}} \lambda x.\gamma(M) : \varphi$, $\Gamma \vdash^{\mathcal{HR}} \gamma(N) : \varphi$, and $\varphi \leq_{\mathcal{HR}} A$;
- $\Gamma \vdash^{\mathcal{HR}} \lambda x.\gamma(M) : \varphi$, $\Gamma \vdash^{\mathcal{HR}} \gamma(N) : \omega$, and $\omega \leq_{\mathcal{HR}} A$;
- $\Gamma \vdash^{\mathcal{HR}} \lambda x.\gamma(M) : \omega$, $\Gamma \vdash^{\mathcal{HR}} \gamma(N) : \varphi$, and $\omega \leq_{\mathcal{HR}} A$.

Therefore we get alternatively:

- $\Gamma, x{:}\varphi \vdash^{\mathcal{HR}} \gamma(M) : \varphi$, and $\Gamma \vdash^{\mathcal{HR}} \Phi\gamma(N) : \varphi$;
- $\Gamma, x{:}\omega \vdash^{\mathcal{HR}} \gamma(M) : \omega$, and $\Gamma \vdash^{\mathcal{HR}} \Phi\gamma(N) : \omega$;
- $\Gamma, x{:}\varphi \vdash^{\mathcal{HR}} \gamma(M) : \omega$, and $\Gamma \vdash^{\mathcal{HR}} \Phi\gamma(N) : \varphi$,

so we can conclude as in previous case.

(ii). By hypotheses there are M_1, M_2 such that $M \twoheadrightarrow_{\beta} M_1$, $M \twoheadrightarrow_{\beta} M_2$ and $P \equiv \gamma(M_1)$, $P' \equiv \gamma(M_2)$. By the Church-Rosser property of $\to_{\beta}$ we can find M_3 such that $M_1 \twoheadrightarrow_{\beta} M_3$ and $M_2 \twoheadrightarrow_{\beta} M_3$. By (i) we can choose $P'' \equiv \gamma(M_3)$. $\square$

3.2 Approximation Theorems

First we prove that if $\Gamma \vdash^{\nabla} P : A$ and $P \in \mathcal{A}_{\nabla}(M)$ then we can build a derivation of $\Gamma \vdash^{\nabla} M : A$.

Theorem 3.2.1 (Approximation Theorem - Part 1) *Let* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{DHM}, \mathcal{HR}\}$.

$$\exists P \in \mathcal{A}_{\nabla}(M).\, \Gamma \vdash^{\nabla} P : A \Rightarrow \Gamma \vdash^{\nabla} M : A.$$

Proof. If $P \in \mathcal{A}_{\nabla}(M)$ then by Definition 3.1.3 there is M' such that $M \twoheadrightarrow_\beta M'$, $P = \dagger(M')$ where $\dagger$ is as in Definition 3.1.3. It suffices to show that $\Gamma \vdash^{\nabla} M' : A$ since this implies $\Gamma \vdash^{\nabla} M : A$ by Corollary 2.2.4.

For $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{DHM}\}$ from $\Gamma \vdash^{\nabla} P : A$ we get $\Gamma \vdash^{\nabla} M' : A$ by Proposition 3.1.5(i) and the definition of the mappings α and β.

For $\nabla \in \{\mathcal{P}a, \mathcal{HR}\}$ we show $\Gamma \vdash^{\nabla} M' : A$ by induction on P.

The case $P \equiv x$ is trivial.

If $P \equiv \lambda x.P'$, then $M' \equiv \lambda x.M''$ where $P' \equiv \gamma(M'')$. By Theorem 2.1.9(ii) from $\Gamma \vdash^{\nabla} P : A$ we get $\Gamma, x{:}B_i \vdash^{\nabla} P' : C_i$ and $\bigcap_{i \in I}(B_i \to C_i) \leq_{\nabla} A$ for some I, B_i, C_i. We get by induction $\Gamma, x{:}B_i \vdash^{\nabla} M'' : C_i$ and so we conclude $\Gamma \vdash^{\nabla} M' : A$ using rules ($\to$I), ($\cap$I) and ($\leq_{\nabla}$).

If $P \equiv P_1 P_2$, then $M' \equiv M_1 M_2$ where $P_1 \equiv \gamma(M_1)$ and $P_2 \equiv \gamma(M_2)$. By Theorem 2.1.9(ii) from $\Gamma \vdash^{\nabla} P : A$ we get $\Gamma \vdash^{\nabla} P_1 : B \to A$, $\Gamma \vdash^{\nabla} P_2 : B$ for some B. By induction this implies $\Gamma \vdash^{\nabla} M_1 : B \to A$ and $\Gamma \vdash^{\nabla} M_2 : B$, so we conclude $\Gamma \vdash^{\nabla} M' : A$ using rule ($\to$E).

If $P \equiv \Phi P_1 P_2$, then $M' \equiv M_1 M_2$ where $P_1 \equiv \gamma(M_1)$ and $P_2 \equiv \gamma(M_2)$. By Theorem 2.1.9(ii) from $\Gamma \vdash^{\nabla} P : A$ we get $\Gamma \vdash^{\nabla} \Phi : B \to C \to A$, $\Gamma \vdash^{\nabla} P_1 : B$, $\Gamma \vdash^{\nabla} P_2 : C$ for some B, C. By induction this implies $\Gamma \vdash^{\nabla} M_1 : B$ and $\Gamma \vdash^{\nabla} M_2 : C$.

For $\nabla = \mathcal{P}a$ as in the proof of Lemma 3.1.7(i) we get $\Gamma \vdash^{\mathcal{P}a}_{\Omega} M_1 : \omega$ and $\Gamma \vdash^{\mathcal{P}a}_{\Omega} M_2 : \omega$. We can conclude $\Gamma \vdash^{\nabla} M' : A$ using rules ($\leq_{\mathcal{P}a}$) and ($\to$E) since $\omega \sim_{\mathcal{P}a} \omega \to \omega$.

For $\nabla = \mathcal{HR}$ as in the proof of Lemma 3.1.7(i) we have the following alternative cases:

- $\Gamma \vdash^{\mathcal{HR}} M_1 : \varphi$, $\Gamma \vdash^{\mathcal{HR}} M_2 : \varphi$, and $\varphi \leq_{\mathcal{HR}} A$;
- $\Gamma \vdash^{\mathcal{HR}} M_1 : \varphi$, $\Gamma \vdash^{\mathcal{HR}} M_2 : \omega$, and $\omega \leq_{\mathcal{HR}} A$;
- $\Gamma \vdash^{\mathcal{HR}} M_1 : \omega$, $\Gamma \vdash^{\mathcal{HR}} M_2 : \varphi$, and $\omega \leq_{\mathcal{HR}} A$.

It is easy to verify that in all cases we can derive $\Gamma \vdash^{\nabla} M' : A$ using rules ($\leq_{\mathcal{HR}}$) and ($\to$E). □

In order to prove the converse of Theorem 3.2.1 we will use a Kripke-like version of stable sets [Mit96].

It is useful to introduce the following definition.

Definition 3.2.2 *Let $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{DHM}, \mathcal{HR}\}$. We write*

$$[A]^{\nabla}_{\Gamma} = \{M \mid \exists P \in \mathcal{A}_{\nabla}(M).\ \Gamma \vdash^{\nabla} P : A\}.$$

By definition we get that $M \in [A]^{\nabla}_{\Gamma}$ and $N \twoheadrightarrow_\beta M$ imply $N \in [A]^{\nabla}_{\Gamma}$. Moreover $\Gamma \Subset \Gamma'$ implies $[A]^{\nabla}_{\Gamma} \subseteq [A]^{\nabla}_{\Gamma'}$ for all types $A \in \mathbf{T}^{\nabla}$.

First we need a technical result.

Lemma 3.2.3 *Let $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{DHM}, \mathcal{HR}\}$. If $\Gamma' = \Gamma, z{:}B$, $z \notin \mathrm{FV}(M)$, and $A \not\sim_{\nabla} \Omega$ for $\nabla = \mathcal{AO}$, then $Mz \in [A]^{\nabla}_{\Gamma'}$ implies $M \in [B \to A]^{\nabla}_{\Gamma}$.*

Proof. Let $P \in \mathcal{A}_{\nabla}(Mz)$ and $\Gamma' \vdash^{\nabla} P : A$. We show by cases on P and M that there is $\hat{P} \in \mathcal{A}_{\nabla}(M)$ such that $\Gamma \vdash^{\nabla} \hat{P} : B \to A$. If $P \in \mathcal{A}_{\nabla}(Mz)$ then we are in one of the following cases:

- $P \equiv \bot$ and $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}\}$;
- $M \twoheadrightarrow_{\beta} \lambda x.M'$ and $P \in \mathcal{A}_{\nabla}(M'[x := z])$;
- $P \equiv P'Q$, $P' \in \mathcal{A}_{\nabla}(M)$ and $Q \in \mathcal{A}_{\nabla}(z)$;
- $P \equiv \Phi P'Q$, $P' \in \mathcal{A}_{\nabla}(M)$, $Q \in \mathcal{A}_{\nabla}(z)$ and $\nabla \in \{\mathcal{P}a, \mathcal{HR}\}$.

The case $P \equiv \bot$ is trivial for $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}\}$ and impossible for $\nabla = \mathcal{AO}$ by Proposition 3.1.5(i).

If $M \twoheadrightarrow_{\beta} \lambda x.M'$ and $P \in \mathcal{A}_{\nabla}(M'[x := z])$ we can choose $\hat{P} \equiv \lambda z.P$.
If $P \equiv P'Q$ where $P' \in \mathcal{A}_{\nabla}(M)$ and $Q \in \mathcal{A}_{\nabla}(z)$, then we can choose $\hat{P} \equiv P'$. In fact by Theorem 2.1.9(ii) from $\Gamma' \vdash^{\nabla} P : A$ we get $\Gamma' \vdash^{\nabla} P' : C \to A$, $\Gamma' \vdash^{\nabla} Q : C$ for some C. By Theorem 3.2.1 $\Gamma' \vdash^{\nabla} Q : C$ implies $\Gamma' \vdash^{\nabla} z{:}C$, so we get $B \leq_{\nabla} C$ by Theorem 2.1.9(i) and we conclude using $(\leq_{\nabla})$ and (*strengthening*) $\Gamma \vdash^{\nabla} P' : B \to A$.

If $P \equiv \Phi P'Q$ where $P' \in \mathcal{A}_{\nabla}(M)$ and $Q \in \mathcal{A}_{\nabla}(z)$, then $\nabla \in \{\mathcal{P}a, \mathcal{HR}\}$, and we can choose $\hat{P} \equiv P'$. In fact by Theorem 2.1.9(ii) from $\Gamma' \vdash^{\nabla} P : A$ we get $\Gamma' \vdash^{\nabla} \Phi : C \to D \to A$, $\Gamma' \vdash^{\nabla} P' : C$, $\Gamma' \vdash^{\nabla} Q : D$ for some C, D. As in the previous case we get $B \leq_{\nabla} D$. For $\nabla = \mathcal{P}a$ using Proposition 3.1.5(ii) as in the proof of Lemma 3.1.7(i), we get $C \leq_{\mathcal{P}a} \omega$, $D \leq_{\mathcal{P}a} \omega$, and $\omega \leq_{\mathcal{P}a} A$. Similarly for $\nabla = \mathcal{HR}$ using Proposition 3.1.5(iii), we get either $C \leq_{\mathcal{HR}} \varphi$, $D \leq_{\mathcal{HR}} \varphi$, and $\varphi \leq_{\mathcal{P}a} A$, or $C \leq_{\mathcal{HR}} \varphi$, $D \leq_{\mathcal{HR}} \omega$, and $\omega \leq_{\mathcal{HR}} A$, or $C \leq_{\mathcal{P}a} \omega$, $D \leq_{\mathcal{HR}} \varphi$, and $\omega \leq_{\mathcal{P}a} A$. In all cases we can conclude $C \leq_{\nabla} D \to A \leq_{\nabla} B \to A$ and therefore by $(\leq_{\nabla})$ and (*strengthening*) $\Gamma \vdash^{\nabla} P' : B \to A$. □

The following definition is crucial, albeit a little involved. It amounts essentially to the definition of the natural set-theoretic semantics of intersection types over a suitable Kripke applicative structure (as defined in [Mit96]), where bases play the role of worlds. In order to keep the treatment elementary we don't develop the full theory of the natural semantics of intersection types in Kripke applicative structures.

The definition below is rather long, since we have different cases for the type ω and for arrow types according to the different type theories under consideration.

Definition 3.2.4 (Kripke type interpretation) *Let* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{DHM}, \mathcal{HR}\}$.

i) $[\![\psi]\!]^{\nabla}_{\Gamma} = [\psi]^{\nabla}_{\Gamma}$ *for* $\psi \in \mathbf{C}_{\infty} \cup \{\Omega, \varphi\}$ *or* $\psi = \omega$ *and* $\nabla = \mathcal{P}a$;

ii) $[\![\omega]\!]^{\nabla}_{\Gamma} = \{M \mid \forall \Gamma' \supseteq \Gamma, \forall \vec{N} \in [\varphi]^{\nabla}_{\Gamma'}.M\vec{N} \in [\omega]^{\nabla}_{\Gamma'}\}$ *for* $\nabla \in \{\mathcal{CDZ}, \mathcal{HR}\}$;

iii) $[\![\omega]\!]^{\mathcal{S}c}_{\Gamma} = \{M \mid \forall \vec{N}.M\vec{N} \in [\omega]^{\mathcal{S}c}_{\Gamma}\}$;

iv) $[\![A \to B]\!]^{\nabla}_{\Gamma} = \{M \mid \forall \Gamma' \supseteq \Gamma, \forall N \in [\![A]\!]^{\nabla}_{\Gamma'}.MN \in [\![B]\!]^{\nabla}_{\Gamma'}\}$, *if* $\nabla \neq \mathcal{AO}$ *or* $B \not\sim_{\mathcal{AO}} \Omega$;

v) $[\![A\to B]\!]^{\mathcal{AO}}_\Gamma = [A\to B]^{\mathcal{AO}}_\Gamma$ *if* $B \sim_{\mathcal{AO}} \Omega$*;*

vi) $[\![A\cap B]\!]^\nabla_\Gamma = [\![A]\!]^\nabla_\Gamma \cap [\![B]\!]^\nabla_\Gamma$.

Notice that, since $M \in [A]^\nabla_\Gamma$ and $N \twoheadrightarrow_\beta M$ imply $N \in [A]^\nabla_\Gamma$, the same property holds for $[\![A]\!]^\nabla_\Gamma$. Moreover $\Gamma \Subset \Gamma'$ implies $[\![A]\!]^\nabla_\Gamma \subseteq [\![A]\!]^\nabla_{\Gamma'}$ for all types $A \in \mathbf{T}^\nabla$.

Lemmas 3.2.5, 3.2.8, Definition 3.2.7 and the final theorem are standard.

Lemma 3.2.5 *Let* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{HR}, \mathcal{DHM}\}$.

i) $x\vec{M} \in [A]^\nabla_\Gamma$ *implies* $x\vec{M} \in [\![A]\!]^\nabla_\Gamma$*;*

ii) $[\![A]\!]^\nabla_\Gamma \subseteq [A]^\nabla_\Gamma$.

Proof. (i) and (ii) can be proved simultaneously by induction on A. We consider only some interesting cases.

(i) Case $A \equiv \omega$ and $\nabla = \mathcal{CDZ}$. Let $\Gamma' \Supset \Gamma$. $[\![\varphi]\!]^{\mathcal{CDZ}}_{\Gamma'} = [\varphi]^{\mathcal{CDZ}}_{\Gamma'}$ (*) by Definition 3.2.4(i).

$$\begin{array}{rcll} x\vec{M} \in [\omega]^{\mathcal{CDZ}}_\Gamma \text{ and } (*) & \Rightarrow & x\vec{M}N \in [\omega]^{\mathcal{CDZ}}_{\Gamma'} & \text{by rules } (\le_{\mathcal{CDZ}}) \text{ and } (\to\text{E}) \\ & & & \text{since } \omega \sim_{\mathcal{CDZ}} \varphi \to \omega \\ & \Rightarrow & x\vec{M} \in [\![\omega]\!]^{\mathcal{CDZ}}_\Gamma & \text{by Definition 3.2.4(ii).} \end{array}$$

Case $A \equiv B \to C$. Let $\Gamma' \Supset \Gamma$ and $\nabla \neq \mathcal{AO}$ or $C \not\sim_{\mathcal{AO}} \Omega$. $[\![B]\!]^\nabla_{\Gamma'} \subseteq [B]^\nabla_{\Gamma'}$ (**) by induction on (ii).

$$\begin{array}{rcll} x\vec{M} \in [A]^\nabla_\Gamma \text{ and } (**) & \Rightarrow & x\vec{M}N \in [C]^\nabla_{\Gamma'} & \text{by rule } (\to\text{E}) \\ & \Rightarrow & x\vec{M}N \in [\![C]\!]^\nabla_{\Gamma'} & \text{by induction on (i)} \\ & \Rightarrow & x\vec{M} \in [\![B \to C]\!]^\nabla_\Gamma & \text{by Definition 3.2.4(iv).} \end{array}$$

(ii) Case $A \equiv B\to C$ and either $\nabla \neq \mathcal{AO}$ or $C \not\sim_{\mathcal{AO}} \Omega$. Let $\Gamma' = \Gamma, z{:}B$ where z is fresh, and suppose $M \in [\![B \to C]\!]^\nabla_\Gamma$; then, since $z \in [\![B]\!]^\nabla_{\{z:B\}}$ by induction on (i), we have

$$\begin{array}{rcll} M \in [\![B \to C]\!]^\nabla_\Gamma \text{ and } z \in [\![B]\!]^\nabla_{\{z:B\}} & \Rightarrow & Mz \in [\![C]\!]^\nabla_{\Gamma'} & \text{by Definition 3.2.4(iv)} \\ & \Rightarrow & Mz \in [C]^\nabla_{\Gamma'} & \text{by induction on (ii)} \\ & \Rightarrow & M \in [B\to C]^\nabla_{\Gamma'} & \text{by Lemma 3.2.3.} \end{array}$$

Case $A \equiv B \cap C$. It follows by induction and Lemma 3.1.7(ii). $\square$

The following lemma essentially states that Kripke type interpretations agree with the corresponding type theories.

Lemma 3.2.6 *i) Let $M \in [A]^{\mathcal{CDZ}}_{\Gamma,z:\omega}$ and $N \in [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma}$ then $M[z := N] \in [A]^{\mathcal{CDZ}}_{\Gamma}$.*

ii) Let $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{DHM}, \mathcal{HR}\}$. Then $A \leq_{\nabla} B$ implies $[\![A]\!]^{\nabla}_{\Gamma} \subseteq [\![B]\!]^{\nabla}_{\Gamma}$ for all $A, B \in \mathbf{T}^{\nabla}$.

Proof. (i) If $M \in [A]^{\mathcal{CDZ}}_{\Gamma,z:\omega}$ then there is $P \in \mathcal{A}_{\mathcal{CDZ}}(M)$ such that $\Gamma, z{:}\omega \vdash^{\mathcal{CDZ}} P : A$. The proof is by induction on P.

The cases $P \equiv \bot$ or $A \sim_{\mathcal{CDZ}} \Omega$ are trivial.

If $P \equiv \lambda x.P'$ then $M \twoheadrightarrow_{\beta} \lambda x.M'$ and $P' \in \mathcal{A}_{\mathcal{CDZ}}(M')$. From $\Gamma, z{:}\omega \vdash^{\mathcal{CDZ}} P : A$ we get $\Gamma, z{:}\omega, x{:}B_i \vdash^{\mathcal{CDZ}} P' : C_i$ and $\bigcap_{i\in I}(B_i{\to}C_i) \leq_{\mathcal{CDZ}} A$ for some I and $B_i, C_i \in \mathbf{T}^{\mathcal{CDZ}}$ by Theorem 2.1.9(ii). By induction there are $P_i \in \mathcal{A}_{\mathcal{CDZ}}(M'[z := N])$ such that $\Gamma, x{:}B_i \vdash^{\mathcal{CDZ}} P_i : C_i$ for all $i \in I$. Then by Lemma 3.1.7(ii) there is $P'' \in \mathcal{A}_{\mathcal{CDZ}}(M'[z := N])$ such that $\Gamma, x{:}B_i \vdash^{\mathcal{CDZ}} P'' : \bigcap_{i\in I} C_i$. We can derive $\Gamma \vdash^{\mathcal{CDZ}} \lambda x.P'' : A$ using $(\to\mathrm{I})$, $(\cap\mathrm{I})$ and $(\leq_{\mathcal{CDZ}})$. Since the λ-abstraction of reduction pairs preserves the reduction relation and the mapping α is closed under the λ-abstraction, we get $\lambda x.P'' \in \mathcal{A}_{\mathcal{CDZ}}(M[z := N])$.

If $P \equiv x\vec{P}$ then $M \twoheadrightarrow_{\beta} x\vec{M}$ and $\vec{P} \in \mathcal{A}_{\mathcal{CDZ}}(\vec{M})$. From $\Gamma, z{:}\omega \vdash^{\mathcal{CDZ}} P : A$ we get $\Gamma, z{:}\omega \vdash^{\mathcal{CDZ}} x : \vec{B} \to A$ and $\Gamma, z{:}\omega \vdash^{\mathcal{CDZ}} \vec{P} : \vec{B}$ by Theorem 2.1.9(ii). By induction there are $\vec{P'} \in \mathcal{A}_{\mathcal{CDZ}}(\vec{M}[z := N])$ such that $\Gamma \vdash^{\mathcal{CDZ}} \vec{P'} : \vec{B}$. If $x \neq z$ we are done since $x\vec{P'} \in \mathcal{A}_{\mathcal{CDZ}}(M[z := N])$ and we can derive $\Gamma \vdash^{\mathcal{CDZ}} x\vec{P'} : A$ using $(\to\mathrm{E})$. Otherwise $\Gamma, z{:}\omega \vdash^{\mathcal{CDZ}} z{:}\vec{B} \to A$ implies $\omega \leq_{\mathcal{CDZ}} \vec{B} \to A$ by Theorem 2.1.9(i). Being $\Sigma^{\mathcal{CDZ}}$ a beta theory by Theorem 2.1.7 from $\omega \sim_{\mathcal{CDZ}} \vec{\varphi} \to \omega$ we obtain $\vec{B} \leq_{\mathcal{CDZ}} \vec{\varphi}$ and $\omega \leq_{\mathcal{CDZ}} A$. So we get $\Gamma \vdash^{\mathcal{CDZ}} \vec{P'} : \vec{\varphi}$, i.e. $\vec{M}[z := N] \in [\varphi]^{\mathcal{CDZ}}_{\Gamma}$. By Definition 3.2.4(ii) $N \in [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma}$ and $\vec{M}[z := N] \in [\omega]^{\mathcal{CDZ}}_{\Gamma}$ imply $M[z := N] \in [\omega]^{\mathcal{CDZ}}_{\Gamma}$. Since $\omega \leq_{\mathcal{CDZ}} A$, by Lemma 3.2.5(ii) we get $M[z := N] \in [A]^{\mathcal{CDZ}}_{\Gamma}$.

