Part III

Intersection types $\lambda_{\cap}^{\mathcal{S}}$ 16.10.2006:1032

Comment: LATEX WARNINGS:

- replace in a systematic way LS by ZS (as you did in Chapter 17) can destroy references;
- changing the macro into which was $\cap$ to $\cap\top$ has introduced a lot of $\cap\top\top$

Contents

III Intersection types $\lambda_\cap^{\mathcal{S}}$ 16.10.2006:1026 1

14 An Exemplary System 16.10.2006:1026 9
14.1 The system of type assignment $\lambda_\cap^{\mathrm{BCD}}$ 10
14.2 The filter model 15
14.3 Completeness of type assignment 17

15 The Systems $\lambda_\cap^{\mathcal{T}}$ and $\lambda_{\cap^\top}^{\mathcal{T}}$ 16.10.2006:1026 19
15.1 Type theories 20
15.2 Type assignment 28
15.3 Type structures 32
15.4 Filters 35
15.5 Exercises 16.10.2006:1026 37

16 Basic Properties 16.10.2006:1026 39
16.1 Inversion theorems 39
16.2 Subject reduction and expansion 45
16.3 Exercises 52

17 Type and Lambda Structures 55
17.1 Algebraic lattices and zip structures 57
17.2 Meet-semi lattices and algebraic lattices 63
17.3 Type and zip structures 70
17.4 From zip to lambda structures 77
17.5 Type structures with top for arrow types 90
17.6 Exercises 94

18 Models 16.10.2006:1026 97
18.1 Lambda models 97
18.1.1 Isomorphisms of λ-models 100
18.2 Filter models 101
18.3 Approximation theorems 111
18.4 Exercises 129

19 Applications 16.10.2006:1026 131
19.1 Realizability interpretation of types 131
19.2 Characterizing syntactic properties 136
19.3 D_∞ models as filter models 143

19.4 Other models 160
19.5 Undecidability of inhabitation 165
19.6 Exercises 182

In a nutshell the intersection type systems considered in this Part form a class of type assignment systems