(ii) We consider only the most interesting cases.

Proof of $[\![\omega{\to}\varphi]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq [\![\varphi]\!]^{\mathcal{CDZ}}_{\Gamma}$. By Lemma 3.2.5(ii) $[\![\omega{\to}\varphi]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq [\omega{\to}\varphi]^{\mathcal{CDZ}}_{\Gamma}$ and by Definition 3.2.2 $[\omega{\to}\varphi]^{\mathcal{CDZ}}_{\Gamma} = [\varphi]^{\mathcal{CDZ}}_{\Gamma}$ since $\omega{\to}\varphi \sim_{\mathcal{CDZ}} \varphi$. Hence we are done since $[\varphi]^{\mathcal{CDZ}}_{\Gamma} = [\![\varphi]\!]^{\mathcal{CDZ}}_{\Gamma}$ by Definition 3.2.4(i).

Proof of $[\![\varphi]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq [\![\omega{\to}\varphi]\!]^{\mathcal{CDZ}}_{\Gamma}$. By Definition 3.2.4(i) $[\![\varphi]\!]^{\mathcal{CDZ}}_{\Gamma} = [\varphi]^{\mathcal{CDZ}}_{\Gamma}$. If $M \in [\varphi]^{\mathcal{CDZ}}_{\Gamma}$ then there is $P \in \mathcal{A}_{\mathcal{CDZ}}(M)$ such that $\Gamma \vdash^{\mathcal{CDZ}} P : \varphi$. The proof is by cases on P.

The case $P \equiv \bot$ is impossible by Proposition 3.1.5(i).

If $P \equiv \lambda z.P'$ then $M \twoheadrightarrow_{\beta} \lambda z.M'$ and $P' \in \mathcal{A}_{\mathcal{CDZ}}(M')$. From $\Gamma \vdash^{\mathcal{CDZ}} P : \varphi$ we get $\Gamma, z{:}\omega \vdash^{\mathcal{CDZ}} P' : \varphi$ by Theorem 2.1.9(ii). This implies $M' \in [\varphi]^{\mathcal{CDZ}}_{\Gamma,z:\omega}$ and also $M' \in [\varphi]^{\mathcal{CDZ}}_{\Gamma',z:\omega}$ for an arbitrary $\Gamma' \sqsupseteq \Gamma$, so we get by (i) $M'[z := N] \in [\varphi]^{\mathcal{CDZ}}_{\Gamma'}$ for an arbitrary $N \in [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma'}$. We conclude observing that $MN \twoheadrightarrow_{\beta} M'[z := N]$.

If $P \equiv x\vec{P}$ notice that $\Gamma \vdash^{\mathcal{CDZ}} P : \varphi$ implies $\Gamma \vdash^{\mathcal{CDZ}} P : \omega{\rightarrow}\varphi$ since $\varphi \sim_{\mathcal{CDZ}} \omega{\rightarrow}\varphi$. Take an arbitrary $\Gamma' \Supset \Gamma$ and an arbitrary $N \in [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma'}$. By Lemma 3.2.5(ii) $[\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma'} \subseteq [\omega]^{\mathcal{CDZ}}_{\Gamma'}$, then there is $P' \in \mathcal{A}_{\mathcal{CDZ}}(N)$ such that $\Gamma' \vdash^{\mathcal{CDZ}} P' : \omega$. We can derive $\Gamma' \vdash^{\mathcal{CDZ}} PP' : \varphi$ from $\Gamma \vdash^{\mathcal{CDZ}} P : \omega{\rightarrow}\varphi$ and $\Gamma' \vdash^{\mathcal{CDZ}} P' : \omega$ using ($\rightarrow$E) and (*weakening*). We conclude by observing that $PP' \in \mathcal{A}_{\mathcal{CDZ}}(MN)$.

Proof of $[\![\varphi{\rightarrow}\omega]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma}$. By Lemma 3.2.5(ii) $[\![\varphi{\rightarrow}\omega]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq [\varphi{\rightarrow}\omega]^{\mathcal{CDZ}}_{\Gamma}$ and by Definition 3.2.2 $[\varphi{\rightarrow}\omega]^{\mathcal{CDZ}}_{\Gamma} = [\omega]^{\mathcal{CDZ}}_{\Gamma}$ since $\varphi{\rightarrow}\omega \sim_{\mathcal{CDZ}} \omega$. Moreover

$$\begin{aligned}
[\![\varphi{\rightarrow}\omega]\!]^{\mathcal{CDZ}}_{\Gamma} &= \{M \mid \forall\Gamma' \Supset \Gamma, \forall N \in [\![\varphi]\!]^{\mathcal{CDZ}}_{\Gamma'}.MN \in [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma'}\} && \text{by Def. 3.2.4(iv)}\\
&= \{M \mid \forall\Gamma' \Supset \Gamma, \forall N \in [\varphi]^{\mathcal{CDZ}}_{\Gamma'}.MN \in [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma'}\} && \text{by Def. 3.2.4(i)}\\
&\subseteq \{M \mid \forall\Gamma' \Supset \Gamma, \forall N, \vec{N} \in [\varphi]^{\mathcal{CDZ}}_{\Gamma'}.MN\vec{N} \in [\omega]^{\mathcal{CDZ}}_{\Gamma'}\} && \text{by Def. 3.2.4(ii).}
\end{aligned}$$

From $[\![\varphi{\rightarrow}\omega]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq [\omega]^{\mathcal{CDZ}}_{\Gamma}$ and $[\![\varphi{\rightarrow}\omega]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq \{M \mid \forall\Gamma' \Supset \Gamma, \forall N, \vec{N} \in [\varphi]^{\mathcal{CDZ}}_{\Gamma'}.MN\vec{N} \in [\omega]^{\mathcal{CDZ}}_{\Gamma'}\}$ we can conclude $[\![\varphi{\rightarrow}\omega]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq \{M \mid \forall\Gamma' \Supset \Gamma, \vec{N} \in [\varphi]^{\mathcal{CDZ}}_{\Gamma'}.M\vec{N} \in [\omega]^{\mathcal{CDZ}}_{\Gamma'}\} = [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma}$.

Proof of $[\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma} \subseteq [\![\varphi{\rightarrow}\omega]\!]^{\mathcal{CDZ}}_{\Gamma}$.

$$\begin{aligned}
M \in [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma} &\Rightarrow \forall\Gamma' \Supset \Gamma, \forall N, \vec{N} \in [\varphi]^{\mathcal{CDZ}}_{\Gamma'}.MN\vec{N} \in [\omega]^{\mathcal{CDZ}}_{\Gamma'} && \text{by Def. 3.2.4(ii)}\\
&\Rightarrow \forall\Gamma' \Supset \Gamma, \forall N \in [\varphi]^{\mathcal{CDZ}}_{\Gamma'}.MN \in [\![\omega]\!]^{\mathcal{CDZ}}_{\Gamma'} && \text{by Def. 3.2.4(i), (ii)}\\
&\Rightarrow M \in [\![\varphi{\rightarrow}\omega]\!]^{\mathcal{CDZ}}_{\Gamma} && \text{by Def. 3.2.4(iv).}
\end{aligned}$$

Proof of $[\![(\varphi{\rightarrow}\varphi) \cap (\omega{\rightarrow}\omega)]\!]^{\mathcal{HR}}_{\Gamma} \subseteq [\![\varphi]\!]^{\mathcal{HR}}_{\Gamma}$. By Lemma 3.2.5(ii) $[\![(\varphi{\rightarrow}\varphi) \cap (\omega{\rightarrow}\omega)]\!]^{\mathcal{HR}}_{\Gamma} \subseteq [(\varphi{\rightarrow}\varphi) \cap (\omega{\rightarrow}\omega)]^{\mathcal{HR}}_{\Gamma}$ and by Definition 3.2.2 $[(\varphi{\rightarrow}\varphi) \cap (\omega{\rightarrow}\omega)]^{\mathcal{HR}}_{\Gamma} = [\varphi]^{\mathcal{HR}}_{\Gamma}$ since $(\varphi{\rightarrow}\varphi) \cap (\omega{\rightarrow}\omega) \sim_{\mathcal{HR}} \varphi$. Hence we are done since $[\varphi]^{\mathcal{HR}}_{\Gamma} = [\![\varphi]\!]^{\mathcal{HR}}_{\Gamma}$ by Definition 3.2.4(i).

Proof of $[\![\varphi]\!]^{\mathcal{HR}}_{\Gamma} \subseteq [\![(\varphi{\rightarrow}\varphi) \cap (\omega{\rightarrow}\omega)]\!]^{\mathcal{HR}}_{\Gamma}$. Let $\Gamma' \Supset \Gamma$.

$$\begin{aligned}
M \in [\![\varphi]\!]^{\mathcal{HR}}_{\Gamma} &\Rightarrow M \in [\varphi]^{\mathcal{HR}}_{\Gamma} && \text{by Definition 3.2.4(i)}\\
&\Rightarrow \exists P \in \mathcal{A}_{\mathcal{HR}}(M)\ \Gamma \vdash^{\mathcal{HR}} P : \varphi\ (*) && \text{by Definition 3.2.2}
\end{aligned}$$

$$\begin{aligned}
N \in [\![\varphi]\!]^{\mathcal{HR}}_{\Gamma'} &\Rightarrow N \in [\varphi]^{\mathcal{HR}}_{\Gamma'} && \text{by Definition 3.2.4(i)}\\
&\Rightarrow \exists P' \in \mathcal{A}_{\mathcal{HR}}(N)\ \Gamma' \vdash^{\mathcal{HR}} P' : \varphi\ ({*}{*}) && \text{by Definition 3.2.2}
\end{aligned}$$

Let $\hat{P} \equiv \Phi P P'$ if P is a λ-abstraction and $\hat{P} \equiv P P'$ otherwise.

$$\begin{array}{llll}
(*) \text{ and } (**) & \Rightarrow & \Gamma' \vdash_{\Omega}^{\mathcal{P}a} \hat{P} : \varphi & \text{by } (Ax\text{-}\Phi\text{-}\mathcal{HR}),\ (\leq_{\mathcal{HR}}),\ (\rightarrow\mathrm{E}) \\
& \Rightarrow & MN \in [\varphi]_{\Gamma'}^{\mathcal{HR}} & \text{since } \hat{P} \in \mathcal{A}_{\mathcal{HR}}(MN) \\
& \Rightarrow & MN \in [\![\varphi]\!]_{\Gamma'}^{\mathcal{HR}} & \text{by Definition 3.2.4(i)} \\
& \Rightarrow & M \in [\![\varphi \rightarrow \varphi]\!]_{\Gamma}^{\mathcal{HR}} & \text{by Definition 3.2.4(iv).}
\end{array}$$

$$\begin{array}{llll}
N \in [\![\omega]\!]_{\Gamma'}^{\mathcal{HR}} & \Rightarrow & N \in [\omega]_{\Gamma'}^{\mathcal{HR}} & \text{by Definition 3.2.4(ii)} \\
& \Rightarrow & \exists P' \in \mathcal{A}_{\mathcal{HR}}(N)\ \Gamma' \vdash^{\mathcal{HR}} P' : \omega\ (***) & \text{by Definition 3.2.2}
\end{array}$$

$$\begin{array}{llll}
\vec{N} \in [\![\varphi]\!]_{\Gamma'}^{\mathcal{HR}} & \Rightarrow & \vec{N} \in [\varphi]_{\Gamma'}^{\mathcal{HR}} & \text{by Definition 3.2.4(ii)} \\
& \Rightarrow & \exists \vec{P} \in \mathcal{A}_{\mathcal{HR}}(\vec{N})\ \Gamma' \vdash^{\mathcal{HR}} \vec{P'} : \vec{\varphi}\ (****) & \text{by Definition 3.2.2}
\end{array}$$

Let $\hat{P} \equiv \Phi P P' \vec{P}$ if P is a λ-abstraction and $\hat{P} \equiv P P' \vec{P}$ otherwise.

$$\begin{array}{llll}
(*), (***) \text{ and } (****) & \Rightarrow & \Gamma' \vdash_{\Omega}^{\mathcal{P}a} \hat{P} : \omega & \text{by } (Ax\text{-}\Phi\text{-}\mathcal{HR}),\ (\leq_{\mathcal{HR}}),\ (\rightarrow\mathrm{E}) \\
& \Rightarrow & MN\vec{N} \in [\omega]_{\Gamma'}^{\mathcal{HR}} & \text{since } \hat{P} \in \mathcal{A}_{\mathcal{HR}}(MN\vec{N}) \\
& \Rightarrow & MN \in [\![\omega]\!]_{\Gamma'}^{\mathcal{HR}} & \text{by Definition 3.2.4(ii)} \\
& \Rightarrow & M \in [\![\omega \rightarrow \omega]\!]_{\Gamma}^{\mathcal{HR}} & \text{by Definition 3.2.4(iv).}
\end{array}$$

Proof of $[\![\omega{\rightarrow}\omega]\!]_{\Gamma}^{\mathcal{P}a} \subseteq [\![\omega]\!]_{\Gamma}^{\mathcal{P}a}$. Let $M \in [\![\omega{\rightarrow}\omega]\!]_{\Gamma}^{\mathcal{P}a}$ and $\Gamma' = \Gamma, z : \omega$, where $z \notin \mathrm{FV}(M)$.

$$\begin{array}{llll}
z \in [\omega]_{\{z:\omega\}}^{\mathcal{P}a} & \Rightarrow & z \in [\![\omega]\!]_{\{z:\omega\}}^{\mathcal{P}a} & \text{by Definition 3.2.4(i)} \\
& \Rightarrow & Mz \in [\![\omega]\!]_{\Gamma'}^{\mathcal{P}a} & \text{by Definition 3.2.4(iv)} \\
& \Rightarrow & Mz \in [\omega]_{\Gamma'}^{\mathcal{P}a} & \text{by Definition 3.2.4(i)} \\
& \Rightarrow & M \in [\omega]_{\Gamma}^{\mathcal{P}a} & \text{by Lemma 3.2.3} \\
& & & \text{and rule } (\leq_{\mathcal{P}a}) \text{ being } \omega \sim_{\mathcal{P}a} \omega \rightarrow \omega \\
& \Rightarrow & M \in [\![\omega]\!]_{\Gamma}^{\mathcal{P}a} & \text{by Definition 3.2.4(i).}
\end{array}$$

Proof of $[\![\omega]\!]_\Gamma^{\mathcal{P}a} \subseteq [\![\omega\to\omega]\!]_\Gamma^{\mathcal{P}a}$. Let $\Gamma' \supseteq \Gamma$.

$$\begin{array}{lll} M \in [\![\omega]\!]_\Gamma^{\mathcal{P}a} & \Rightarrow \quad M \in [\omega]_\Gamma^{\mathcal{P}a} & \text{by Definition 3.2.4(i)} \\ & \Rightarrow \quad \exists P \in \mathcal{A}_{\mathcal{P}a}(M)\ \Gamma \vdash_\Omega^{\mathcal{P}a} P : \omega\ (*) & \text{by Definition 3.2.2} \end{array}$$

$$\begin{array}{lll} N \in [\![\omega]\!]_{\Gamma'}^{\mathcal{P}a} & \Rightarrow \quad N \in [\omega]_{\Gamma'}^{\mathcal{P}a} & \text{by Definition 3.2.4(i)} \\ & \Rightarrow \quad \exists P' \in \mathcal{A}_{\mathcal{P}a}(N)\ \Gamma' \vdash_\Omega^{\mathcal{P}a} P' : \omega\ (**) & \text{by Definition 3.2.2} \end{array}$$

Let $\hat{P} \equiv \Phi P P'$ if P is a λ-abstraction and $\hat{P} \equiv PP'$ otherwise.

$$\begin{array}{llll} (*) \text{ and } (**) & \Rightarrow & \Gamma' \vdash_\Omega^{\mathcal{P}a} \hat{P} : \omega & \text{by } (Ax\text{-}\Phi\text{-}\mathcal{P}a),\ (\leq_{\mathcal{P}a}),\ (\to\text{E}) \\ & \Rightarrow & MN \in [\omega]_{\Gamma'}^{\mathcal{P}a} & \text{since } \hat{P} \in \mathcal{A}_{\mathcal{P}a}(MN) \\ & \Rightarrow & MN \in [\![\omega]\!]_{\Gamma'}^{\mathcal{P}a} & \text{by Definition 3.2.4(i)} \\ & \Rightarrow & M \in [\![\omega \to \omega]\!]_\Gamma^{\mathcal{P}a} & \text{by Definition 3.2.4(iv).} \end{array}$$

□

Definition 3.2.7 (Semantic Satisfiability) *Let ρ be a mapping from term variables to terms and $[\![M]\!]_\rho = M[\vec{x} := \rho(\vec{x})]$. Then we write:*

i) $\nabla, \rho, \Gamma \models M : A$ *iff* $[\![M]\!]_\rho \in [\![A]\!]_\Gamma^\nabla$*;*

ii) $\nabla, \rho, \Gamma' \models \Gamma$ *iff* $\nabla, \rho, \Gamma' \models x : B$ *for all* $x{:}B \in \Gamma$*;*

iii) $\Gamma \models^\nabla M : A$ *iff* $\nabla, \rho, \Gamma' \models \Gamma$ *implies* $\nabla, \rho, \Gamma' \models M : A$ *for all mappings* ρ *and all* ∇*-basis* Γ'.

In line with the previous remarks, the following result can be construed also as the soundness of the natural semantics of intersection types over a particular Kripke applicative structure, where bases play the role of worlds.

Lemma 3.2.8 (Soundness) *Let* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{HR}\}$ *then* $\Gamma \vdash^\nabla M : A$ *implies* $\Gamma \models^\nabla M : A$.

Proof. The proof is by induction on the derivation of $\Gamma \vdash^\nabla M : A$. Cases (Ax), and (Ax-$\Omega$) are immediate. Cases ($\to$E) and ($\cap$I) follow by induction. Case ($\leq_\nabla$) is ok by Lemma 3.2.6(ii). In the case ($\to$I), suppose that $M \equiv \lambda y.R$, $A \equiv B\to C$ and $\Gamma, y : B \vdash^\nabla R : C$ has been derived.

Let $\nabla \neq \mathcal{AO}$ or $C \not\sim_{\mathcal{AO}} \Omega$. Now, if $[\![T]\!]_\rho \in [\![B]\!]_{\Gamma'}^\nabla$ where $\Gamma' \supseteq \Gamma$, from the induction hypothesis

$$[\![R]\!]_{\rho[y:=[\![T]\!]_\rho]} \in [\![C]\!]_{\Gamma'}^\nabla.$$

By the invariance of $[\![\ ]\!]^{\nabla}$ under β-expansion we get

$$[\![(\lambda y.R)T]\!]_{\rho} = [\![\lambda y.R]\!]_{\rho}[\![T]\!]_{\rho} \in [\![C]\!]^{\nabla}_{\Gamma'},$$

hence by Definition 3.2.4(iv)

$$[\![\lambda y.R]\!]_{\rho} \in [\![B \to C]\!]^{\nabla}_{\Gamma},$$

the term T being arbitrary.

The case $\nabla = \mathcal{AO}$ and $C \sim_{\mathcal{AO}} \Omega$ follows easily observing that $\lambda y.P \in \mathcal{A}_{\mathcal{AO}}([\![\lambda y.R]\!]_{\rho})$ for all $P \in \mathcal{A}_{\mathcal{AO}}([\![R]\!]_{\rho})$ and all ρ such that $\rho(y) = y$. We can derive $\vdash^{\mathcal{AO}}_{\Omega} \lambda y.P : B \to C$ using (Ax-Ω), ($\to$I) and ($\leq_{\mathcal{AO}}$). Hence we conclude $[\![M]\!]_{\rho} \in [A]^{\nabla}_{\Gamma}$ which implies $[\![M]\!]_{\rho} \in [\![A]\!]^{\nabla}_{\Gamma}$ by Definition 3.2.4(v). $\square$

Finally we give the crucial result.

Theorem 3.2.9 (Approximation Theorem - Part 2) *Let* $\nabla \in \{\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{DHM}, \mathcal{HR}\}$.

$$\Gamma \vdash^{\nabla} M : A \Rightarrow \exists P \in \mathcal{A}_{\nabla}(M).\ \Gamma \vdash^{\nabla} P : A.$$

Proof. Let $\rho_0(x) = x$. By Lemma 3.2.5(i) $\nabla, \rho_0, \Gamma \models \Gamma$. Then $\Gamma \vdash^{\nabla} M : A$ implies $M = [\![M]\!]_{\rho_0} \in [\![A]\!]^{\nabla}_{\Gamma}$ by Lemma 3.2.8. So we conclude $M \in [A]^{\nabla}_{\Gamma}$ by Lemma 3.2.5(ii). $\square$

Theorems 3.2.1 and 3.2.9 were first proven for $\nabla = \mathcal{BCD}$ in [BCDC83], for $\nabla = \mathcal{S}c$ in [RDR98], for $\nabla = \mathcal{CDZ}$ in [CDCZ87], for $\nabla = \mathcal{AO}$ in [AO93], for $\nabla = \mathcal{P}a$, and for $\nabla = \mathcal{HR}$ in [HRDR92].

3.3 Some applications of the Approximation Theorems

The Approximation Theorems, as stated in the present chapter, can be fruitfully used to study the λ-theory of a given λ-model once we describe such a λ-model as a *filter λ-model*. Filter models can be introduced concisely by the following definition (see [Bar] for more details). Not all filter models are lambda models in the sense of [HL80]: the set of lambda theories which give rise to lambda models is characterised in [Bar].

Definition 3.3.1 *i) A ∇-filter is a set $X \subseteq \mathbf{T}^{\nabla}$ such that:*

(a) $\Omega \in X$;

(b) if $A \leq_{\nabla} B$ and $A \in X$, then $B \in X$;

(c) if $A, B \in X$, then $A \cap B \in X$.

ii) $\mathcal{F}^{\nabla}$ *denotes the set of* ∇*-filters over* $\mathbf{T}^{\nabla}$.

iii) Application $\cdot : \mathcal{F}^{\nabla} \times \mathcal{F}^{\nabla} \to \mathcal{F}^{\nabla}$ *is defined as*

$$X \cdot Y = \{B \mid \exists A \in Y. A \to B \in X\}.$$

iv) Env_{∇} *is the set of all mappings from the set of term variables to* $\mathcal{F}^{\nabla}$.

v) The interpretation function: $[\![\ ]\!]^{\nabla} : \Lambda \times \mathsf{Env}_{\nabla} \to \mathcal{F}^{\nabla}$ *is defined by*

$$[\![M]\!]^{\nabla}_{\rho} = \{A \in \mathbf{T}^{\nabla} \mid \exists \Gamma \models \rho.\ \Gamma \vdash^{\nabla} M : A\},$$

where ρ *ranges over* Env_{∇} *and* $\Gamma \models \rho$ *iff* $(x{:}B) \in \Gamma$ *implies* $B \in \rho(x)$.

vi) The triple $\langle \mathcal{F}^{\nabla}, \cdot, [\![\ ]\!]^{\nabla} \rangle$ *is called the* (filter) model ∇.

Most of the applications of Approximation Theorems to the study of the fine structure of a filter λ-model follow a similar pattern. One usually starts out by focusing on a semidecidable property of lambda terms, which is decidable on approximate normal forms, *e.g.* having a head normal form, being reducible to an n-fold abstraction, being reducible to a closed term, ... Then, by induction on the structure of approximate normal forms, repeatedly using the Generation Theorem (Theorem 2.1.9), one proves a completeness result, namely that an approximate normal form satisfies such a property if and only if it has a type of a given shape, possibly in a given basis.

Since the model is a filter λ-model, this completeness result has a semantical counterpart at the level of compact elements. Finally, being the property under consideration continuous, and so true for an arbitrary lambda term whenever it is true for an approximant of it, by the Approximation Theorem, the completeness result can be used to characterise the interpretation of an arbitrary lambda term.

By way of example in the present section we will discuss some interesting properties of the models $\mathcal{BCD}, \mathcal{S}c, \mathcal{CDZ}, \mathcal{AO}, \mathcal{P}a, \mathcal{DHM}, \mathcal{HR}$. More precisely we will show that:

- the models $\mathcal{BCD}, \mathcal{CDZ}, \mathcal{S}c, \mathcal{DHM}$ are sensible (according to [Bar84, Definition 4.1.7]);
- the top element in the model $\mathcal{AO}$ is the interpretation of the terms of order ∞;
- the model $\mathcal{P}a$ characterises the terms reducible to closed terms;
- the model $\mathcal{HR}$ characterises the terms reducible to $\lambda\mathsf{I}$-terms.

The rest of this section is devoted to the proof of these properties. Other uses of the Approximation Theorem can be found in the corresponding relevant papers, i.e. [BCDC83, CDCZ87, RDR98, AO93, HRDR92, Bar].

Theorem 3.3.2 *The models* $\mathcal{BCD}, \mathcal{CDZ}, \mathcal{S}c, \mathcal{DHM}$ *are sensible, i.e. they equate all unsolvable terms. Moreover the bottom element of the model is the interpretation of all unsolvable terms.*

Proof. It follows immediately from the Approximation Theorem and the fact that $\perp$ is the only approximant of an unsolvable term for the mapping α. Notice that by Proposition 3.1.5(i) the interpretation of unsolvable terms is the set of types equivalent to Ω, i.e. the bottom element of the model. □

Let us recall the definition of term of order ∞ [Lon83].

Definition 3.3.3 *A term M is of order ∞ iff for all integers n there is M' such that*

$$M \twoheadrightarrow_\beta \lambda x_1 \dots \lambda x_n.M'.$$

Theorem 3.3.4 *A term M is of order ∞ iff $[\![M]\!]^{\mathcal{AO}}_\rho = \mathbf{T}^{\mathcal{AO}}$ for all $\rho \in \mathsf{Env}_{\mathcal{AO}}$.*

Proof. It is easy to check by structural induction on types that

$$\text{for all } A \in \mathbf{T}^{\mathcal{AO}} \text{ there is } n \text{ such that } \Omega^n \to \Omega \leq_{\mathcal{AO}} A.$$

So by the Approximation Theorem it suffices to show that if $P \in \Lambda\perp$ is an approximate normal form:

$$\vdash^{\mathcal{AO}} P : \Omega^n \to \Omega \text{ iff } P \equiv \lambda x_1 \dots \lambda x_n.P' \text{ for some } P'.$$

If $P \equiv \lambda x_1 \dots \lambda x_n.P'$ we can derive $\vdash^{\mathcal{AO}} P : \Omega^n \to \Omega$ using axiom (Ax-Ω) and rule ($\to$I).

Vice versa if we assume by contradiction that $P \equiv \lambda x_1 \dots \lambda x_m.P'$ for $m < n$ and P' is either $\perp$ or $x\vec{P}$ for some x, $\vec{P}$, then by Theorem 2.1.9(iii) $\vdash^{\mathcal{AO}} P : \Omega^n \to \Omega$ implies $\{x_1{:}\Omega, \dots, x_m{:}\Omega\} \vdash^{\mathcal{AO}} P' : \Omega^{n-m} \to \Omega$. But this latter judgment cannot be derived by Proposition 3.1.5(i) if $P' \equiv \perp$ and by Theorem 2.1.9(ii) and (i) if $P' \equiv x\vec{P}$. □

Theorem 3.3.5 *A term M reduces to a closed term iff $[\![M]\!]^{\mathcal{P}a}_\rho \supseteq\uparrow \omega$ for all $\rho \in \mathsf{Env}_{\mathcal{P}a}$.*

Proof. By the Approximation Theorem it suffices to check that if $P \in \Lambda\Phi$:

$$\{x{:}\omega \mid x \in Var\} \vdash^{\mathcal{P}a} P : \omega \;\text{ if and only if }\; FV(P) \subseteq Var.$$

The proof is by structural induction on P. Let $\Gamma_\omega = \{x{:}\omega \mid x \in Var\}$. The key property is that $\omega \sim_{\mathcal{P}a} \omega \to \omega$.

The case $P \equiv \lambda y.P'$ follows by Theorem 2.1.9(iii) and using the induction hypothesis.

If $P \equiv y\vec{P}$ then by Theorem 2.1.9(ii) and (i) we need $y \in Var$. Since $\mathcal{P}a$ is a beta theory we get as the proof of Lemma 3.1.7(i) $\Gamma_\omega \vdash^{\mathcal{P}a} \vec{P} : \vec{\omega}$, so we conclude using the induction hypothesis.

Similarly if $P \equiv \Phi P' \vec{P}$ then by Theorem 2.1.9(ii) and Proposition 3.1.5(ii) we have $\Gamma_\omega \vdash^{\mathcal{P}a} P' : \omega$ and $\Gamma_\omega \vdash^{\mathcal{P}a} \vec{P} : \vec{\omega}$. $\square$

Lastly we work out the characterisation of terms reducible to λI-terms.

We define the set of terms we want to characterise and the set obtained by adding the constant Φ to it.

Definition 3.3.6 *i) The set $\Lambda\mathsf{I}\beta$ of $\lambda\mathsf{I}\beta$-terms is the subset of Λ such that $M \in \Lambda\mathsf{I}\beta$ iff M reduces to a term in which all abstracted variables occur at least once (i.e. M reduces to a λI-term).*

ii) The set $\Lambda\mathsf{I}\beta\Phi$ of $\lambda\mathsf{I}\beta\Phi$-terms is obtained by adding the constant Φ to the formation rules of $\lambda\mathsf{I}\beta$-terms.

In the following key lemma we show that each approximate normal form $P \in \Lambda\mathsf{I}\beta\Phi$ in typable with φ from the $\mathcal{HR}$-basis all whose predicates are φ. To deal properly with the structural induction on approximate normal forms in the case of abstractions we also show that if $x \in P$ then P is typable with ω from the $\mathcal{HR}$-basis containing $x{:}\omega$ and all whose other predicates are φ. This is useful since $\varphi \sim_{\mathcal{HR}} (\varphi{\to}\varphi) \cap (\omega{\to}\omega)$.

Lemma 3.3.7 *Let $P \in \Lambda\Phi$ be an approximate normal form, then:*

i) $P \in \Lambda\mathsf{I}\beta\Phi$ iff $\Gamma_\varphi \vdash^{\mathcal{HR}} P : \varphi$;

ii) if $P \in \Lambda\mathsf{I}\beta\Phi$ and $x \in P$ then $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P : \omega$;

iii) if $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P : \omega$ then $x \in P$,

where $\Gamma_\varphi = \{y{:}\varphi \mid y \in Var\}$, and $\Gamma^x_\varphi = \{x{:}\omega\} \cup \{y{:}\varphi \mid y \in Var,\ y \neq x\}$.

Proof. (i), (ii) and (iii) can be simultaneously proved by induction on P.

The case $P \equiv y$ is trivial.

Let $P \equiv \lambda z.P'$. For (i)($\Rightarrow$) notice that $P \in \Lambda\mathsf{I}\beta\Phi$ implies $P' \in \Lambda\mathsf{I}\beta\Phi$ and $z \in P'$. Then we have by induction $\Gamma_\varphi \vdash^{\mathcal{HR}} P' : \varphi$ and $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P' : \omega$, so by rules ($\to$I), ($\cap$I) and ($\leq_{\mathcal{HR}}$) we get $\Gamma_\varphi \vdash^{\mathcal{HR}} P : \varphi$. For (i)($\Leftarrow$) $\Gamma_\varphi \vdash^{\mathcal{HR}} P : \varphi$ implies $\Gamma_\varphi \vdash^{\mathcal{HR}} P' : \varphi$ and $\Gamma^z_\varphi \vdash^{\mathcal{HR}} P' : \omega$ by Theorem 2.1.9(iii) since $\varphi \sim_{\mathcal{HR}} (\varphi{\to}\varphi) \cap (\omega{\to}\omega)$ and $\Sigma^{\mathcal{HR}}$ is beta. By induction $P' \in \Lambda\mathsf{I}\beta\Phi$ and $z \in P'$ so we get $P \in \Lambda\mathsf{I}\beta\Phi$. For (ii) if $x \in P$ then $x \in P'$, and by induction $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P' : \omega$, so by rules ($\to$I) and ($\leq_{\mathcal{HR}}$) we get $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P : \omega$. For (iii) $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P : \omega$ implies $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P' : \omega$ so we get by induction $x \in P'$, i.e. $x \in P$.

Let $P \equiv \chi P' \vec{P}$ where $\chi \in \{\Phi, z\}$. For (i)($\Rightarrow$) notice that $P \in \Lambda\mathsf{I}\beta\Phi$ implies $P' \in \Lambda\mathsf{I}\beta\Phi$ and $\vec{P} \in \Lambda\mathsf{I}\beta\Phi$. Then we have by induction $\Gamma_\varphi \vdash^{\mathcal{HR}} P' : \varphi$ and $\Gamma_\varphi \vdash^{\mathcal{HR}} P_i : \varphi$ for all i, so by axiom either (Ax) or (Ax-Φ-$\mathcal{HR}$) and rules ($\to$E), ($\leq_{\mathcal{HR}}$) we get $\Gamma_\varphi \vdash^{\mathcal{HR}} P : \varphi$. For (i)($\Leftarrow$) notice that $\Gamma_\varphi \vdash^{\mathcal{HR}} \chi : A$ implies $\varphi \leq_{\mathcal{HR}} A$ by Theorem 2.1.9(i) when $\chi \equiv z$ or by Proposition 3.1.5(iii) when $\chi \equiv \Phi$. So $\Gamma_\varphi \vdash^{\mathcal{HR}} P : \varphi$ implies $\Gamma_\varphi \vdash^{\mathcal{HR}} P' : \varphi$ and $\Gamma_\varphi \vdash^{\mathcal{HR}} P_i : \varphi$ for all i by Theorem 2.1.9(ii) since

$\varphi \sim_{\mathcal{HR}} (\varphi\to\varphi) \cap (\omega\to\omega)$. By induction $P' \in \Lambda\mathsf{I}\beta\Phi$ and $P_i \in \Lambda\mathsf{I}\beta\Phi$ for all i, so we get $P \in \Lambda\mathsf{I}\beta\Phi$. For (ii) if $x \in P$ then either $\chi \equiv x$ or $x \in P'\vec{P}$. In the first case $x{:}\omega \vdash^{\mathcal{HR}} x{:}\omega$ so by rules (*weakening*), ($\to$E), ($\leq_{\mathcal{HR}}$) we get $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P : \omega$. In the second case by induction either $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P' : \omega$ or $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P_i : \omega$ for some i, so by axiom either (Ax) or (Ax-Φ-$\mathcal{HR}$) and rules ($\to$E), ($\leq_{\mathcal{HR}}$) we get $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P : \omega$. For (iii) by an argument similar to that showing (i)($\Leftarrow$) $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P : \omega$ and $\chi \not\equiv x$ imply either $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P' : \omega$ or $\Gamma^x_\varphi \vdash^{\mathcal{HR}} P_i : \omega$ for some i, so we get by induction either $x \in P'$ or $x \in P_i$ for some i, i.e. $x \in P$. □

Now we can prove the final result:

Theorem 3.3.8 *Let $\rho_\varphi(x) = \uparrow \varphi$ for all variables x. A term $M \in \Lambda\mathsf{I}\beta$ iff $[\![M]\!]^{\mathcal{HR}}_{\rho_\varphi} \supseteq \uparrow \varphi$.*

Proof. Easy from Theorems 3.2.1, 3.2.9 and Lemma 3.3.7(i) observing that $M \in \Lambda\mathsf{I}\beta$ iff $\mathcal{A}_{\mathcal{HR}}(M) \subseteq \Lambda\mathsf{I}\beta\Phi$. □

Theorem 3.3.8 was first proved in [HRDR92] by purely semantic means. Those theorems have been proven independently for specific systems.

4

Compositional Characterisations of λ-terms

This Chapter is organised as follows. In Section 4.1 we introduce the various properties of λ-terms on which we shall focus. In Section 4.2 we give the compositional characterisations of such properties and we prove the *soundness* of the characterisations. Completeness is proved in Section 4.3. The auxiliary notion of *polarised normal form*, which is instrumental to the study of persistent normal forms, is discussed in Section 4.4.

4.1 Some distinguished properties of λ-terms

In this section we introduce the distinguished classes of λ-terms which we shall focus on in this chapter.

We shall consider first *termination properties*. In particular we shall discuss the crucial property of being *strongly normalising* and the three properties of having a *β-normal form*, of having a *head normal form*, and of having a *weak head normal form*.

Definition 4.1.1 (Normalization property)

i) M is a normal form, $M \in \mathsf{NF}$, *if M cannot be further reduced;*

ii) M is strongly normalising, $M \in \mathsf{SN}$, *if all reductions starting at M are finite;*

iii) M has a normal form, $M \in \mathsf{N}$, *if M reduces to a normal form;*

iv) M has a head normal form, $M \in \mathsf{HN}$, *if M reduces to a term of the form $\lambda\vec{x}.y\vec{M}$ (where possibly y appears in $\vec{x}$);*

v) M has a weak head normal form, $M \in \mathsf{WN}$, *if M reduces to an abstraction or to a term starting with a free variable.*

For each of the above properties, but SN, in the above definition, we shall consider also the corresponding *persistent* version (see Definition 4.1.2). *Persistently normalising* terms have been introduced in [BDC75].

Definition 4.1.2 (Persistent normalisation property)

i) A term M is persistently normalising*, $M \in \mathsf{PN}$, if $M\vec{N} \in \mathsf{N}$ for all terms $\vec{N}$ in* N.

ii) A term M is a persistently normalising normal form*, $M \in \mathsf{PNF}$, if it is both persistently normalising and it is a normal form.*

iii) A term M is persistently head normalising*, $M \in \mathsf{PHN}$, if $M\vec{N} \in \mathsf{HN}$ for all terms $\vec{N}$.*

iv) A term M is persistently weak normalising*, $M \in \mathsf{PWN}$, if $M\vec{N} \in \mathsf{WN}$ for all terms $\vec{N}$.*

Example 4.1.3 *Let* $\mathbf{I} \equiv \lambda x.x$, $\Delta \equiv \lambda x.xx$, $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$, $\mathbf{K} \equiv \lambda xy.x$.

- $\lambda x.yx \in \mathsf{PNF}$.
- $\lambda x.x\Delta\Delta \in \mathsf{NF}$, *but* $\lambda x.x\Delta\Delta \notin \mathsf{PNF}$, *since* $(\lambda x.x\Delta\Delta)\mathbf{I} \twoheadrightarrow_\beta \Delta\Delta \notin \mathsf{N}$.
- $\mathbf{II} \in \mathsf{SN}$, *but* $\mathbf{II} \notin \mathsf{NF}$ *and* $\mathbf{II} \notin \mathsf{PN}$, *since* $\mathbf{II}\Delta\Delta \twoheadrightarrow_\beta \Delta\Delta \notin \mathsf{N}$.
- $(\lambda x.y)(\Delta\Delta) \in \mathsf{PN}$, *but* $(\lambda x.y)(\Delta\Delta) \notin \mathsf{PNF}$ *and* $(\lambda x.y)(\Delta\Delta) \notin \mathsf{SN}$.
- $\lambda y.(\lambda x.y\Delta)(\Delta\Delta) \in \mathsf{N}$, *but* $\lambda y.(\lambda x.y\Delta)(\Delta\Delta) \notin \mathsf{SN}$ *and* $\lambda y.(\lambda x.y\Delta)(\Delta\Delta) \notin \mathsf{PN}$, *since* $(\lambda y.(\lambda x.y\Delta)(\Delta\Delta))\Delta \twoheadrightarrow_\beta \Delta\Delta \notin \mathsf{N}$.
- $\lambda x.y(\Delta\Delta) \in \mathsf{PHN}$, *but* $\lambda x.y(\Delta\Delta) \notin \mathsf{N}$.
- $\lambda x.x(\Delta\Delta) \in \mathsf{HN}$, *but* $\lambda x.x(\Delta\Delta) \notin \mathsf{N}$ *and* $\lambda x.x(\Delta\Delta) \notin \mathsf{PHN}$, *since* $(\lambda x.x(\Delta\Delta))\Delta \longrightarrow_\beta \Delta(\Delta\Delta) \notin \mathsf{HN}$.
- $\mathbf{YK} \in \mathsf{PWN}$, *but* $\mathbf{YK} \notin \mathsf{HN}$.
- $\lambda x.\Delta\Delta \in \mathsf{WN}$, *but* $\lambda x.\Delta\Delta \notin \mathsf{HN}$ *and* $\lambda x.\Delta\Delta \notin \mathsf{PWN}$, *since* $(\lambda x.\Delta\Delta)M \longrightarrow_\beta \Delta\Delta \notin \mathsf{WN}$.

The following proposition, represented pictorially by Figure 4.1, illustrates mutual implications between the above notions:

Proposition 4.1.4 *The following* strict *inclusions hold:*

$$\begin{gathered}
\mathsf{PNF} \subsetneq \mathsf{NF} \subsetneq \mathsf{SN} \subsetneq \mathsf{N} \\
\mathsf{PNF} \subsetneq \mathsf{PN} \subsetneq \mathsf{N} \subsetneq \mathsf{HN} \\
\mathsf{PN} \subsetneq \mathsf{PHN} \subsetneq \mathsf{HN} \subsetneq \mathsf{WN} \\
\mathsf{PHN} \subsetneq \mathsf{PWN} \subsetneq \mathsf{WN}.
\end{gathered}$$

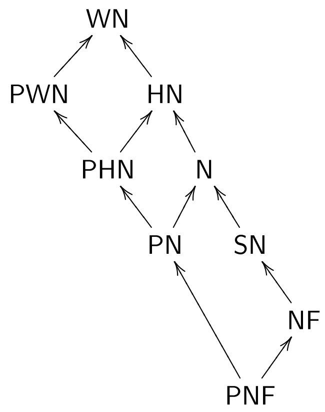

Figure 4.1: Inclusion between sets of λ-terms

No other inclusion holds between the above sets.

The following characterisation of strongly normalising terms will be very useful in the sequel.

Proposition 4.1.5 ([Sev96, HL99]) *The set* SN *is the least set of terms closed under the following rules:*

$$\frac{M_1 \in \mathsf{SN}, \ldots, M_n \in \mathsf{SN}, (n \geq 0)}{xM_1 \ldots M_n \in \mathsf{SN}} \qquad \frac{M \in \mathsf{SN}}{\lambda x.M \in \mathsf{SN}}$$

$$\frac{M[x := N]M_1 \ldots M_n \in \mathsf{SN} \quad N \in \mathsf{SN}, (n \geq 0)}{(\lambda x.M)NM_1 \ldots M_n \in \mathsf{SN}}$$

The proof of the above proposition follows by suitable inductions.

Intersection types can be used to characterise *compositionally* also other evaluation properties of terms, which are not linked to termination. In this chapter we shall consider, by way of example, the property of *reducing to a closed term.* Hence we conclude this section with the definition of:

Definition 4.1.6 (Closable term) *M is* closable*, $M \in \mathsf{C}$, if M reduces to a closed term.*

4.2 Characterising compositionally properties of λ-terms

In this section we put to use *intersection type disciplines* to give a *compositional* characterisation of evaluation properties of λ-terms. In view of Theorem 2.2.4(i) we can only characterise properties which are closed under, at least, $\beta\mathbf{I}$-expansion, hence we will not be able to characterise NF and PNF.

In this section we give the main result of the chapter, Theorem 4.2.2. For each of the properties introduced in Section 4.1, Theorem 4.2.2 provides a *compositional* characterisations in terms of Intersection Type Assignment Systems. Soundness of these characterisations will be proved in the present section (and in Section 4.4) and completeness will be proved in Section 4.3.

Some of the properties characterised in Theorem 4.2.2 had received already characterisations in terms of intersection type disciplines. The most significant case is that of strongly normalising terms. One of the original motivations for introducing intersection types in [Pot80] was precisely that of achieving such a characterisation. Alternative characterisations appear in [Lei86, Bak92, Kri90, Ghi96, AC98, HL99]. In [CDCZ87] both normalising and persistently normalising terms had been characterised using intersection types. The type assignment system in [CDCZ87] has also been discussed in [Ber00]. Closed terms were characterised in [HRDR92]. The characterisations appearing in Theorem 4.2.2 strengthen and generalise all earlier results, since all previous papers consider only specific type theories, and hence in our view Theorem 4.2.2 appears more intrinsic.

Before giving the main theorem a last definition is necessary.

Definition 4.2.1 *i) A type theory Σ^{∇} is an* arrow-type theory *if $\Omega \in \mathbf{C}^{\nabla}$, the axioms of $\mathcal{CDV}$ are admissible in Σ^{∇} and $\forall a \in \mathbf{C}^{\nabla}\ \exists I,\ \{A_i, B_i\}_{i\in I}$ $a \sim_{\nabla} \bigcap_{i\in I}(A_i \to B_i)$.*

ii) A type A contains a type B (notation $B \in A$) if and only if $A \equiv C[B]$ for some context $C[\,]$.

iii) A basis Γ contains a type A (notation $A \in \Gamma$) if and only if there is $x{:}B \in \Gamma$ such that $A \in B$.

iv) A type A contains a type B modulo $\sim_{\nabla}$ (notation $B \in_{\nabla} A$) if and only if there is $A' \sim_{\nabla} A$ such that $B \in A'$.

v) A basis Γ contains a type A modulo $\sim_{\nabla}$ (notation $A \in_{\nabla} \Gamma$) if and only if there is $x{:}B \in \Gamma$ such that $A \in_{\nabla} B$.

The theories $\Sigma^{\mathcal{S}c}$, $\Sigma^{\mathcal{P}a}$, $\Sigma^{\mathcal{CDZ}}$ and $\Sigma^{\mathcal{DHM}}$ of Figure 1.2 are arrow-type theories. For example, $\Omega \notin \omega$ but $\Omega \in_{\mathcal{S}c} \omega$ since $\omega \sim_{\mathcal{S}c} \Omega \to \omega$ and $\Omega \in \Omega \to \omega$.

Finally we can state the main result:

Theorem 4.2.2 (Characterisation)

1 Normalisation properties

i) (strongly normalising terms) A λ-term $M \in$ SN if and only if for all type theories Σ^{∇} there exist $A \in \mathbf{T}^{\nabla}$ and a ∇-basis Γ such that $\Gamma \vdash^{\nabla} M : A$.

Moreover in the system $\lambda\cap_{\mathcal{B}}^{\mathcal{CDV}}$ *the terms satisfying the latter property are precisely the strongly normalising ones.*

ii) *(normalising terms) A* λ*-term* $M \in \mathsf{N}$ *if and only if for all type theories* Σ^{∇} *such that* $\{\Omega\} \subset \mathbf{C}^{\nabla}$, [1] *there exist* $A \in \mathbf{T}^{\nabla}$ *and a* $\bigtriangledown$*-basis* Γ *such that* $\Gamma \vdash_{\Omega}^{\nabla} M : A$ *and* $\Omega \notin A, \Gamma$. *Moreover in the system* $\lambda\cap_{\Omega}^{\mathcal{BCD}}$ *the terms satisfying the latter property are precisely the ones which have a normal form. Furthermore, in the system* $\lambda\cap_{\Omega}^{\mathcal{CDZ}}$ *the terms typable with type* φ *in the* $\mathcal{CDZ}$*-basis all of whose predicates are* ω*, are precisely the ones which have a normal form.*

iii) *(head normalising terms) A* λ*-term* $M \in \mathsf{HN}$ *if and only if for all type theories* Σ^{∇} *such that* $\Omega \in \mathbf{C}^{\nabla}$*, for all* $A \in \mathbf{T}^{\nabla}$ *and there exist a* $\bigtriangledown$*-basis* Γ *and two integers* m, n *such that* $\Gamma \vdash_{\Omega}^{\nabla} M : (\Omega^m \to A)^n \to A$. *Moreover in the system* $\lambda\cap_{\Omega}^{\mathcal{BCD}}$ *the terms satisfying the latter property are precisely the ones which have a head normal form. Furthermore, in the system* $\lambda\cap_{\Omega}^{\mathcal{DHM}}$ *the terms typable with type* φ *in the* $\mathcal{DHM}$*-basis all of whose predicates are* ω*, are precisely the ones which have a head normal form.*

iv) *(weak head normalising terms) A* λ*-term* $M \in \mathsf{WN}$ *if and only if for all type theories* Σ^{∇} *such that* $\Omega \in \mathbf{C}^{\nabla}$*, there exists a* $\bigtriangledown$*-basis* Γ *such that* $\Gamma \vdash_{\Omega}^{\nabla} M : \Omega \to \Omega$. *Moreover in the system* $\lambda\cap_{\Omega}^{\mathcal{AO}}$ *the terms satisfying the latter property are precisely the ones which have a weak head normal form.*

2 Persistent normalisation properties

i) *(persistently normalising terms) A* λ*-term* $M \in \mathsf{PN}$ *if and only if for all arrow-type theories* Σ^{∇} *and all* $A \in \mathbf{T}^{\nabla}$ *with* $\Omega \notin_{\nabla} A$ *there exists a* $\bigtriangledown$*-basis* Γ *such that* $\Omega \notin_{\nabla} \Gamma$ *and* $\Gamma \vdash_{\Omega}^{\nabla} M : A$. *Moreover in the system* $\lambda\cap_{\Omega}^{\mathcal{CDZ}}$ *the terms typable with type* ω *in the* $\mathcal{CDZ}$*-basis all of whose predicates are* ω *are precisely the persistently normalising ones.*

ii) *(persistently head normalising terms) A* λ*-term* $M \in \mathsf{PHN}$ *if and only if for all type theories* Σ^{∇} *such that* $\Omega \in \mathbf{C}^{\nabla}$ *and all* $A \in \mathbf{T}^{\nabla}$ *there exists a* $\bigtriangledown$*-basis* Γ *and an integer* n *such that* $\Gamma \vdash_{\Omega}^{\nabla} M : \Omega^n \to A$. *Moreover in the systems* $\lambda\cap_{\Omega}^{\mathcal{SC}}$ *and* $\lambda\cap_{\Omega}^{\mathcal{DHM}}$ *the terms typable with type* ω *in the basis all of whose predicates are* ω*, are precisely the persistently head normalising ones.*

iii) *(persistently weak normalising terms) A* λ*-term* $M \in \mathsf{PWN}$ *if and only if for all type theories* Σ^{∇} *such that* $\Omega \in \mathbf{C}^{\nabla}$ *and all integers* n *there exists a* $\bigtriangledown$*-basis* Γ *such that* $\Gamma \vdash_{\Omega}^{\nabla} M : \Omega^n \to \Omega$. *Moreover in the system* $\lambda\cap_{\Omega}^{\mathcal{AO}}$ *the terms satisfying the latter property are precisely the persistently weak normalising ones.*

[1] The condition $\{\Omega\} \subset \mathbf{C}^{\nabla}$ says that $\mathbf{C}^{\nabla}$ contains Ω and at least one other constant.

3 Closability *(closed terms) A λ-term $M \in \mathsf{C}$ if and only if for all type theories Σ^∇ such that $\Omega \in \mathbf{C}^\nabla$ and $\omega \sim_\nabla \omega \to \omega$ for some $\omega \in \mathbf{C}^\nabla$, M is typable with type ω, for the empty ∇-basis. Moreover in the system $\lambda\cap_\Omega^{\mathcal{P}a}$ the terms satisfying the latter property are precisely the terms which reduce to closed terms.*

The proofs of the *only if* parts of the Theorem are mainly straightforward inductions and case split, and follow, but the case of persistently normalising terms (2.i), which is proved in Section 4.4. The syntactic characterisation of the persistently normalising normal forms is quite technical. Our proof essentially follows the line of [CDCZ87], but here we completely develop arguments that there were only sketched.

The proofs of the *if* parts require the set-theoretic semantics of intersection types using stable sets [Bar, ADCH00], also called saturated sets by [GLT89], which is developed in Section 4.3.

PROOF. Proof of ($\Rightarrow$).

(1.iv) By Theorem 2.2.4(iv) it suffices to consider M in weak head normal form. If $M \equiv \lambda x.N$ then we get $\vdash_\Omega^\nabla N : \Omega$ by (Ax-Ω) and $\vdash_\Omega^\nabla M : \Omega \to \Omega$ by rule ($\to$I). If $M \equiv x\vec{M}$, where m is the length of $\vec{M}$, we derive $x : \Omega^{m+1} \to \Omega \vdash_\Omega^\nabla M : \Omega \to \Omega$ using (Ax-Ω) and ($\to$E).

(1.iii) Again by Theorem 2.2.4(iv) it suffices to consider M in head normal form. Let $M \equiv \lambda\vec{y}.x\vec{M}$ where $\vec{y}$ has length n and $\vec{M}$ has length m. We have $x{:}\Omega^m{\to}A \vdash_\Omega^\nabla x\vec{M} : A$ using rule ($\to$E). By rule ($\to$I) this implies
$x{:}\Omega^m{\to}A \vdash_\Omega^\nabla M : (\Omega^m{\to}A)^n{\to}A$. For $\lambda\cap_\Omega^{\mathcal{DHM}}$ by choosing $A \equiv \omega$ we get from above $x{:}\Omega^m \to \omega \vdash_\Omega^{\mathcal{DHM}} M : (\Omega^m \to \omega)^n \to \omega$. By rules ($\leq_{\mathcal{DHM}}$) and ($\leq_{\mathcal{DHM}}$ L) this implies $x{:}\omega \vdash_\Omega^{\mathcal{DHM}} M : \varphi$ since $\omega \sim_{\mathcal{DHM}} \Omega \to \omega$, $\omega \leq_{\mathcal{DHM}} \varphi$ and $\varphi \sim_{\mathcal{DHM}} \omega \to \varphi$.

(1.ii) Similarly, it's sufficient to consider M in normal form. The proof is by induction on M. The only interesting case is $M \equiv x\vec{M}$ where $\vec{M} \equiv M_1 \dots M_m$. By induction we have $\Gamma_j \vdash_\Omega^\nabla M_j : A_j$, for some Γ_j, A_j not containing Ω and for $j \leq m$. This implies: $\uplus_{j \leq m}\Gamma_j \uplus \{x{:}A_1{\to}\dots{\to}A_m{\to}A\} \vdash_\Omega^\nabla x\vec{M} : A$, where A is an arbitrary type not containing Ω.
For $\lambda\cap_\Omega^{\mathcal{CDZ}}$ let $\Gamma = \{x{:}\omega \mid x \in \mathrm{FV}(M)\}$. If $M \equiv x\vec{M}$ then by induction we have $\Gamma \vdash_\Omega^{\mathcal{CDZ}} M_j : \varphi$ and this implies $\Gamma \vdash_\Omega^{\mathcal{CDZ}} x\vec{M} : \omega$, since $\omega \sim_{\mathcal{CDZ}} \varphi \to \omega$. By rule ($\leq_{\mathcal{CDZ}}$) we conclude $\Gamma \vdash_\Omega^{\mathcal{CDZ}} M : \varphi$. If $M \equiv \lambda y.N$ then by induction we have $\Gamma, y : \omega \vdash_\Omega^{\mathcal{CDZ}} N : \varphi$ and this implies $\Gamma \vdash_\Omega^{\mathcal{CDZ}} M : \omega \to \varphi$. By rule ($\leq_{\mathcal{CDZ}}$) we conclude $\Gamma \vdash_\Omega^{\mathcal{CDZ}} M : \varphi$.

(1.i) By induction on the structure of strongly normalising terms (see Proposition 4.1.5). The only interesting case is $M \equiv (\lambda x.R)N\vec{M}$ where m is the length of $\vec{M}$ and both $R[x := N]\vec{M}$ and N are strongly normalising. By induction hypothesis there are Γ, A, Γ', B such that $\Gamma \vdash^\nabla R[x := N]\vec{M} : A$ and $\Gamma' \vdash^\nabla N : B$. We get

$\Gamma \uplus \Gamma' \vdash^{\nabla} R[x := N]\vec{M} : A$ and $\Gamma \uplus \Gamma' \vdash^{\nabla} N : B$, so if $m = 0$ it follows from Theorem 2.2.2(i). If $m > 0$ by iterated applications of Generation Theorem 2.1.9(ii) to $\Gamma \vdash^{\nabla} R[x := N]\vec{M} : A$ we have

$$\Gamma \vdash^{\nabla} R[x := N] : B_1^{(i)} \to \ldots \to B_m^{(i)} \to B^{(i)}, \quad \Gamma \vdash^{\nabla} M_j : B_j^{(i)}, \quad (j \leq m)$$

and $\bigcap_{i \in I} B^{(i)} \leq_{\nabla} A$ for some $I, B_j^{(i)}(j \leq m), B^{(i)} \in \mathbf{T}^{\nabla}$. As in case $m = 0$ we obtain $\Gamma \uplus \Gamma' \vdash^{\nabla} (\lambda x.R)N : B_1^{(i)} \to \ldots \to B_m^{(i)} \to B^{(i)}$. So we can conclude $\Gamma \uplus \Gamma' \vdash^{\nabla} (\lambda x.R)N\vec{M} : A$.

(2.iii) If M is persistently weak head normalising then either M is an unsolvable term of order ∞ (as defined in [AO93]), i.e. for all n there is N such that $M =_{\beta} \lambda x_1 \ldots x_n.N$, or M is a solvable term such that the head variable of its head normal form is free. In fact if M is an unsolvable term of a finite order, i.e. $M =_{\beta} \lambda x_1 \ldots x_n.N$ where N is unsolvable and it does not reduce to an abstraction, then $M\vec{N} \notin \mathsf{WN}$ where $\vec{N}$ are n arbitrary λ-terms. If $M =_{\beta} \lambda \vec{x} y \vec{z}.y\vec{N}$ we get $M\vec{x}(\Delta\Delta)\vec{z} \twoheadrightarrow_{\beta} \Delta\Delta\vec{N}' \notin \mathsf{WN}$, where $\vec{N}' = \vec{N}[y := \Delta\Delta]$.
If M is an unsolvable term of order ∞, i.e. for all n, there is N such that $M =_{\beta} \lambda x_1 \ldots x_n.N$, we can derive $\vdash_{\Omega}^{\nabla} \lambda x_1 \ldots x_n.N : \Omega^n \to \Omega$ by (Ax-Ω) and rule ($\to$I). If M is a solvable term such that the head variable of its head normal form is free, i.e. $M =_{\beta} \lambda \vec{x}.y\vec{N}$, we can derive for all l $y : \Omega^{m+l} \to \Omega \vdash_{\Omega}^{\nabla} \lambda \vec{x}.y\vec{N} : \Omega^{n+l} \to \Omega$, where m is the length of $\vec{N}$ and n is the length of $\vec{x}$.

(2.ii) By (2.iii) the head variable of the head normal form of M must be free. We can type a term of the shape $\lambda \vec{x}.y\vec{N}$ where $y \notin \vec{x}$ as follows $y : \Omega^m \to A \vdash_{\Omega}^{\nabla} \lambda \vec{x}.y\vec{N} : \Omega^n \to A$, where m is the length of $\vec{N}$ and n is the length of $\vec{x}$. For $\lambda\cap_{\Omega}^{\mathcal{DHM}}$ by choosing $A \equiv \omega$ we get $y{:}\Omega^m \to \omega \vdash_{\Omega}^{\mathcal{DHM}} M : \Omega^n \to \omega$, so we conclude $y{:}\omega \vdash_{\Omega}^{\mathcal{DHM}} M : \omega$ since $\omega \sim_{\mathcal{DHM}} \Omega \to \omega$.

(3) Let $\Gamma_{\omega} = \{x : \omega \mid x \in \mathsf{V}\}$. It is easy to verify by induction on the definition of λ-terms that using $\omega \sim_{\nabla} \omega \to \omega$ we can derive $\Gamma_{\omega} \vdash_{\Omega}^{\nabla} M : \omega$ for all λ-terms M. By Theorem 2.2.4(iv) and (strengthening) we obtain that $\vdash_{\Omega}^{\nabla} M : \omega$ whenever M reduces to a closed term.

□

Remark 4.2.3 *From the proofs of* (2.iii) *and* (2.ii) *it follows that* $\mathsf{PHN} = \mathsf{PWN} \cap \mathsf{HN}$.

4.3 Set-theoretic semantics using stable sets

This section is devoted to prove the *if* parts of Theorem 4.2.2, by showing that all the given characterisations are complete.

The proof technique which we shall adopt to achieve this is uniform for all properties, and it is based on the *set theoretic semantics of intersection types* [ADCH00]. The set-theoretic semantics of a type, for a given applicative structure, is a subset of the structure itself. *Intersection* is interpreted as *set-theoretic intersection*, $\leq$ is interpreted as *set-theoretic inclusion*, and $A \rightarrow B$ is interpreted *à la logical relation*, i.e. as a subset of the points of the structure whose functional behaviour is that of mapping all points in A into B.

In the present context, there is only one applicative structure under consideration. This is the *term structure* Λ, i.e. the applicative structure whose domain are the λ-terms and where application is just juxtaposition of terms.

In order to ensure that the interpretations of types consist of terms which satisfy appropriate properties, we need to give the set-theoretic semantics using special classes of *stable* sets, for suitable notions of stability. These stability properties amount essentially to suitable invariants for the set-theoretic operators corresponding to the type constructors. This proof technique has been used by various authors, e.g. stable sets [Kri90], admissible relations [Mit96], essentially in connection with strongly normalising terms. Here we develop a full-blown version of this technique, which is applicable to many other evaluation properties.

We will consider two interpretations of the arrow type constructor, the *simple semantics* and the *weak semantics*. To this end we give the following definition:

Definition 4.3.1 *Let $X, Y \subseteq \Lambda$:*

i) $X \Rightarrow Y = \{M \in \Lambda \mid \forall N \in X \; MN \in Y\}$

ii) $X \Rightarrow^{\mathsf{W}} Y = \{M \in \mathsf{WN} \mid \forall N \in X \; MN \in Y\}$.

Now, in accordance to the set-theoretic semantics we put:

Definition 4.3.2 *(Type Interpretation)*

i) The simple interpretation $[\![\;]\!]$ *of types in* $\mathbf{T}^{\nabla}$ *induced by the type environment* $\mathcal{V} : \mathbf{C}^{\nabla} \rightarrow \mathcal{P}(\Lambda)$ *is defined by:*

  (a) $[\![\Omega]\!]_{\mathcal{V}} = \Lambda$ *if* $\Omega \in \mathbf{C}^{\nabla}$*;*

  (b) $[\![A]\!]_{\mathcal{V}} = \mathcal{V}(A)$ *if* $A \in \mathbf{C}^{\nabla}$ *and* $A \not\sim_{\nabla} \Omega$*;*

  (c) $[\![A \rightarrow B]\!]_{\mathcal{V}} = [\![A]\!]_{\mathcal{V}} \Rightarrow [\![B]\!]_{\mathcal{V}}$*;*

  (d) $[\![A \cap B]\!]_{\mathcal{V}} = [\![A]\!]_{\mathcal{V}} \cap [\![B]\!]_{\mathcal{V}}$.

ii) The weak interpretation $[\![\;]\!]^{\mathsf{W}}$ *of types in* $\mathbf{T}^{\nabla}$ *induced by the type environment* $\mathcal{V} : \mathbf{C}^{\nabla} \rightarrow \mathcal{P}(\Lambda)$ *is defined as the simple interpretation but for clause (c), which now is taken to be:*
  (c′) $[\![A \rightarrow B]\!]^{\mathsf{W}}_{\mathcal{V}} = [\![A]\!]^{\mathsf{W}}_{\mathcal{V}} \Rightarrow^{\mathsf{W}} [\![B]\!]^{\mathsf{W}}_{\mathcal{V}}$.

Notice that if $\Omega \in \mathbf{C}^{\nabla}$ then $[\![\Omega]\!]_{\mathcal{V}} = [\![\Omega]\!]^{\mathsf{W}}_{\mathcal{V}} = [\![\Omega \rightarrow \Omega]\!]_{\mathcal{V}} = \Lambda$ and $[\![\Omega \rightarrow \Omega]\!]^{\mathsf{W}}_{\mathcal{V}} = \mathsf{WN}$.

The interest of these semantics lies in the Soundness Theorem 4.3.5, below. But in order to be able to state it we need some further definitions.

Definition 4.3.3 *i) A type environment* $\mathcal{V}$ agrees *with a type theory* Σ^∇ *if and only if*

(a) $\forall N \in [\![A]\!]_\mathcal{V}.\ M[x := N] \in [\![B]\!]_\mathcal{V}$ *implies* $\lambda x.M \in [\![A \to B]\!]_\mathcal{V}$;

(b) *if* $A \leq_\nabla B$ *then* $[\![A]\!]_\mathcal{V} \subseteq [\![B]\!]_\mathcal{V}$.

ii) A type environment $\mathcal{V}$ W-agrees *with a type theory* Σ^∇ *if and only if*

(a) $\forall N \in [\![A]\!]^\mathsf{W}_\mathcal{V}.\ M[x := N] \in [\![B]\!]^\mathsf{W}_\mathcal{V}$ *implies* $\lambda x.M \in [\![A \to B]\!]^\mathsf{W}_\mathcal{V}$;

(b) *if* $A \leq_\nabla B$ *then* $[\![A]\!]^\mathsf{W}_\mathcal{V} \subseteq [\![B]\!]^\mathsf{W}_\mathcal{V}$.

Looking at the weak interpretations of Ω and $\Omega \to \Omega$ it is clear that no environment can W-agree with Σ^∇ whenever $\Omega \sim_\nabla \Omega \to \Omega$.

Definition 4.3.4 *(Semantic Satisfiability) Let* $\rho : \mathsf{V} {\to} \Lambda$.

i) $[\![M]\!]_\rho = M[\vec{x} := \vec{N}]$ *where* $\vec{x} = FV(M)$ *and* $\rho(\vec{x}) = \vec{N}$;

ii) $\rho, \mathcal{V} \models M : A$ *if and only if* $[\![M]\!]_\rho \in [\![A]\!]_\mathcal{V}$;

iii) $\rho, \mathcal{V} \models \Gamma$ *if and only if* $\rho, \mathcal{V} \models x : B$ *for all* $x{:}B \in \Gamma$;

iv) $\Gamma \models^\nabla M : A$ *if and only if* $\rho, \mathcal{V} \models \Gamma$ *implies* $\rho, \mathcal{V} \models M : A$ *for all* $\mathcal{V}$ *which agree with* Σ^∇, *and all* ρ.

v) Similarly

- $\rho, \mathcal{V} \models_\mathsf{W} \Gamma$ *if and only if* $[\![x]\!]_\rho \in [\![B]\!]^\mathsf{W}_\mathcal{V}$ *for all* $x{:}B \in \Gamma$;
- $\Gamma \models^\nabla_\mathsf{W} M : A$ *if and only if* $\rho, \mathcal{V} \models_\mathsf{W} \Gamma$ *implies* $[\![M]\!]_\rho \in [\![A]\!]^\mathsf{W}_\mathcal{V}$ *for all* $\mathcal{V}$ *which* W-*agrees with* Σ^∇ *and all* ρ.

Finally we can give:

Theorem 4.3.5 (Soundness) $\Gamma \vdash^\nabla M : A$ *implies* $\Gamma \models^\nabla M : A$ *and* $\Gamma \models^\nabla_\mathsf{W} M : A$.

Proof. By induction on derivations. The restriction to type environments which agree with Σ^∇ is essential for the soundness of rules ($\to$I) and ($\leq_\nabla$). □

The above theorem is a very powerful tool for proving properties of typable terms, which will be constantly used in the completeness part of the proof of Theorem 4.2.2. Roughly the idea is the following. In order to show that a term, typable in a given type theory (or with a given type, in a given type theory) has a given property, we pick a suitable type environment which agrees with that type theory and show that all terms in the interpretations of all the types (or in the interpretation of the type in question) satisfy that property. Usually variables belong to the interpretations of types, or else we are interested only in closable terms. So, in both cases, by taking the *identity* term environment $\rho_0(x) = x$ one has that $[\![M]\!]_{\rho_0} = M$, and so, if a term is typable, then it satisfies the property in question.

The difficulty, of course, lies in showing that the properties in question are satisfied by the sets in the range of the type environments and that they are preserved by the "intersection" and the "arrow" constructions. As is normal with these inductive proofs, a possibly stronger hypothesis than the one that all terms in the interpretation of the type satisfy the property in question has to be assumed. After [Kri90] we shall refer to these induction hypotheses as *stability* properties.

The stability properties we shall be interested in are the following:

Definition 4.3.6 *i) A set $X \subseteq \mathsf{WN}$ is* WN-type-stable *if it contains $x\vec{M}$ for all $\vec{M} \in \Lambda$, and it is closed under head expansion of redexes;*

ii) A set $X \subseteq \mathsf{HN}$ is HN-type-stable *if it contains $x\vec{M}$ for all $\vec{M} \in \Lambda$ and it is closed under head expansion of redexes;*

iii) A set $X \subseteq \mathsf{N}$ is N-type-stable *if it contains $x\vec{M}$ for all $\vec{M} \in \mathsf{N}$ and it is closed under head expansion of redexes;*

iv) A set $X \subseteq \mathsf{SN}$ is SN-type-stable *if it contains $x\vec{M}$ for all $\vec{M} \in \mathsf{SN}$ and it is closed under head expansion of λ-**I**-redexes or of λ-**K**-redexes[2] whose argument is in SN.*

Notice that none of the stable sets in the above definition can be empty.

The above definitions were given essentially to be able to show the following proposition, namely that the stability properties are preserved under suitable *set-theoretic* constructions. This result will imply, *inter alia*, that all sets in the range of the appropriate type interpretations satisfy the appropriate stability property.

Proposition 4.3.7 *Let $\mathsf{S} \in \{\mathsf{WN}, \mathsf{HN}, \mathsf{N}, \mathsf{SN}\}$, $\mathsf{T} \in \{\mathsf{HN}, \mathsf{N}, \mathsf{SN}\}$, and $X, Y \subseteq \Lambda$.*

i) If Y is closed under head expansion of some kinds of redexes then both $X \Rightarrow^{\mathsf{W}} Y$ and $X \Rightarrow Y$ are closed under head expansion of the same kinds of redexes for all $X \subseteq \Lambda$;

ii) If X, Y are closed under head expansion of some kinds of redexes then $X \cap Y$ is closed under head expansion of the same kinds of redexes;

iii) Each S is S-type-stable;

iv) $\Lambda \Rightarrow^{\mathsf{W}} \Lambda$ is WN-type-stable;

v) If Y is WN-type-stable then $\Lambda \Rightarrow^{\mathsf{W}} Y$ is WN-type-stable;

vi) If X, Y are WN-type-stable then $X \Rightarrow^{\mathsf{W}} Y$ is WN-type-stable;

vii) If Y is HN-type-stable then $\Lambda \Rightarrow Y$ is HN-type-stable;

[2] $(\lambda x.M)N$ is a λ-**K**-redex if and only if $x \notin FV(M)$.

viii) If X, Y are T*-type-stable then $X \Rightarrow Y$ is* T*-type-stable;*

ix) If X, Y are S*-type-stable then $X \cap Y$ is* S*-type-stable;*

x) If X is S*-type-stable then $X \cap \Lambda$ is* S*-type-stable.*

Proof. We show only (iv), (v), (vi), (vii), and (viii), the other points being immediate.

First notice that $X \Rightarrow^{\mathsf{W}} Y \subseteq \mathsf{WN}$ for all $X, Y \subseteq \Lambda$ by definition. Moreover $Mx \in \mathsf{T}$ implies $M \in \mathsf{T}$ for $\mathsf{T} \in \{\mathsf{HN}, \mathsf{N}, \mathsf{SN}\}$, and therefore from $Y \subseteq \mathsf{T}$ and $x \in X$ we get $X \Rightarrow Y \subseteq \mathsf{T}$.

If Y is Λ or it is WN-type-stable, then it contains $x\vec{M}$ for all $\vec{M} \in \Lambda$ and therefore $x\vec{M} \in X \Rightarrow^{\mathsf{W}} Y$ for all $\vec{M} \in \Lambda$ and for all $X \subseteq \Lambda$. Similarly $x\vec{M} \in X \Rightarrow Y$ for all $\vec{M} \in \mathsf{T}$ and for all $X \subseteq \Lambda$ whenever Y is T-type-stable for $\mathsf{T} \in \{\mathsf{HN}, \mathsf{N}, \mathsf{SN}\}$. We conclude using points (i) and (ii). □

Now we define the type environments which will be considered in the completeness part of the proof of Theorem 4.2.2.

Definition 4.3.8 *(Type Environments)*

i) The type environment $\mathcal{V}_{\mathcal{CDV}}$ *is defined by:*

$$\mathcal{V}(A) = \mathsf{SN} \text{ if } A \in \mathbf{C}_\infty.$$

ii) The type environment $\mathcal{V}^1_{\mathcal{BCD}}$ *is defined by:*

$$\mathcal{V}(A) = \mathsf{HN} \text{ if } A \in \mathbf{C}_\infty.$$

iii) The type environment $\mathcal{V}^2_{\mathcal{BCD}}$ *is defined by:*

$$\mathcal{V}(A) = \mathsf{N} \text{ if } A \in \mathbf{C}_\infty.$$

iv) The type environment $\mathcal{V}_{\mathcal{CDZ}}$ *is defined by:*

$$\mathcal{V}(\omega) = \mathsf{PN}; \quad \mathcal{V}(\varphi) = \mathsf{N}.$$

v) The type environment $\mathcal{V}_{\mathcal{DHM}}$ *is defined by:*

$$\mathcal{V}(\omega) = \mathsf{PHN}; \quad \mathcal{V}(\varphi) = \mathsf{HN}.$$

vi) The type environment $\mathcal{V}_{\mathcal{S}c}$ *is defined by:*

$$\mathcal{V}(\omega) = \mathsf{PHN}.$$

vii) The type environment $\mathcal{V}_{\mathcal{P}a}$ *is defined by:*

$$\mathcal{V}(\omega) = \mathsf{C}.$$

Notation 4.3.9 *$\mathcal{V}_{\mathcal{BCD}}$ stands for both $\mathcal{V}^1_{\mathcal{BCD}}$ and $\mathcal{V}^2_{\mathcal{BCD}}$.*

It is easy to verify, using the following Propositions 4.3.10 and 4.3.11, that each type environment $\mathcal{V}_{\nabla}$ above agrees (or W-agrees) with the corresponding type theory Σ^{∇}. Moreover all type environments agree and W-agree with the type theory $\Sigma^{\mathcal{AO}}$: this follows from Proposition 4.3.11(iii) taking into account the interpretations of Ω and $\Omega \to \Omega$ (see Definition 4.3.2 and the following sentence).

Proposition 4.3.10 *i)* $\mathsf{PN} = \mathsf{N} \Rightarrow \mathsf{PN}$.

ii) $\mathsf{PHN} = \Lambda \Rightarrow \mathsf{PHN}$.

iii) $\mathsf{N} = \mathsf{PN} \Rightarrow \mathsf{N}$.

iv) $\mathsf{HN} = \mathsf{PHN} \Rightarrow \mathsf{HN}$.

v) $\mathsf{C} = \mathsf{C} \Rightarrow \mathsf{C}$.

Proof. All cases are immediate but the inclusion $\mathsf{N} \subseteq \mathsf{PN} \Rightarrow \mathsf{N}$. We show that if $M \in \mathsf{PN}$ and $N \in \mathsf{NF}$ then $NM \in \mathsf{N}$. If N is λ-free, i.e. N is of the shape $x\vec{N}$, then NM is the normal form $x\vec{N}M$. Otherwise let $N \equiv \lambda x.N'$. The proof is by induction on the number of occurrences of x in N'. The basic step, that is x does not occur in N', is immediate since $NM \longrightarrow_\beta N'$. If x occurs in N', let $N' \equiv C[x]$, where the hole in $C[\,]$ identifies the left-most occurrence of x in N'. Let y be fresh: by induction $(\lambda x.C[y])M \twoheadrightarrow_\beta C'[y] \in \mathsf{NF}$. By construction there is exactly one hole in $C'[\ ]$. Let $\vec{N}$ be all the terms to which $[\ ]$ is applied in $C'[\ ]$. Since $M \in \mathsf{PN}$, $M\vec{N} \in \mathsf{N}$ and therefore $(\lambda y.C'[y])M \in \mathsf{N}$ too. We conclude $NM \in \mathsf{N}$ since $NM =_\beta (\lambda xy.C[y])MM =_\beta (\lambda y.C'[y])M$. □

Proposition 4.3.11 *i) For $\nabla \in \{\mathcal{BCD}, \mathcal{CDZ}, \mathcal{S}c, \mathcal{P}a, \mathcal{DHM}\}$ and for all types $A \in \mathbf{T}^{\nabla}$, all $M, N \in \Lambda$:*

$$\text{If } M[x := N] \in [\![A]\!]_{\mathcal{V}_\nabla} \text{ then } (\lambda x.M)N \in [\![A]\!]_{\mathcal{V}_\nabla}.$$

ii) For all types $A \in \mathbf{T}^{\mathcal{CDV}}$ and all $M \in \Lambda$, all $N \in \mathsf{SN}$:

$$\text{If } M[x := N] \in [\![A]\!]_{\mathcal{V}_{\mathcal{CDV}}} \text{ then } (\lambda x.M)N \in [\![A]\!]_{\mathcal{V}_{\mathcal{CDV}}}.$$

iii) For all types $A \in \mathbf{T}^{\mathcal{AO}}$, all $M, N \in \Lambda$ and all environments $\mathcal{V}$:

$$\text{If } M[x := N] \in [\![A]\!]^{\mathsf{W}}_{\mathcal{V}} \text{ then } (\lambda x.M)N \in [\![A]\!]^{\mathsf{W}}_{\mathcal{V}}.$$

Proof. The proofs by induction on the structure of A follow from Definition 4.3.8 and Proposition 4.3.7(i),(ii). □

Proof. Proof of Theorem 4.2.2($\Leftarrow$). Take $\rho_0(x) = x$. Notice that $\rho_0, \mathcal{V} \models \Gamma$ and $\rho_0, \mathcal{V} \models_{\mathsf{W}} \Gamma$ for all $\mathcal{V}$ and Γ such that if $x{:}B \in \Gamma$ then either $[\![B]\!]_{\mathcal{V}}$ is Λ or $[\![B]\!]_{\mathcal{V}}$ is S-type-stable for some $\mathsf{S} \in \{\mathsf{WN}, \mathsf{HN}, \mathsf{N}, \mathsf{SN}\}$, since in both cases $[\![B]\!]_{\mathcal{V}}$ will contain all free variables.

(1.iv) It is easy to check using Proposition 4.3.7 that for all $A \in \mathbf{T}^{\mathcal{AO}}$ and all $\mathcal{V}$ either $A \sim_{\mathcal{AO}} \Omega$ and $[\![A]\!]^{\mathsf{W}}_{\mathcal{V}} = \Lambda$ or $A \not\sim_{\mathcal{AO}} \Omega$ and $[\![A]\!]^{\mathsf{W}}_{\mathcal{V}}$ is WN-type-stable. From above we get $\rho_0, \mathcal{V} \models_{\mathsf{W}} \Gamma$ for all $\mathcal{V}$ and Γ. Moreover $[\![\Omega \to \Omega]\!]^{\mathsf{W}}_{\mathcal{V}} = \mathsf{WN}$. Then from $\Gamma \vdash^{\mathcal{AO}}_{\Omega} M : \Omega \to \Omega$ we get by soundness $\Gamma \models^{\mathcal{AO}}_{\mathsf{W}} M : \Omega \to \Omega$, i.e. $M = [\![M]\!]_{\rho_0} \in [\![\Omega \to \Omega]\!]^{\mathsf{W}}_{\mathcal{V}} \subseteq \mathsf{WN}$, so we conclude $M \in \mathsf{WN}$.

(1.iii) For $\lambda\cap^{\mathcal{BCD}}_{\Omega}$ it is easy to check using Proposition 4.3.7 that for all $A \in \mathbf{T}^{\mathcal{BCD}}$ either $A \sim_{\mathcal{BCD}} \Omega$ and $[\![A]\!]_{\mathcal{V}^1_{\mathcal{BCD}}} = \Lambda$ or $A \not\sim_{\mathcal{BCD}} \Omega$ and $[\![A]\!]_{\mathcal{V}^1_{\mathcal{BCD}}}$ is HN-type-stable. So we have $\rho_0, \mathcal{V}^1_{\mathcal{BCD}} \models \Gamma$ for all Γ. From $\Gamma \vdash^{\mathcal{BCD}}_{\Omega} M : (\Omega^m \to A)^n \to A$ we get by soundness $\Gamma \models^{\mathcal{BCD}} M : (\Omega^m \to A)^n \to A$, i.e. $M = [\![M]\!]_{\rho_0} \in \mathsf{HN}$.
For $\lambda\cap^{\mathcal{DHM}}_{\Omega}$ let Γ_ω be the $\mathcal{DHM}$-basis all whose predicates are ω. By Definition 4.1.2 each free variable belongs to PHN and therefore $\rho_0, \mathcal{V}_{\mathcal{DHM}} \models \Gamma_\omega$. From $\Gamma_\omega \vdash^{\mathcal{DHM}} M : \varphi$ we get by soundness $M \in \mathsf{HN}$.

(1.ii) For $\lambda\cap^{\mathcal{BCD}}_{\Omega}$ observe that by Proposition 4.3.7 $[\![A]\!]_{\mathcal{V}^2_{\mathcal{BCD}}}$ is N-type-stable whenever Ω does not occur in A. Therefore $\rho_0, \mathcal{V}^2_{\mathcal{BCD}} \models \Gamma$, since by hypothesis Ω does not occur in Γ. So as in case (1.iii) we get by soundness $M \in \mathsf{N}$.
For $\lambda\cap^{\mathcal{CDZ}}_{\Omega}$ the proof is similar to that of case (1.iii) for $\lambda\cap^{\mathcal{DHM}}_{\Omega}$.

(1.i) The proof is similar to that of case (1.ii) for $\lambda\cap^{\mathcal{BCD}}_{\Omega}$ by observing that $[\![A]\!]_{\mathcal{V}_{\mathcal{CDV}}}$ is SN-type-stable for all $A \in \mathbf{T}^{\mathcal{CDV}}$.

(2.iii) By definition $[\![\Omega^{n+1} \to \Omega]\!]^{\mathsf{W}}_{\mathcal{V}} = \{M \in \mathsf{WN} \mid \forall \vec{N} \text{ of length } n, M\vec{N} \in \mathsf{WN}\}$, then $M \in \bigcap_{n \in \mathbf{N}} [\![\Omega^n \to \Omega]\!]^{\mathsf{W}}_{\mathcal{V}}$ if and only if $M \in \mathsf{PWN}$. The result follows by soundness as in case (1.iv).

(2.ii) For $\lambda\cap^{\mathcal{S}c}_{\Omega}$ first notice that $\omega \leq_{\mathcal{S}c} A$ for all $A \in \mathbf{T}^{\mathcal{S}c}$. This can be easily checked by induction on A. If $A \equiv B \to C$ then by induction $\omega \leq_{\mathcal{S}c} C$ so we get $\omega \sim_{\mathcal{S}c} \Omega \to \omega \leq_{\mathcal{S}c} B \to C$ by rule (η) since $B \leq_{\mathcal{S}c} \Omega$ by axiom (Ω). If for all $A \in \mathbf{T}^{\mathcal{S}c}$ there are a $\mathcal{S}c$-basis Γ and an integer n such that $\Gamma \vdash^{\mathcal{S}c}_{\Omega} M : \Omega^n \to A$, by choosing $A \equiv \omega$ we get that there is a $\mathcal{S}c$-basis Γ_0 such that $\Gamma_0 \vdash^{\mathcal{S}c}_{\Omega} M : \omega$ by rule $(\leq_{\mathcal{S}c})$ since $\Omega^n \to \omega \sim_{\mathcal{S}c} \omega$. This implies $\Gamma_\omega \vdash^{\mathcal{S}c}_{\Omega} M : \omega$ by rule $(\leq_{\mathcal{S}c}$ L). So it suffices to show that $\Gamma_\omega \vdash^{\mathcal{S}c}_{\Omega} M : \omega$ implies $M \in \mathsf{PHN}$. This can be proved similarly to case (1.ii) for $\lambda\cap^{\mathcal{CDZ}}_{\Omega}$ using the type interpretation $\mathcal{V}_{\mathcal{S}c}$.
For $\lambda\cap^{\mathcal{DHM}}_{\Omega}$ the proof is similar since $\omega \leq_{\mathcal{DHM}} \varphi$ and $\omega \sim_{\mathcal{DHM}} \Omega \to \omega$.

(2.i) The first observation is that $\Omega \in A$ iff $\Omega \in_{\mathcal{CDZ}} A$. We now show that $\omega \leq_{\mathcal{CDZ}} A \leq_{\mathcal{CDZ}} \varphi$ for all $A \in \mathbf{T}^{\mathcal{CDZ}}$ such that $\Omega \notin A$ by induction on A. The only interesting case is $A \equiv B \to C$: in this case by induction $\omega \leq_{\mathcal{CDZ}} B \leq_{\mathcal{CDZ}} \varphi$, $\omega \leq_{\mathcal{CDZ}} C \leq_{\mathcal{CDZ}} \varphi$ so we get $\omega \sim_{\mathcal{CDZ}} \varphi \to \omega \leq_{\mathcal{CDZ}} B \to C \sim_{\mathcal{CDZ}} \omega \to \varphi \sim_{\mathcal{CDZ}} \varphi$ by rule (η). If for all $A \in \mathbf{T}^{\mathcal{CDZ}}$ such that $\Omega \notin A$ there is a $\mathcal{CDZ}$-basis Γ such that $\Omega \notin \Gamma$ and $\Gamma \vdash^{\mathcal{CDZ}}_{\Omega} M : A$, by choosing $A \equiv \omega$ we get that there is a $\mathcal{CDZ}$-basis Γ_0 such that $\Omega \notin \Gamma_0$ and $\Gamma_0 \vdash_{\Omega}^{\mathcal{CDZ}} M : \omega$. This implies $\Gamma_\omega \vdash_{\Omega}^{\mathcal{CDZ}} M : \omega$ by rule ($\leq_{\mathcal{CDZ}}$ L). So it suffices to show that $\Gamma_\omega \vdash_{\Omega}^{\mathcal{CDZ}} M : \omega$ implies $M \in \mathsf{PN}$. This can be proved similarly to case (1.ii) for $\lambda\cap_{\Omega}^{\mathcal{CDZ}}$.

(3) Clearly $\rho, \mathcal{V} \models \emptyset$ for all $\rho, \mathcal{V}$. The result follows immediately by soundness.

$\square$

4.4 Polarised normal forms

In this section we will show that, for all arrow-type theories Σ^{∇}, each persistently normalising λ-term M can be typed with an arbitrary type not containing Ω modulo $\sim_{\nabla}$ from a suitable ∇-basis. Our proof is organised as follows. First we introduce the notions of adjacent occurrences of variables, positive and negative variables, polarised normal forms, principal decorations and replacement paths. Then we show the key property (Lemma 4.4.13):

for each normal form with adjacent occurrences of negative variables we can build a substitution such that the resulting term does not have normal form.

This fact suggests the notions of positive normal forms and strongly polarised normal forms. We conclude by showing that:

- each persistently normalising normal form is a positive normal form (Theorem 4.4.15);
- the principal decoration of a positive normal form is a strongly polarised normal form (Proposition 4.4.20);
- each strongly polarised normal form which is a principal decoration can be typed with an arbitrary type not containing Ω modulo $\sim_{\nabla}$ from a suitable ∇-basis in all arrow-type theories Σ^{∇} (Theorem 4.4.23).

We give now some definitions concerning only terms in normal form. We do forbid α-conversion: in this way also the names of bound variables are meaningful. Moreover this leads us to consider λ-terms in which different bound variables may have the same names, and also bound and free variables may have the same name.

Definition 4.4.1 *i) In a normal form of the shape $x\vec{M}(\lambda\vec{z}.y\vec{N})$ we say that the showed occurrences of x and y are* adjacent. *Notice that we can have $x \equiv y$.*

ii) Two (not necessary distinct!) variables have adjacent occurrences in a normal form M if and only if they have adjacent occurrences in a subterm of M.

iii) If $M \equiv xN_1 \dots N_i \dots N_m$ we say that the subterm N_i is the i-th argument of x in M.

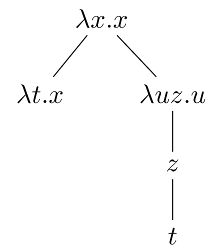

Figure 4.2: Böhm tree of $\lambda x.x(\lambda t.x)(\lambda uz.u(zt))$.

iv) If $M \equiv \lambda y_1 \ldots y_j \ldots y_n.x\vec{N}$ *we say that:*

(a) the variables $y_1 \ldots y_j \ldots y_n$ *are the variable bound by the initial abstractions of* M*;*

(b) the variable y_j *is the variable bound by the* j*-th abstraction of* M*.*

Remark 4.4.2 *An alternative definition of adjacent occurrences can be done using the Böhm trees of* λ*-terms as defined in [Bar84] (Definition 10.1.4): two occurrences* x, y *are adjacent in* M *if and only if they correspond to two nodes father-son in the Böhm tree of* M *with labels* $\lambda\vec{z}.x$ *and* $\lambda\vec{t}.y$ *for some* $\vec{z}, \vec{t}$*.*

Example 4.4.3 *In the normal form* $\lambda x.x(\lambda t.x)(\lambda uz.u(zt))$*:*

- *the underlined occurrences of variables are adjacent:*

$$\lambda x.\underline{x}(\lambda t.\underline{x})(\lambda uz.u(zt)) \qquad \lambda x.\underline{x}(\lambda t.x)(\lambda uz.\underline{u}(zt))$$

$$\lambda x.x(\lambda t.x)(\lambda uz.\underline{u}(\underline{z}t)) \qquad \lambda x.x(\lambda t.x)(\lambda uz.u(\underline{z}\underline{t}))$$

- $\lambda uz.u(zv)$ *is the 2-th argument of* x *in* $x(\lambda t.x)(\lambda uz.u(zt))$
- x *is the variable bound by the initial abstraction of* $\lambda x.x(\lambda t.x)(\lambda uz.u(zt))$
- z *is the variable bound by the 2-th abstraction of* $\lambda uz.u(zt)$*.*

Figure 4.2 shows the Böhm tree of $\lambda x.x(\lambda t.x)(\lambda uz.u(zt))$*.*

We need to introduce the notion of *polarity* for term variables.

Definition 4.4.4 (Polarised normal forms) *Assume that the variables of* λ*-calculus are partitioned into two infinite sets of positive and negative variables, i.e.* x^+ *or* x^-*. Let* $\Lambda^{\pm}$ *be the resulting language of polarised* λ*-terms, and define the set of* polarised normal forms, $\mathsf{NF}^{\mathsf{i},\mathsf{j}}$ *as follows:*

$$(+\ app)\ \frac{\vec{M} \in \mathsf{NF}^{+,+} \cup \mathsf{NF}^{+,-}}{x^+\vec{M} \in \mathsf{NF}^{+,+} \cap \mathsf{NF}^{-,+}} \qquad (-\ app)\ \frac{\vec{M} \in \mathsf{NF}^{-,+} \cup \mathsf{NF}^{-,-}}{x^-\vec{M} \in \mathsf{NF}^{+,-} \cap \mathsf{NF}^{-,-}}$$

$$(+\ abs)\ \frac{M \in \mathsf{NF}^{+,\mathsf{j}}}{\lambda x^+.M \in \mathsf{NF}^{+,\mathsf{j}}} \qquad (-\ abs)\ \frac{M \in \mathsf{NF}^{-,\mathsf{j}}}{\lambda x^-.M \in \mathsf{NF}^{-,\mathsf{j}}}$$

Notice that $\mathsf{NF}^{\mathsf{i},\mathsf{j}} \subseteq \mathsf{NF}$ for all $\mathsf{i},\mathsf{j} \in \{+,-\}$.

Example 4.4.5 *We can derive $\lambda x^+.x^+x^+ \in \mathsf{NF}^{+,+}$ as follows:*

$$\dfrac{\dfrac{\dfrac{}{x^+ \in \mathsf{NF}^{+,+}}\ (+\ app)}{x^+x^+ \in \mathsf{NF}^{+,+}}\ (+\ app)}{\lambda x^+.x^+x^+ \in \mathsf{NF}^{+,+}}\ (+\ abs)$$

Similarly we can derive $\lambda x^-.x^-x^- \in \mathsf{NF}^{-,-}$.

Rule $(+\ app)$ says that we can apply a positive variable only to normal forms whose initial bound variables are positive independently from the polarity of the head variables. The so obtained normal form belongs both to $\mathsf{NF}^{+,+}$ and $\mathsf{NF}^{-,+}$: in fact it is λ-free and it's head variable is positive. Similarly rule $(-\ app)$ allows to apply a negative variable to all normal forms whose initial bound variables are negative independently from the polarity of the head variables. The rules for abstractions force all consecutive abstractions to have the same sign.

In other words, $M \in \mathsf{NF}^{\mathsf{i},\mathsf{j}}$ means that the variables bound by the initial abstractions of M have polarity i, the head variable of M has polarity j and the components of M belong to $\mathsf{NF}^{\mathsf{j},+}$ or to $\mathsf{NF}^{\mathsf{j},-}$. A λ-free normal form can belong to both $\mathsf{NF}^{+,\mathsf{j}}$ and $\mathsf{NF}^{-,\mathsf{j}}$, since we do not know the polarity of missing bound variables.

Remark 4.4.6 *Looking at the Böhm tree of a polarised normal form we always have that:*

- *the variables abstracted in the same node have the same polarities;*
- *if the node $\lambda\vec{z}.x^{\mathsf{i}}$ is the father of the node $\lambda t_1^{\mathsf{j}} \ldots t_n^{\mathsf{j}}.y$ then $\mathsf{i} = \mathsf{j}$.*

There is a natural way of associating polarised normal forms to normal forms.

Definition 4.4.7 (Decoration) *A polarised normal form $N \in \Lambda^{\pm}$ is a* decoration *of a normal form $M \in \Lambda$ if and only if M is obtained from N by erasing all polarities.*

Example 4.4.5 shows that a normal form can have more than one decoration. To get a one-one correspondence between polarised normal forms and normal forms it suffices to force the polarities of the free variables and of the variables bound by the initial abstractions.

Definition 4.4.8 (Principal Decoration) *A polarised normal form $N \in \Lambda^{\pm}$ is the* principal decoration *of a normal form $M \in \Lambda$ if and only if:*

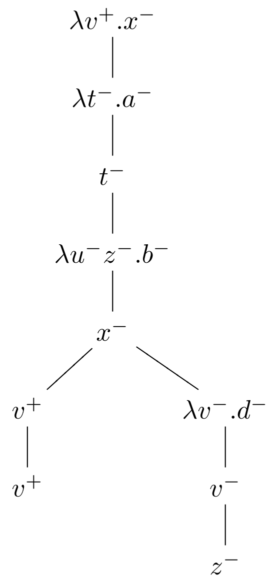

Figure 4.3: Böhm tree of the principal decoration of $\lambda v.x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d(vz))))))$.

i) N is a decoration of M;

ii) all variables bound by the initial abstractions of N are positive;

iii) all free variables in N are negative.

Example 4.4.9 *$\lambda x^+.x^+x^+$ is a principal decoration of $\lambda x.xx$, but $\lambda x^-.x^-x^-$ is not. The principal decoration of $\lambda v.x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d(vz))))))$ is*

$$\lambda v^+.x^-(\lambda t^-.a^-(t^-(\lambda u^- z^-.b^-(x^-(v^+v^+)(\lambda v^-.d^-(v^-z^-))))))$$

(see Figure 4.3).

It is easy to check the soundness of previous definition, i.e. that the principal decoration of a normal form is unique, since according to Definition 4.4.4 the polarities of the variables abstracted in proper subterms are uniquely determined. More precisely if $x^{\mathsf{i}}\vec{N}$ is a subterm of N then all variables abstracted in the initial abstractions of the terms in $\vec{N}$ must have polarity i. Clearly all principal decorations belong $\mathsf{NF}^{+,+} \cup \mathsf{NF}^{+,-}$.

Remark 4.4.10 *We can build the principal decoration of a normal form M by using the Böhm tree of M as follows. First we give positive polarities to all variables bound by the initial abstractions of the root and negative polarities to all free variables. Then* *we propagate polarities by giving the polarity* i *to all variables bound in a node whose father has head variables of polarity* i.

The above discussion allow us to identify normal forms with their principal decorations. So from now on until the end of this subsection **we convene that variables in normal forms have the polarities of the corresponding principal decorations**.

We need to introduce the *replacement path* of an occurrence of a negative variable in a normal form. This notion was first defined in a less formal way and for the same aim in [CDCZ87]. Intuitively the replacement path of an occurrence of a negative variable says if that occurrence is free or bound, and in the last case where it is bound. This is useful in order to replace that occurrence using substitutions of free variables.

The replacement path of a free occurrence of a variable is the variable itself. The replacement path of a bound occurrence of a negative variable is the name of a free variable (hence of a negative variable) followed by a sequence of integer pairs. If the replacement path of a given occurrence of a variable y in a normal form M is $x\langle i_1, j_1\rangle \dots \langle i_n, j_n\rangle$, then there is an occurrence of x in M such that if z_1 is the variable bound by the j_1-th abstraction of the i_1-th argument of x, then there is an occurrence of z_1 in M such that if z_2 is the variable bound by j_2-th abstraction of the i_2 argument of z_1, then ... there is an occurrence of z_{n-1} in M such that y is the variable bound by the j_n-abstraction of the i_n-th argument of z_{n-1}. A constructive definition is Definition 4.4.11.

As usual $\mathsf{C}[\,]$ will denote a context. We convene that the hole $[\,]$ occurs only once in $\mathsf{C}[\,]$ and that $\mathsf{C}[\,]$ is in normal form (we say that $\mathsf{C}[\,]$ is a *normal context*). In this way if $M \equiv \mathsf{C}[x]$ then $\mathsf{C}[\,]$ uniquely identifies one occurrence of x in the normal form M.

Definition 4.4.11 *i) The* replacement path $\pi(x, \mathsf{C}[\,])$ *of a variable x in a normal context $\mathsf{C}[\,]$ is defined by:*

$$\frac{\textit{if the given occurrence of } x \textit{ is free in } \mathsf{C}[x]}{\pi(x, \mathsf{C}[\,]) = x}$$

$$\frac{\textit{if } \pi(x, \mathsf{C}[\,]) \textit{ is defined}}{\pi(x, y\vec{N}\mathsf{C}[\,]\vec{N'}) = \pi(x, \mathsf{C}[\,])}$$

$$\frac{\pi(x, \mathsf{C}[\,]) = y\alpha \textit{ and } z \neq y}{\pi(x, \lambda z.\mathsf{C}[\,]) = y\alpha}$$

$$\frac{\pi(x, \mathsf{C}[\,]) = y_j\alpha}{\pi(x, zN_1 \dots N_{i-1}(\lambda y_1 \dots y_j \dots y_n.\mathsf{C}[\,])N_{i+1} \dots N_m) = z\langle i, j\rangle\alpha}$$

ii) The replacement path of a given occurrence of a variable x in a normal form M, identified by $\mathsf{C}[\,]$, is $\pi(x, \mathsf{C}[\,])$.

iii) A variable x occurs with replacement path $y\alpha$ in a normal form M if and only if $M \equiv \mathsf{C}[x]$ and $\pi(x, \mathsf{C}[\,]) = y\alpha$ for some context $\mathsf{C}[\,]$.

$$\begin{array}{c}
v \text{ is free in } d(vz) \\ \hline
\pi(v, d([\,]z)) = v \\ \hline
\pi(v, x(vv)(\lambda v.d([\,]z))) = x\langle 2,1\rangle \\ \hline
\pi(v, b(x(vv)(\lambda v.d([\,]z)))) = x\langle 2,1\rangle \\ \hline
\pi(v, \lambda z.b(x(vv)(\lambda v.d([\,]z)))) = x\langle 2,1\rangle \\ \hline
\pi(v, \lambda uz.b(x(vv)(\lambda v.d([\,]z)))) = x\langle 2,1\rangle \\ \hline
\pi(v, t(\lambda uz.b(x(vv)(\lambda v.d([\,]z))))) = x\langle 2,1\rangle \\ \hline
\pi(v, a(t(\lambda uz.b(x(vv)(\lambda v.d([\,]z)))))) = x\langle 2,1\rangle \\ \hline
\pi(v, \lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d([\,]z)))))) = x\langle 2,1\rangle \\ \hline
\pi(v, x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d([\,]z))))))) = x\langle 2,1\rangle \\ \hline
\pi(v, \lambda v.x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d([\,]z))))))) = x\langle 2,1\rangle
\end{array}$$

$$\begin{array}{c}
z \text{ is free in } b(x(vv)(\lambda v.d(vz))) \\ \hline
\pi(z, b(x(vv)(\lambda v.d(v[\,])))) = z \\ \hline
\pi(z, t(\lambda uz.b(x(vv)(\lambda v.d(v[\,]))))) = t\langle 1,2\rangle \\ \hline
\pi(z, a(t(\lambda uz.b(x(vv)(\lambda v.d(v[\,])))))) = t\langle 1,2\rangle \\ \hline
\pi(z, x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d(v[\,]))))))) = x\langle 1,1\rangle\langle 1,2\rangle \\ \hline
\pi(v, \lambda v.x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d(v[\,]))))))) = x\langle 1,1\rangle\langle 1,2\rangle
\end{array}$$

Figure 4.4: Examples of replacement paths.

Example 4.4.12 *Let*

$$\begin{array}{lcl} \mathsf{C}_1[\,] & \equiv & \lambda v.[\,](\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d(vz)))))) \\ \mathsf{C}_2[\,] & \equiv & \lambda v.x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d([\,]z)))))) \\ \mathsf{C}_3[\,] & \equiv & \lambda v.x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d(v[\,])))))) \\ \mathsf{C}_4[\,] & \equiv & \lambda v.x(\lambda t.a(t(\lambda uz.b(x([\,]v)(\lambda v.d(vz)))))). \end{array}$$

Then $\pi(x, \mathsf{C}_1[\,]) = x$, $\pi(v, \mathsf{C}_2[\,]) = x\langle 2, 1\rangle$, $\pi(z, \mathsf{C}_3[\,]) = x\langle 1, 1\rangle\langle 1, 2\rangle$, *while* $\pi(v, \mathsf{C}_1[\,])$ *and* $\pi(v, \mathsf{C}_4[\,])$ *are undefined. Figure 4.4 shows the derivations of* $\pi(v, \mathsf{C}_2[\,]) = x\langle 2, 1\rangle$ *and* $\pi(z, \mathsf{C}_3[\,]) = x\langle 1, 1\rangle\langle 1, 2\rangle$. *Figure 4.3 gives the Böhm tree of the principal decoration of* $\lambda v.x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d(vz))))))$ *which is* $\mathsf{C}_1[x]$, $\mathsf{C}_2[v]$, $\mathsf{C}_3[z]$ *and* $\mathsf{C}_4[v]$.

The first case of definition 4.4.11 is the basic step in computing replacement paths. The second and third cases simply allow to inherit replacement paths from subterms. The crucial case is the last one, which builds the replacement path in a context from that of a proper sub-context taking into account where the first variable in the given path is bound. Notice that the second and fourth cases are mutually exclusive since $\pi(x, \mathsf{C}[\,]) = y_j\alpha$ implies that $\pi(x, \lambda y_1 \dots y_j \dots y_n.\mathsf{C}[\,])$ is undefined, being (the current occurrence of) y_j positive in $\lambda y_1 \dots y_j \dots y_n.\mathsf{C}[x]$.

Notice that the replacement path of an occurrence can be undefined in a subterm and defined in the whole term. This comes from the last clause of Definition 4.4.11. In Example 4.4.12 $\pi(v, \lambda v.d([\,]z))$ is undefined, while $\pi(v, \mathsf{C}_2[\,]) = x\langle 2, 1\rangle$.

It is easy to verify that all occurrences of negative variables have defined replacement paths. Vice versa all occurrences of positive variables have undefined replacement paths

We can now put replacement paths to use in order to state and to prove the key property that if two negative variables have adjacent occurrences in a normal form M then we can replace the free variables of M by normal forms such that the so obtained term does not have a normal form.

Lemma 4.4.13 *If there are two adjacent occurrences of the (negative) variables* z, t *in a normal form* M *with replacements paths* $x\alpha$, $y\beta$, *then there are normal forms* X, Y *such that* $M[x := X, y := Y]$ *does not have a normal form (possibly* $x \equiv y$ *and, in this case,* $X \equiv Y$*).*

Proof. Actually we prove a stronger statement, i.e. we require X and Y to be λ-terms in which all abstracted variables occur at least once as arguments of a free variable. This makes sure that all subterms of all terms obtained out of $M[x := X, y := Y]$ by reduction will never be erased. So we need only build a reduct of $M[x := X, y := Y]$ containing an unsolvable subterm.

The proof is by induction on the sum of the lengths of the current replacement paths, i.e. of α and β. We convene that (the current occurrence of) z is on the left of (the current occurrence of) t in M.

Basic step: In this case α and β are empty, and therefore $x \equiv z$ and $y \equiv t$. By definition M has a subterm of the shape $xN_1 \dots N_{i-1}(\lambda u_1 \dots u_n.yN'_1 \dots N'_m)$. If

$x \not\equiv y$ a possible choice is

$$X \equiv \lambda v_1 \dots v_i.av_1 \dots v_i(v_i u_1 \dots u_n \Delta)$$

$$Y \equiv \lambda w_1 \dots w_m w.bw_1 \dots w_m w(ww)$$

where $\Delta \equiv \lambda w.ww$. Since

$$\begin{aligned}&(xN_1 \dots N_{i-1}(\lambda u_1 \dots u_n.yN'_1 \dots N'_m))[x := X, y := Y] \\ &\quad \twoheadrightarrow_\beta a\hat{N}_1 \dots \hat{N}_{i-1}(\lambda u_1 \dots u_n w.b\hat{N}'_1 \dots \hat{N}'_m w(ww))(b\hat{N}'_1 \dots \hat{N}'_m \Delta(\Delta\Delta))\end{aligned}$$

where $\hat{N} = N[x := X, y := Y]$ (possibly with indexes and $'$), then $M[x := X, y := Y]$ has an unsolvable subterm.

If $x \equiv y$ we can choose

$$X \equiv \lambda v_1 \dots v_k.av_1 \dots v_k(v_{m+1}v_{m+1})(v_i u_1 \dots u_n(v_i u_1 \dots u_n)),$$

where k is the maximum between i and $m+1$. We get

$$\begin{aligned}&(xN_1 \dots N_{i-1}(\lambda u_1 \dots u_n.xN'_1 \dots N'_m))[x := X] \\ &\quad \twoheadrightarrow_\beta \lambda v_{i+1} \dots v_k.a\hat{N}_1 \dots \hat{N}_{i-1}(\lambda u_1 \dots u_n.Q)v_{i+1} \dots v_k(PP)(QQ)\end{aligned}$$

where $\hat{N} = N[x := X]$ (possibly with indexes and $'$), $Q \equiv \lambda v_{m+1} \dots v_k.a\hat{N}'_1 \dots \hat{N}'_m v_{m+1} \dots v_k(v_{m+1}v_{m+1})(Ru_1 \dots u_n(Ru_1 \dots u_n))$, P, R are suitable terms and

$$QQ \twoheadrightarrow_\beta \lambda v_{m+2} \dots v_k.aN'_1 \dots N'_m Q v_{m+2} \dots v_k(QQ)(Ru_1 \dots u_n(Ru_1 \dots u_n)).$$

Since QQ reduces to a term containing QQ, $M[x := X]$ does not have a normal form.

Induction step: we need to distinguish five possible cases:

i) $x \not\equiv y$ and α not empty;

ii) $x \not\equiv y$ and α empty;

iii) $x \equiv y$ and α, β both not empty;

iv) $x \equiv y$ and α empty while β not empty;

v) $x \equiv y$ and α not empty while β empty.

In all cases we exhibit a normal form X' such that there are adjacent occurrences in the normal form M' of $M[x := X']$ but the sum of the lengths of the replacement paths of these occurrences is less than the sum of the lengths of α and β. This allows us to apply the induction.

i) *Case $x \not\equiv y$ and α not empty.* Assume that $\alpha = \langle i, j\rangle\alpha'$. Then M has a subterm of the shape $xN_1 \dots N_{i-1}(\lambda u_1 \dots u_j.N')$ and $u_j\alpha'$ is the replacement path of the current occurrence of z in N'. Notice that $z \equiv u_j$ if α' is empty. Let

$$X' \equiv \lambda v_1 \dots v_i.xv_1 \dots v_i(v_i u_1 \dots u_j)$$

and M' be the normal form of $M[x := X']$ (the existence of M' comes from the fact that X' in $M[x := X']$ is only applied to free variables). On M' we can observe that:

- the occurrences which are adjacent in M remain adjacent in M';
- the variables which occur with replacement path $x\langle i, l\rangle\gamma$ where $1 \leq l \leq j$ in M occur also with replacement path $u_l\gamma$ in M';
- the variables which occur with replacement path $y\gamma$ occur with the same replacement path in M'.

By the above observations there are adjacent occurrences of z, t in M' with replacement paths respectively $u_j\alpha'$ and $y\beta$. Then induction hypothesis applies and we can find normal forms U_j, Y such that $M'[u_j := U_j, y := Y]$ does not reduce to a normal form. Therefore we can choose

$$X \equiv \lambda v_1 \dots v_i.xv_1 \dots v_i(v_iu_1 \dots U_j).$$

ii) *Case $x \not\equiv y$ and α empty.* Then $z \equiv x$. Assume that $\beta = \langle i, j\rangle\beta'$. Then M has a subterm of the shape $yN_1 \dots N_{i-1}(\lambda u_1 \dots u_j.N')$ and $u_j\beta'$ is the replacement path of the current occurrence of t in N'. Notice that $t \equiv u_j$ if β' is empty. Let

$$Y' \equiv \lambda v_1 \dots v_i.xv_1 \dots v_i(v_iu_1 \dots u_j)$$

and M' be the normal form of $M[y := Y']$. On M' we can observe that:

- the occurrences which are adjacent in M remain adjacent in M';
- the variables which occur with replacement path $y\langle i, l\rangle\gamma$ where $1 \leq l \leq j$ in M occur also with replacement path $u_l\gamma$ in M'.

By the above observations there are adjacent occurrences of x, t in M' such that the replacement path of t is $u_j\beta'$. Then induction hypothesis applies and we can find normal forms X, U_j such that $M'[x := X, u_j := U_j]$ does not reduce to a normal form. Therefore we can choose

$$Y \equiv \lambda v_1 \dots v_i.yv_1 \dots v_i(v_iu_1 \dots U_j).$$

iii) *Case $x \equiv y$ and α, β both not empty.* Assume that $\alpha = \langle i, j\rangle\alpha'$ and $\beta = \langle n, m\rangle\beta'$. Then M has a subterm of the shape $xM_1 \dots M_{i-1}(\lambda u_1 \dots u_j.M^*)$, such that $u_j\alpha'$ is the replacement path of the current occurrence of z in M^*, and a subterm of the shape $xN_1 \dots N_{n-1}(\lambda v_1 \dots v_m.N^*)$, such that $v_m\beta'$ is the replacement path of the current occurrence of t in N^* (these subterms may coincide). Observe that the current occurrences of z, t must be subterms of both M^* and N^*, and this implies that the showed occurrences of x are either nested or they coincide and $i = n$. Notice that $z \equiv u_j$ if α' is empty and $t \equiv v_m$ if β' is empty.

We first consider the sub-case $i \neq n$. This implies $u_j \not\equiv v_m$. Let

$$X' \equiv \lambda w_1 \dots w_k.xw_1 \dots w_k(w_iu_1 \dots u_j)(w_nv_1 \dots v_m),$$

where k is the maximum between i and n, and let M' be the normal form of $M[x := X']$. On M' we can observe that:

- the occurrences which are adjacent in M remain adjacent in M';
- the variables which occur with replacement path $x\langle i,l\rangle\gamma$ where $1 \leq l \leq j$ in M occur also with replacement path $u_l\gamma$ in M';
- the variables which occur with replacement path $x\langle n,h\rangle\gamma$ where $1 \leq h \leq m$ in M occur also with replacement path $v_h\gamma$ in M'.

By the above observations there are adjacent occurrences of z, t in M' with replacement paths respectively $u_j\alpha'$ and $v_m\beta'$. Then induction hypothesis applies and we can find normal forms U_j, V_m such that $M'[u_j := U_j, v_m := V_m]$ does not reduce to a normal form. Then we can choose

$$X \equiv \lambda w_1 \dots w_k.xw_1 \dots w_k(w_iu_1 \dots U_j)(w_nv_1 \dots V_m).$$

If $i = n$, let

$$X' \equiv \lambda w_1 \dots w_i.xw_1 \dots w_i(w_iu_1 \dots u_k),$$

where k is the maximum between j and m. On the normal form M' of $M[x := X']$ we can observe that:

- the occurrences which are adjacent in M remain adjacent in M';
- the variables which occur with replacement path $x\langle i,l\rangle\gamma$ where $1 \leq l \leq k$ in M occur also with replacement path $u_l\gamma$ in M'.

By the above observations if β' is not empty then there are adjacent occurrences of z, t in M' with replacement paths respectively $u_j\alpha'$ and $u_m\beta'$. If β' is empty then there are adjacent occurrences of z, u_m in M' with replacement paths respectively $u_j\alpha'$ and u_m. In both cases induction hypothesis applies and we can find normal forms U_j, U_m such that $M'[u_j := U_j, u_m := U_m]$ (or $M'[u_j := U_j]$ if $j = m$) does not reduce to a normal form. Then we can choose

$$X = \begin{cases} \lambda w_1 \dots w_i.xw_1 \dots w_i(w_iu_1 \dots U_j \dots U_m), & \text{if } j < m = k \\ \lambda w_1 \dots w_i.xw_1 \dots w_i(w_iu_1 \dots U_j), & \text{if } j = m = k \\ \lambda w_1 \dots w_i.xw_1 \dots w_i(w_iu_1 \dots U_m \dots U_j), & \text{if } m < j = k. \end{cases}$$

iv) *Case $x \equiv y$ and α empty, while β not empty.* Then $z \equiv x$ and $t \not\equiv x$. Assume that $\beta = \langle i,j\rangle\beta'$. Then M has a subterm of the shape $xM_1 \dots M_{i-1}(\lambda u_1 \dots u_j.M^*)$, such that $u_j\beta'$ is the replacement path of t in M^*, and a subterm of the shape $xN_1 \dots N_{k-1}(\lambda v_1 \dots v_n.tN_1' \dots N_m')$ (these subterms may coincide). Notice that $t \equiv u_j$ if β' is empty. Let

$$X' \equiv \lambda w_1 \dots w_i.xw_1 \dots w_i(w_iu_1 \dots u_j)$$

and let M' be the normal form of $M[x := X']$. On M' we can observe that:

- the occurrences which are adjacent in M remain adjacent in M';
- the variables which occur with replacement path $x\langle i,l\rangle\gamma$ where $1 \leq l \leq j$ in M occur also with replacement path $u_l\gamma$ in M'.

By the above observations there are adjacent occurrences of x, t in M' with replacement paths respectively x and $u_j\beta'$. Then induction hypothesis applies and we can find normal forms X'', U_j such that $M'[x := X'', u_j := U_j]$ does not reduce to a normal form. Notice that X'' will be built in the basic step, so

$$X'' \equiv \lambda w_1 \dots w_k.aw_1 \dots w_k(w_k v_1 \dots v_n \Delta).$$

If the subterms $xM_1 \dots M_{i-1}(\lambda u_1 \dots u_j.M^*)$ and $xN_1 \dots N_{k-1}(\lambda v_1 \dots v_n.tN'_1 \dots N'_m)$ do not coincide then the second one must be a subterm of M^*. We can choose

$$X \equiv \lambda w_1 \dots w_h.aw_1 \dots w_h(w_i u_1 \dots u_{j-1}U_j)(w_k v_1 \dots v_n \Delta)$$

where h is the maximum between i and k.

To see why this works, observe that a subterm of (a reduct of) $M[x := X]$ will be $X\hat{N}_1 \dots \hat{N}_{k-1}(\lambda v_1 \dots v_n.T\hat{N}'_1 \dots \hat{N}'_m)$, where T is a subterm of U_j and $\hat{N}$ is a substitution instance of N (possibly with indexes and $'$)[3]. Notice that T is built in the basic step, and therefore

$$T \equiv \lambda r_1 \dots r_m r.br_1 \dots r_m r(rr).$$

Now if $R \equiv \lambda v_1 \dots v_n.T\hat{N}'_1 \dots \hat{N}'_m$ we get

$$X\hat{N}_1 \dots \hat{N}_{k-1}R \twoheadrightarrow_\beta \lambda w_{k+1} \dots w_h.a\hat{N}_1 \dots \hat{N}_{k-1}Rw_{k+1} \dots w_h S(Rv_1 \dots v_n\Delta),$$

$$Rv_1 \dots v_n\Delta \twoheadrightarrow_\beta b\hat{N}'_1 \dots \hat{N}'_m\Delta(\Delta\Delta),$$

where S is a suitable term.

If the subterms $xM_1 \dots M_{i-1}(\lambda u_1 \dots u_j.M^*)$ and $xN_1 \dots N_{k-1}(\lambda v_1 \dots v_n.tN'_1 \dots N'_m)$ coincide we get $i = k$, $M_l \equiv N_l$ $(1 \leq l \leq i)$, $t \equiv u_j$ and $j \leq n$, $u_l \equiv v_l$ $(1 \leq l \leq j)$. In this case we can choose

$$X \equiv \lambda w_1 \dots w_i.aw_1 \dots w_i(w_i u_1 \dots u_{j-1}U_j u_{j+1} \dots u_n\Delta).$$

Notice that U_j is built in the basic step, and therefore

$$U_j \equiv \lambda r_1 \dots r_m r.br_1 \dots r_m r(rr).$$

To see why this works, observe that a subterm of (a reduct of) $M[x := X]$ will be $xN_1 \dots N_{i-1}(\lambda u_1 \dots u_n.u_jN'_1 \dots N'_m)[x := X]$. Now we get

$$X\hat{N}_1 \dots \hat{N}_{i-1}(\lambda u_1 \dots u_n.u_j\hat{N}'_1 \dots \hat{N}'_m)$$

$$\twoheadrightarrow_\beta a\hat{N}_1 \dots \hat{N}_{i-1}(\lambda u_1 \dots u_n.u_j\hat{N}'_1 \dots \hat{N}'_m)(U_j\hat{\hat{N}}'_1 \dots \hat{\hat{N}}'_m\Delta),$$

$$U_j\hat{\hat{N}}'_1 \dots \hat{\hat{N}}'_m\Delta \twoheadrightarrow_\beta b\hat{\hat{N}}'_1 \dots \hat{\hat{N}}'_m\Delta(\Delta\Delta),$$

[3] Following [Bar84] (Definition 10.3.2) a substitution instance of a term P is the result of substituting some terms for some free variables in P.

where $\hat{N} = N[x := X]$ and $\hat{\hat{N}} = \hat{N}[u_j := U_j]$ (possibly with indexes and $'$).

v) *Case $x \equiv y$ and α not empty, while β empty.* Then $z \not\equiv x$ and $t \equiv x$. Assume that $\alpha = \langle i, j\rangle\alpha'$. Then M has a subterm of the shape $xM_1 \dots M_{i-1}(\lambda u_1 \dots u_j.M^*)$, such that the replacement path of the current occurrence of z in M^* is $u_j\alpha'$, and M^* has a subterm of the shape $zN_1 \dots N_{k-1}(\lambda v_1 \dots v_n.xN'_1 \dots N'_m)$. Notice that $z \equiv u_j$ if α' is empty. Let h be the maximum between i and $m+1$ (so $h \geq 1$) and

$$X' \equiv \lambda w_1 \dots w_h.xw_1 \dots w_{h-1}(xw_h(w_iu_1 \dots u_j)).$$

On the normal form M' of $M[x := X']$ we can observe that:

- the occurrences which are adjacent in M remain adjacent in M';
- the variables which occur with replacement path $x\langle i, l\rangle\gamma$ where $1 \leq l \leq j$ in M occur also with replacement path $u_l\gamma$ in M'.

By the above observations there are adjacent occurrences of z, x in M' with replacement paths respectively $u_j\alpha'$ and x. Then induction hypothesis applies and we can find normal forms X'', U_j such that $M'[x := X'', u_j := U_j]$ does not reduce to a normal form. Notice that $\lambda v_1 \dots v_n.xN'_1 \dots N'_m[x := X']$ reduces to a normal form of the shape $\lambda v_1 \dots v_nw_{m+1} \dots w_h.xR_1 \dots R_h$ for suitable normal forms $R_1, \dots, R_h$. Since X'' will be built in the basic step, it will be

$$X'' \equiv \lambda r_1 \dots r_{h+1}.br_1 \dots r_{h+1}(r_{h+1}r_{h+1}).$$

This suggests us to choose

$$X \equiv \lambda s_1 \dots s_{h+1}.xs_1 \dots s_{h-1}(xs_hs_{h+1}(s_iu_1 \dots u_{j-1}U_j)(s_{h+1}s_{h+1})).$$

To see why this works, observe that a subterm of (a reduct of) $M[x := X]$ will be $Z\hat{N}_1 \dots \hat{N}_{k-1}(\lambda v_1 \dots v_n.X\hat{N'}_1 \dots \hat{N'}_m)$ where Z is a subterm of U_j and $\hat{N}$ is a substitution instance of N (possibly with indexes and $'$). Notice that Z is built in the basic step, and therefore

$$Z \equiv \lambda q_1 \dots q_k.aq_1 \dots q_k(q_kv_1 \dots v_nw_{m+1} \dots w_h\Delta).$$

Now if $R \equiv \lambda v_1 \dots v_n.X\hat{N}'_1 \dots \hat{N}'_m$ we get

$$Z\hat{N}_1 \dots \hat{N}_{k-1}R \twoheadrightarrow_\beta a\hat{N}_1 \dots \hat{N}_{k-1}R(Rv_1 \dots v_nw_{m+1} \dots w_h\Delta),$$

$$Rv_1 \dots v_nw_{m+1} \dots w_h\Delta \twoheadrightarrow_\beta x\hat{N'}_1 \dots \hat{N'}_mw_{m+1} \dots w_{h-1}(xw_h\Delta S(\Delta\Delta)),$$

where S is a suitable term.

Notice that in all cases X and Y are normal forms. □

The previous lemma suggests us to define a set of normal forms (the positive normal forms) which includes the set PNF of persistently normalising normal forms we want to characterise, as proved in Theorem 4.4.15.

Definition 4.4.14 (Positive normal forms) *A normal form M is a positive normal form ($M \in \mathsf{NF}^+$) if and only if*

i) the head variable of M is free (or equivalently negative)

ii) there are no adjacent occurrences of positive variables in M.

.

Theorem 4.4.15 *The persistently normalising normal forms are positive normal forms, i.e.* $\mathsf{PNF} \subseteq \mathsf{NF}^+$.

PROOF. We show that if M does not belong to NF^+ then M does not belong to PNF. The easier case is when M does not belong to NF^+ since the head variable of M is bound. Let $M \equiv \lambda\vec{x}y\vec{z}.y\vec{N}$. Then clearly $M\vec{x}(\lambda\vec{t}u.u\Delta\vec{t})\vec{z}\Delta \twoheadrightarrow_\beta \Delta\Delta\vec{N}'$ where $\vec{t}$ has the same length as $\vec{N}$ and $\vec{N}' \equiv \vec{N}[y := \lambda\vec{t}u.u\Delta\vec{t}]$.

Otherwise there must be adjacent occurrences of positive variables z, t in M. Let $M \equiv \lambda\vec{x}.y\vec{N}$. From Definitions 4.4.8 and 4.4.11 we get that all positive variables of M are negative variables of $y\vec{N}$ and their replacement paths in $y\vec{N}$ start with one variable belonging to $\vec{x}$. Let $x_i\alpha, x_j\beta$ be the replacement paths of z, t in $y\vec{N}$ (possibly $i = j$). By Lemma 4.4.13 there are normal forms X_i, X_j such that $y\vec{N}[x_i := X_i, x_j := X_j]$ (or, when $i = j$, one normal form X_i such that $y\vec{N}[x_i := X_i]$) does not reduce to normal form. Now by choosing

$$X_l = \begin{cases} X_i & \text{if } l = i \\ X_j & \text{if } l = j \\ x_l & \text{otherwise} \end{cases}$$

we get that $M\vec{X}$ does not reduce to normal form and this implies $M \notin \mathsf{PNF}$. $\square$

Example 4.4.16 *If*

$$M \equiv x(\lambda y.a(y(\lambda z.b(x(\lambda t.c(\underline{zt})))))),$$

then the underlined occurrences of the variables z, t are adjacent in M and their replacement paths in M are respectively $x\langle 1,1\rangle\langle 1,1\rangle$ and $x\langle 1,1\rangle$. Following the proof of Lemma 4.4.13 we can consider the normal form M' of $M[x := X']$ where $X' \equiv \lambda u.xu(uy)$. We get

$$M' \equiv x(\lambda y.a(y(\lambda z.b(x(\lambda t.c(zt))(c(zy))))))(a(y(\lambda z.b(x(\lambda t.c(zt))(c(\underline{zy})))))).$$

The underlined occurrences of the variables z, y are adjacent in M' with replacement paths respectively $y\langle 1,1\rangle$ and y. Now the normal form of $M'[y := Y']$, where $Y' \equiv$

$\lambda v.y(yv(vz))$, is

$$\begin{aligned} M'' \;\equiv\;& x(\lambda y.a(y(\lambda z.b(x(\lambda t.c(zt))(c(zy))))))\\ & (a(y(y(\lambda z.b(x(\lambda t.c(zt))(c(z(\lambda v.y(yv(vz))))))))\\ & (b(y(\lambda t.c(zt))(c(\underline{z}(\lambda v.\underline{y}(yv(vz)))))))))). \end{aligned}$$

In M'' the underlined occurrences of the variable z, y are adjacent with replacement paths z, y. So if we choose $Z \equiv \lambda w.aw(wv\Delta)$, $Y'' \equiv \lambda r_1 r_2.br_1r_2(r_2r_2)$ we get $M''[z := Z, y := Y'']$ which does not have a normal form. Now following the proof we build $Y \equiv \lambda s_1 s_2.y(ys_1s_2(s_1Z)(s_2s_2))$, and also $M''[y := Y]$ does not have a normal form. Lastly replacing Y to y in X' we obtain the term $X \equiv \lambda u.xu(uY)$ and one can check that the application of $\lambda x.M$ to X does not have a normal form. The crucial steps are:

$(\lambda x.M)X \twoheadrightarrow_\beta XN_1 \twoheadrightarrow_\beta xN_1(N_1Y)$ *where* $N_1 \equiv \lambda y.a(y(\lambda z.b(X(\lambda t.c(zt)))))$

$N_1Y \twoheadrightarrow_\beta a(YN_2)$ *where* $N_2 \equiv \lambda z.b(X(\lambda t.c(zt)))$

$YN_2 \twoheadrightarrow_\beta \lambda s_2.y(yN_2s_2(N_2Z)(s_2s_2))$

$N_2Z \twoheadrightarrow_\beta b(X(\lambda t.c(Zt)))$

$X(\lambda t.c(Zt)) \twoheadrightarrow_\beta xN_4(N_4Y)$ *where* $N_4 \equiv \lambda t.c(Zt)$

$N_4Y \twoheadrightarrow_\beta c(ZY) \twoheadrightarrow_\beta c(aY(Yv\Delta))$

$Yv\Delta \twoheadrightarrow_\beta y(yv\Delta(vZ)(\Delta\Delta))$.

Figure 4.5 shows the Böhm trees of M, M' and M''.

Theorem 4.4.15 suggests to consider a proper subset of the polarised normal forms, i.e. the polarised normal forms not containing adjacent occurrences of positive variables. This can be obtained by the simple move of restricting the hypothesis in rule (+ *app*). We call them strongly polarised normal forms.

Definition 4.4.17 (Strongly polarised normal forms) *The set of* strongly polarised normal forms, $\mathsf{SNF}^{\mathsf{i,j}}$ *is the subset of the set of polarised normal forms,* $\mathsf{NF}^{\mathsf{i,j}}$, *defined as follows:*

$$(+\,appr)\ \frac{\vec{M} \in \mathsf{SNF}^{+,-}}{x^+\vec{M} \in \mathsf{SNF}^{+,+} \cap \mathsf{SNF}^{-,+}} \qquad (-\,app)\ \frac{\vec{M} \in \mathsf{SNF}^{-,+} \cup \mathsf{SNF}^{-,-}}{x^-\vec{M} \in \mathsf{SNF}^{+,-} \cap \mathsf{SNF}^{-,-}}$$

$$(+\,abs)\ \frac{M \in \mathsf{SNF}^{+,\mathsf{j}}}{\lambda x^+.M \in \mathsf{SNF}^{+,\mathsf{j}}} \qquad (-\,abs)\ \frac{M \in \mathsf{SNF}^{-,\mathsf{j}}}{\lambda x^-.M \in \mathsf{SNF}^{-,\mathsf{j}}}$$

Example 4.4.18 *The λ-term $\lambda x^+y^+.a^-(x^+a^-)(\lambda z^-.z^-y^+)$ (see Figure 4.6) is a strongly polarised normal form. Instead, the principal decoration of the λ-term $\lambda v.x(\lambda t.a(t(\lambda uz.b(x(vv)(\lambda v.d(vz))))))$ (see Figure 4.3) is not a strongly polarised normal form since there are adjacent occurrences of the positive variable v^+.*

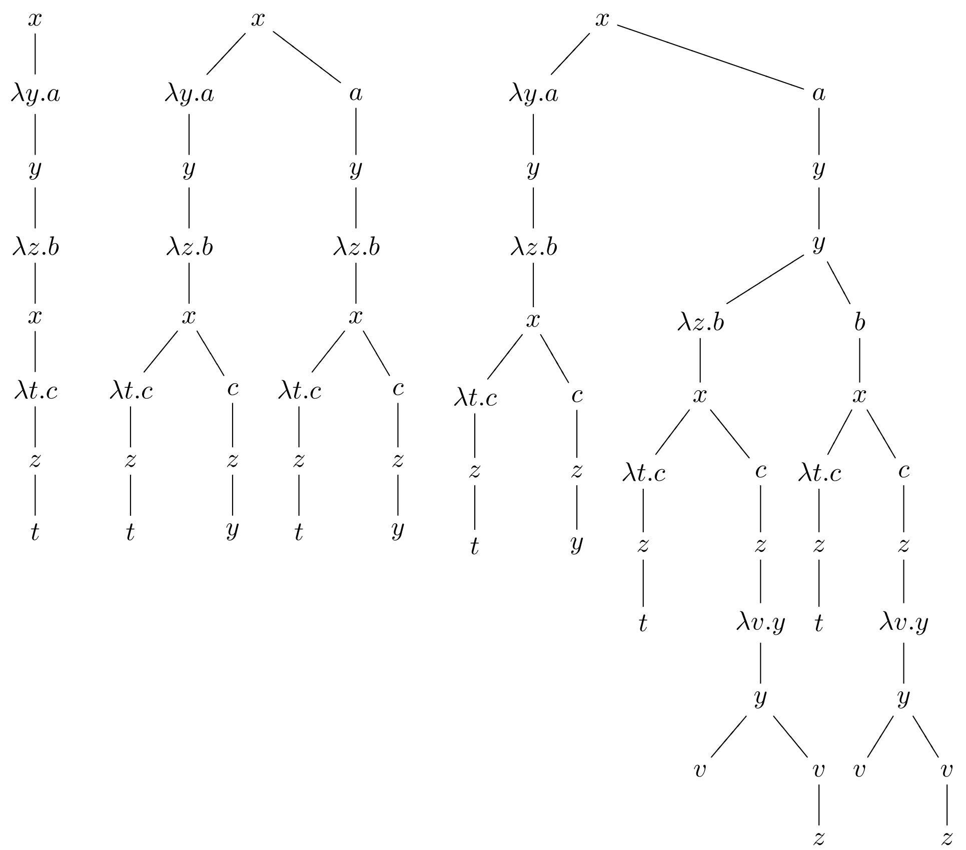

Figure 4.5: Böhm trees of M, M', M'' as defined in Example 4.4.16.

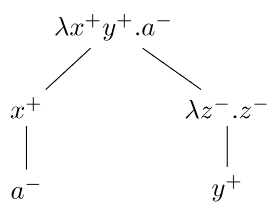

Figure 4.6: Böhm tree of the strongly polarised normal form $\lambda x^+ y^+ .a^-(x^+ a^-)(\lambda z^- .z^- y^+)$.

Rule ($+$ *appr*) says that we can apply a positive variable only to normal forms whose head variable is negative. I.e. in a strongly polarised normal form we cannot have two adjacent occurrences of positive variables.

Remark 4.4.19 *In the Böhm tree of a strongly polarised normal form all sons of a node whose head variable is positive have negative head variables.*

It is clear that not all principal decorations of normal forms are strongly polarised normal forms, take as an example $\lambda x.xx$. But we can easily see that:

Proposition 4.4.20 *The positive normal forms are exactly the normal forms whose principal decorations belong to* $\mathsf{SNF}^{+,-}$.

We can prove that strongly polarised normal forms can be typed starting from arbitrary types not containing Ω (modulo $\sim_\nabla$) for positive variables, whenever Σ^∇ is an arrow type theory. Moreover under the same hypothesis the strongly polarised normal forms which are principal decorations of positive normal forms can be typed with an arbitrary type not containing Ω modulo $\sim_\nabla$ (Theorem 4.4.23). First we need to show a property of arrow type theories (Lemma 4.4.22).

We associate to each type the minimum number of external arrows.

Definition 4.4.21 *Let* Σ^∇ *be a type theory. The mapping* $|\ |: \mathbf{T}^\nabla \to I\!N$ *is defined inductively on types as follows:*

$$\begin{array}{lcll} |A| & = & 0 & \textit{if } A \in \mathbf{C}^\nabla; \\ |A \to B| & = & 1 + |B|; & \\ |A \cap B| & = & \min\{|A|, |B|\}. & \end{array}$$

Lemma 4.4.22 *Let* Σ^∇ *be an arrow type theory. For each* $A \in \mathbf{T}^\nabla$ *and for each integer* n:

i. there is $A' \in \mathbf{T}^\nabla$ *such that* $A' \sim_\nabla A$ *and* $|A'| \geq n$;

ii. there is $A' \in \mathbf{T}^\nabla$ *such that* $A' \sim_\nabla A$ *and* $A' \equiv \bigcap_{i\in I}(\vec{B_i} \to C_i)$ *where* $\vec{B_i}$ *has length* n *for all* $i \in I$.

Proof.

(i) It is enough to show that for each $A \in \mathbf{T}^\nabla$ we can find $A' \in \mathbf{T}^\nabla$ such that $A' \sim_\nabla A$ and $|A'| > |A|$. The proof is by induction on A. The case $A \in \mathbf{C}^\nabla$ follows immediately from the definition of arrow type theory. If $A \equiv B \to C$ then by induction there is $C' \in \mathbf{T}^\nabla$ such that $C' \sim_\nabla C$ and $|C'| > |C|$. We can choose $A' \equiv B \to C'$. If $A \equiv B \cap C$ then by induction there are $B', C' \in \mathbf{T}^\nabla$ such that $B' \sim_\nabla B$, $C' \sim_\nabla C$ and $|B'| > |B|$, $|C'| > |C|$. We can choose $A' \equiv B' \cap C'$.

(ii) By (i) it suffices to show that for each $A \in \mathbf{T}^\nabla$ with $|A| \geq n$ there is $A' \in \mathbf{T}^\nabla$ such that $A' \sim_\nabla A$ and $A' \equiv \bigcap_{i \in I}(\vec{B}_i \to C_i)$ where $\vec{B}_i$ has length n for all $i \in I$. The proof is by induction on A. The case $A \in \mathbf{C}^\nabla$ is trivial since $n = 0$. If $A \equiv B \to C$ then $|C| \geq n-1$. By induction there is $C' \in \mathbf{T}^\nabla$ such that $C' \sim_\nabla C$ and $C' \equiv \bigcap_{i \in I}(\vec{D}_i \to E_i)$ where $\vec{D}_i$ has length $n-1$ for all $i \in I$. We can choose $A' \equiv \bigcap_{i \in I}(B \to \vec{D}_i \to E_i)$, since $A' \sim_\nabla A$ by rules ($\to$-$\cap$) and (η). The case $A \equiv B \cap C$ is easy by induction.

$\square$

Theorem 4.4.23 *Let Σ^∇ be an arrow type theory. Let $M \in \mathsf{SNF}^{\mathsf{i},\mathsf{j}}$ and let $\vec{x^+}$ and $\vec{y^-}$ be the positive and negative variables which occur free in M:*

i) if $\mathsf{i} = +$ and $\mathsf{j} = -$ then for all types $\vec{A}$ with $\Omega \notin_\nabla \vec{A}$ and for all types A with $\Omega \notin_\nabla A$ there exist types $\vec{B}$ with $\Omega \notin_\nabla \vec{B}$ such that $\vec{x^+} : \vec{A}, \vec{y^-} : \vec{B} \vdash^\nabla M : A$.

ii) otherwise for all types $\vec{A}$ with $\Omega \notin_\nabla \vec{A}$ there exist types $\vec{B}$ with $\Omega \notin_\nabla \vec{B}$ and a type A with $\Omega \notin_\nabla A$ such that $\vec{x^+} : \vec{A}, \vec{y^-} : \vec{B} \vdash^\nabla M : A$.

Proof. We prove (i) and (ii) simultaneously by induction on the structure of strongly polarised normal forms. We convene that all considered types do not contain occurrences of Ω modulo $\sim_\nabla$. By x^+ we denote an arbitrary element of $\vec{x}$. Similarly for y^-.

(i) If $M \in \mathsf{SNF}^{+,-}$ then M is of the shape $\lambda \vec{z^+}.y^- \vec{N}$ where $\vec{N} \in \mathsf{SNF}^{-,+} \cup \mathsf{SNF}^{-,-}$. Since Σ^∇ is an arrow type theory, then by Lemma 4.4.22(ii) each type is equivalent to an intersection of arrow types, each one of the shape $\vec{C} \to D$ where the length of $\vec{C}$ is an arbitrary integer. So it suffices to prove that M has all types of the shape $\vec{C} \to D$, where $\vec{C}$ has the length of $\vec{z}$. By the induction hypothesis (ii) there are types $\vec{B}$ and $\vec{E}$ such that for all types $\vec{A}$ and $\vec{C}$ we have: $\vec{x^+} : \vec{A}, \vec{z^+} : \vec{C}, \vec{y^-} : \vec{B} \vdash^\nabla \vec{N} : \vec{E}$. Now let Γ be the ∇-basis obtained by adding the premise $y^- : \vec{E} \to D$ to $\vec{x^+} : \vec{A}, \vec{z^+} : \vec{C}, \vec{y^-} : \vec{B}$. We get $\Gamma \vdash^\nabla y^- \vec{N} : D$ and we can conclude using rule ($\to$ I).

(ii) If $M \in \mathsf{SNF}^{-,-}$ then M is of the shape $\lambda \vec{z^-}.t^- \vec{N}$ where $\vec{N} \in \mathsf{SNF}^{-,+} \cup \mathsf{SNF}^{-,-}$ and $t^- \in \vec{y^-} \cup \vec{z^-}$. By the induction hypothesis (ii) there are types $\vec{B}, \vec{C}$ and $\vec{E}$ such that for all types $\vec{A}$ we get: $\vec{x^+} : \vec{A}, \vec{z^-} : \vec{C}, \vec{y^-} : \vec{B} \vdash^\nabla \vec{N} : \vec{E}$. Now let Γ be the ∇-basis obtained by adding the premise $t^- : \vec{E} \to D$, where D is arbitrary, to $\vec{x^+} : \vec{A}, \vec{z^-} : \vec{C}, \vec{y^-} : \vec{B}$. We get $\Gamma \vdash^\nabla t^- \vec{N} : D$ and we can conclude using rule ($\to$ I).
If $M \in \mathsf{SNF}^{-,+}$ then M is of the shape $\lambda \vec{z^-}.x^+ \vec{N}$ where $\vec{N} \in \mathsf{SNF}^{+,-}$. Let A' be the type of the variable x^+ and n the length of $\vec{N}$. By Lemma 4.4.22(ii) there is a type $\vec{E} \to D$ such that $\vec{E}$ has length $\geq n$ and $A' \leq_\nabla \vec{E} \to D$. By the induction

hypothesis (i) there are types $\vec{B}$ and $\vec{C}$ such that for all types $\vec{A}$ and $\vec{E}$ we have: $\vec{x^+} : \vec{A}, \vec{z^-} : \vec{C}, \vec{y^-} : \vec{B} \vdash^{\nabla} \vec{N} : \vec{E}$. We get $\vec{x^+} : \vec{A}, \vec{z^-} : \vec{C}, \vec{y^-} : \vec{B} \vdash^{\nabla} x^+\vec{N} : D$ and we can conclude using rule $(\to \mathrm{I})$.

If $M \in \mathsf{SNF}^{+,+}$ then M is of the shape $\lambda\vec{z^+}.t^+\vec{N}$ where $\vec{N} \in \mathsf{SNF}^{+,-}$ and $t^+ \in \vec{x^+} \cup \vec{z^+}$. If $t^+ \in \vec{x^+}$ the proof goes as in previous case. Otherwise the proof is similar, since we can assume $t^+ : \vec{E} \to D$, where $\vec{E}$ has the length of $\vec{N}$, for arbitrary types $\vec{E}, D$ and conclude as in previous case.

$\square$

Proof. Proof of Theorem 4.2.2(2.i)($\Rightarrow$). The theory of polarised normal forms has been introduced to get this result. If $M \in \mathsf{PN}$ then by definition its normal form $M' \in \mathsf{PNF}$. By Theorem 4.4.15 $M' \in \mathsf{NF}^+$, so M' has all types not containing Ω modulo $\sim_{\nabla}$(from ∇-bases not containing Ω modulo $\sim_{\nabla}$) in an arbitrary arrow type theory by Proposition 4.4.20 and Theorem 4.4.23. We can conclude that also M has the same types in an arbitrary arrow type theory by Theorem 2.2.4(iv). $\square$

Remark 4.4.24 *By Theorems 4.2.2(2.i)($\Leftarrow$), 4.4.23, and Proposition 4.4.20 we get* $\mathsf{NF}^+ \subseteq \mathsf{PNF}$. *Therefore from Theorem 4.4.15 we can conclude that the persistently normalising normal forms are exactly the positive normal forms, i.e.* $\mathsf{PNF} = \mathsf{NF}^+$.

Conclusions

Intersection type theories are necessary to look for Approximation theorems because types have to be invariant under β-conversion. None of the system of the λ-cube, including the simply typed λ-calculus, are closed under β-expansion. Moreover intersection type assignment systems without Ω don't preserve β-expansion, see [Pot80, HL93].

It is interesting to remark that the proof method of Section 3.2 does not easily extend to strict intersection types as defined in [Bak92]. Actually for strict intersection types Lemma 3.2.3 fails, since for example we can derive $x{:}\psi\to\psi, z{:}\psi\cap\phi \vdash xz{:}\psi$ but we cannot derive $x{:}\phi\to\phi \vdash x{:}\psi\cap\phi\to\phi$.

It would be interesting to develop in full the natural semantics of intersection types in Kripke applicative structures and more generally in pre-sheaf models.

In this thesis we have only considered Approximation Theorems for ordinary λ-calculus. It would be worthwhile exploring the techniques developed in this thesis for establishing corresponding Approximation Theorems for restricted λ-calculi, such as Plotkin's λ_v-calculus [Plo75].

Two natural questions, at least, lurk behind Chapter 4: "can we characterise in some significant way the class of evaluation properties which we can characterise using intersection types?" and "is there a method for going from a logical specification of a property to the appropriate intersection type theory?".

Regarding the first question, we have seen that the properties have to be closed, at least, under some form of β-expansion. But clearly this is not the whole story. Probably the answer to this question is linked to some very important open problems in the theory of the denotational semantics of untyped λ-calculus, like the existence of a denotational model whose theory is precisely $\lambda\beta$. As far as the latter question is concerned, we really have no idea. It seems that we are still missing something in our understanding of intersection types.

Of course there are some partial answers. For instance by looking at what happens in particular filter models, one can draw some inspiration and sometimes even provide some interesting characterisations. In this thesis we proved that closable sets can be characterised in two different ways: using the Approximation Theorem only for the model $\mathcal{P}a$ (Theorem 3.3.5) and using stable sets in the general case (Theorem 4.2.2.3). Instead, for terms which *reduce to terms of the* λ-**I**-*calculus*, we only found a proof based on the Approximation Theorem (Theorem 3.3.8) for the model $\mathcal{HR}$. We do not know how to generalize the result by means of stable sets, the main difficulty being that of finding a suitable interpretation for the type ω. These characterisations however appear quite *accidental*. And we feel that we lack yet a general theory which could allow us to streamline the approach. Given the model we can start to guess.

And when we are successful, we can achieve generality only artificially, by considering all those type theories which extend the theory of the filter model in question.

For one thing this method of drawing inspiration from filter models is interesting, in that it provides some very interesting conjectures. Perhaps the best example concerns *persistently strongly normalising* terms. These are those strongly normalising terms M, such that for all vectors $\vec{N}$ of strongly normalising terms, $M\vec{N}$ is still strongly normalising. Consider the filter model introduced in [HL99], generated by the type theory obtained by pruning the type theory $\Sigma^{\mathcal{CDZ}}$ of all types including Ω, i.e. generated by the theory $\Sigma^{\mathcal{HL}} = \Sigma(\{\varphi, \omega\}, \mathcal{CDV} \cup \{(\omega\varphi), (\varphi{\to}\omega), (\omega{\to}\varphi)\})$. The natural conjecture is then, in analogy to what happens for persistently normalising terms, "are the terms typable with ω in $\lambda\cap^{\mathcal{HL}}_{\mathcal{B}}$, for the $\mathcal{HL}$-basis where all variables have type ω precisely the *persistently strongly normalising* ones?". Completeness is clear, but to show soundness some independent syntactical characterisation of that class of terms appears necessary. The set of persistently strongly normalising terms does not include $\mathsf{PN} \cap \mathsf{SN}$. A counter example is $M \equiv \lambda x.a((\lambda y.b)(xx))$ since $M(\lambda z.zz) \notin \mathsf{SN}$. This conjecture still resists proof.

The results and the techniques of Chapter 4 have been widely used and developed in [DCG03], which mainly focus on the construction of λ-models characterising computational properties of terms.

Finally, I would like to mention some related papers I developed during my Ph.D. studies. The main focus of these papers is on the correspondence between Intersection types and Routley and Meyer's minimal relevant logic $\mathbf{B}_+$ [RM72], pointed out by Meyer and by Venneri [Ven94]. The first question which I investigated is: "does the whole $\mathbf{B}_+$ induce a λ-model?". The answer is negative, but interestingly becomes positive either by restricting to Harrop filters [DCMM03] or by considering the call-by-value λ-calculus [BDCLM00]. The paper [DCFGM02] comments that $\mathbf{B}_+$ with the semantic subtyping of [FCB01] gives a simpler formulation of the subtyping relation. Lastly, in the pure logic framework the paper [MMB01] shows that the addition of classical Boolean negation to $\mathbf{B}_+$ can be recast in terms of a classical first-order metalogic.

Bibliography

[AB91] F. Alessi and F. Barbanera. Strong conjunction and intersection types. In *Mathematical Foundation of Computer Science 1991 (Kazimierz Dolny)*, volume 520 of *Lecture Notes in Computer Science*, pages 64–73. Springer, 1991.

[Abr91] S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51(1-2):1–77, 1991.

[AC98] R. M. Amadio and P.-L. Curien. *Domains and lambda-calculi.* Cambridge University Press, Cambridge, 1998.

[ADCH00] F. Alessi, M. Dezani-Ciancaglini, and F. Honsell. A complete characterization of the complete intersection-type theories. In *ITRS'00 Workshop (Geneva, 2000)*, volume 8 of *Proceedings in Informatics*, pages 287–301. Carleton-Scientific, 2000.

[Ale91] F. Alessi. *Strutture di tipi, teoria dei domini e modelli del lambda calcolo.* PhD thesis, Torino University, 1991.

[AO93] S. Abramsky and C.-H. L. Ong. Full abstraction in the lazy lambda calculus. *Information and Computation*, 105(2):159–267, 1993.

[Bak92] S. van Bakel. Complete restrictions of the intersection type discipline. *Theoretical Computer Science*, 102(1):135–163, 1992.

[Bak93] S. van Bakel. Principal type schemes for the strict type assignment system. *Journal of Logic and Computation*, 3(6):643–670, 1993.

[Bar] H.P. Barendregt. *Typed λ-calculus and applications.* North-Holland. To appear.

[Bar84] H.P. Barendregt. *The Lambda Calculus: its Syntax and Semantics.* North-Holland, revised edition, 1984.

[Bar92] H.P. Barendregt. Lambda calculi with types. In S. Abramsky, Dov M. Gabbay, and Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 117–309. Oxford University Press, 1992.

[BCDC83] H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *The Journal of Symbolic Logic*, 48(4):931–940, 1983.

[BDC75] C. Böhm and M. Dezani-Ciancaglini. λ-terms as total or partial functions on normal forms. In *Lambda-Calculus and Computer Science Theory (Rome, 1975)*, volume 37 of *Lecture Notes in Computer Science*, pages 96–121. Springer, 1975.

[BDCL95] F. Barbanera, M. Dezani-Ciancaglini, and U. de' Liguoro. Intersection and union types: syntax and semantics. *Information and Computation*, 119(2):202–230, 1995.

[BDCLM00] S. van Bakel, D. Dezani-Ciancaglini, U. de' Liguoro, and Y. Motohama. The minimal relevant logic and the call-by-value lambda calculus. Technical Report TR-ARP-05-2000, ANU, 2000.

[Ber00] C. Berline. From computation to foundations via functions and applications: The λ-calculus and its webbed models. *Theoretical Computer Science*, 249:81–161, 2000.

[BG00] H. Barendregt and S. Ghilezan. Lambda terms for natural deduction, sequent calculus and cut-elimination. *Journal of Functional Programming*, 10:121–134, 2000.

[BM94] F. Barbanera and S. Martini. Proof-functional connectives and realizability. *Archive for Mathematical Logic*, 33(3):189–211, 1994.

[CDC80] M. Coppo and M. Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame Journal of Formal Logic*, 21(4):685–693, 1980.

[CDCV80] M. Coppo, M. Dezani-Ciancaglini, and B. Venneri. Principal type schemes and λ-calculus semantics. In *To H. B. Curry: essays on combinatory logic, lambda calculus and formalism*, pages 535–560. Academic Press, 1980.

[CDCV81] M. Coppo, M. Dezani-Ciancaglini, and B. Venneri. Functional characters of solvable terms. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 27(1):45–58, 1981.

[CDCZ87] M. Coppo, M. Dezani-Ciancaglini, and M. Zacchi. Type theories, normal forms, and D_∞-lambda-models. *Information and Computation*, 72(2):85–116, 1987.

[CF58] H.B. Curry and R. Feys. *Combinatory Logic*, volume I. North-Holland, 1958.

[CHDCL84] M. Coppo, F. Honsell, M. Dezani-Ciancaglini, and G. Longo. Extended type structures and filter lambda models. In *Logic colloquium '82 (Florence, 1982)*, pages 241–262. North-Holland, 1984.

[CHS72] H.B. Curry, J.R. Hindley, and J.P. Seldin. *Combinatory Logic*, volume II. North-Holland, 1972.

[Cur34] H.B. Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Science of the USA*, 20:584–590, 1934.

[Cur36] H.B. Curry. First properties of functionality in combinatory logic. *Toholku Mathematical Journal*, 41:371–401, 1935/36.

[DCFGM02] M. Dezani-Ciancaglini, A. Frisch, E. Giovannetti, and Y. Motohama. The relevance of semantic sybtyping. *Electric Notes in Theoretical Computer Science*, 70(1), 2002. URL: `http://www.elsevier.nl/locate/entcs/volume70.html`.

[DCG03] M. Dezani-Ciancaglini and S. Ghilezan. Two behavioural lambda models. In *TYPES 2002 (Berg en Dal)*, volume 2246 of *Lecture Notes in Computer Science*, pages 127–147. Springer, 2003.

[DCHM] M. Dezani-Ciancaglini, F. Honsell, and Y. Motohama. Compositional characterization of *lambda*-terms using intersection types. *Theoretical Computer Science*. To appear.

[DCHM00] M. Dezani-Ciancaglini, F. Honsell, and Y. Motohama. Compositional characterizations of λ-terms using intersection types (extended abstract). In *Mathematical Foundations of Computer Science 2000 (Bratislava)*, volume 1893 of *Lecture Notes in Computer Science*, pages 304–314. Springer, 2000.

[DCHM01] M. Dezani-Ciancaglini, F. Honsell, and Y. Motohama. Approximation theorems for intersection type systems. *Journal of Logic and Computation*, 11(3):395–417, 2001.

[DCMM03] M. Dezani-Ciancaglini, R. K. Meyer, and Y. Motohama. The semantics of entailment omega. *Notre Dame Journal of Formal Logic*, 43(3):129–145, 2003.

[DGH93] P. Di Gianantonio and F. Honsell. An abstract notion of application. In *Typed lambda calculi and applications (Utrecht, 1993)*, volume 664 of *Lecture Notes in Computer Science*, pages 124–138. Springer, 1993.

[EHRDR92] L. Egidi, F. Honsell, and S. Ronchi Della Rocca. Operational, denotational and logical descriptions: a case study. *Fundamenta Informaticae*, 16(2):149–169, 1992.

[Eng81] E. Engeler. Algebras and combinators. *Algebra Universalis*, 13(3):389–392, 1981.

[FCB01] A. Frisch, G. Castagna, and V. Benzaken. Semantic subtyping. In *17th IEEE Symposium on Logic in Computer Science (Copenhagen, 2001)*. IEEE Computer Society Press, 2001.

[Ghi96] S. Ghilezan. Strong normalization and typability with intersection types. *Notre Dame Journal of Formal Logic*, 37(1):44–52, 1996.

[GHK$^+$80] G.K. Gierz, K.H. Hoffmann, K. Keimel, J.D. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer, 1980.

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

[Hin82] J. R. Hindley. The simple semantics for Coppo–Dezani–Sallé types. In *International symposium on programming (Turin, 1982)*, volume 137 of *Lecture Notes in Computer Science*, pages 212–226. Springer, 1982.

[HL80] R. Hindley and G. Longo. Lambda-calculus models and extensionality. *Z. Math. Logi Grundlag. Math.*, 26(4):289–310, 1980.

[HL93] F. Honsell and M. Lenisa. Some results on the full abstraction problem for restricted lambda calculi. In *Mathematical foundations of computer science 1993 (Gdańsk)*, volume 711 of *Lecture Notes in Computer Science*, pages 84–104. Springer, 1993.

[HL99] F. Honsell and M. Lenisa. Semantical analysis of perpetual strategies in λ-calculus. *Theoretical Computer Science*, 212(1-2):183–209, 1999.

[How80] W. A. Howard. The formulas-as-types notion of construction. In *To H. B. Curry: essays on combinatory logic, lambda calculus and formalism*, pages 479–490. Academic Press, 1980.

[HRDR92] F. Honsell and S. Ronchi Della Rocca. An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus. *Journal of Computer and System Sciences*, 45(1):49–75, 1992.

[Kri90] J.-L. Krivine. *Lambda-calcul Types et modèles*. Masson, 1990. English translation: Lambda-calculus, types and models, Ellis Horwood, 1993.

[LE85] E. G. K. López-Escobar. Proof functional connectives. In *Methods in Mathematical Logic (Carcas, 1993)*, volume 1130 of *Lecture Notes in Mathematics*, pages 208–221. Springer, 1985.

[Lei86] D. Leivant. Typing and computational properties of lambda expressions. *Theoretical Computer Science*, 44(1):51–68, 1986.

[Lon83] G. Longo. Set-theoretical models of λ-calculus: theories, expansions, isomorphisms. *Annals of Pure and Applied Logic*, 24(2):153–188, 1983.

[Lon87] G. Longo. On Church's formal theory of functions and functionals. Technical Report 9, Department of Computing Science, Pisa University, 1987.

[Mey95] R.K. Meyer. Types and the boolean system $\mathbf{CB}^+$. Technical report, ANU University, Canberra, 1995. ARP Technical Report TR-SRS-5-95.

[Mit96] J. Mitchell. *Foundation for Programmimg Languages*. MIT Press, 1996.

[MMB01] R.K. Meyer, Y. Motohama, and V. Bono. Truth translations of basic relevant logics. The 29th Annual Meeting of the Society for Exact Philosophy (Montreal, 2001), 2001.

[Par76] D. Park. The Y-combinator in Scott's λ-calculus models (revised version). Theory of Computation Report 13, Department of Computer Science, University of Warwick, 1976.

[Pit94] A. M. Pitts. Computational adequacy via 'mixed' inductive definitions. In *Mathematical foundations of Programing Semantics 1993 (New Orleans)*, volume 802 of *Lecture Notes in Computer Science*, pages 72–82. Springer, 1994.

[Plo75] G. D. Plotkin. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1(2):125–159, 1975.

[Plo93] G. D. Plotkin. Set-theoretical and other elementary models of the λ-calculus. *Theoretical Computer Science*, 121(1-2):351–409, 1993.

[Pot80] G. Pottinger. A type assignment for the strongly normalizable λ-terms. In *To H. B. Curry: essays on combinatory logic, lambda calculus and formalism*, pages 561–577. Academic Press, 1980.

[PRDRR99] A. Pravato, S. Ronchi Della Rocca, and L. Roversi. The call-by-value lambda calculus: a semantic investigation. *Mathematical structures in computer science*, 9(5):617–650, 1999.

[RdR82] S. Ronchi della Rocca. Characterization theorems for a filter lambda model. *Information and Control*, 54(3):201–216, 1982.

[RDR98] S. Ronchi Della Rocca. Lecture notes on semantics and types. Internal report, Torino University, 1998.

[RDRV84] S. Ronchi Della Rocca and B. Venneri. Principal type schemes for an extended type theory. *Theoretical Computer Science*, 28(1-2):151–169, 1984.

[RM72] R. Routley and R.K. Meyer. The semantics of entailment –III. *Journal of Philosophical Logic*, 1:192–208, 1972.

[Sco72] D.S. Scott. Continuous lattices. In *Toposes, algebraic geometry and logic (Halifax, 1971)*, volume 274 of *Lecture Notes in Mathematics*, pages 97–136. Springer, 1972.

[Sco75] D.S. Scott. Open problems. In *Lambda Calculus and Computer Science Theory (Rome, 1975)*, volume 37 of *Lecture Notes in Computer Science*, page 369. Springer, 1975.

[Sev96] P. Severi. *Normalisation in lambda calculus and its relation to type inference*. PhD thesis, Eindhoven University of Technology, 1996.

[Ven94] B. Venneri. Intersection types as logical formulae. *Journal of Logic and Computation*, 4(2):109–124, 1994.

[Wad76] C. P. Wadsworth. The relation between computational and denotational properties for Scott's D_∞-models of the lambda-calculus. *SIAM Journal on Computing*, 5(3):488–521, 1976.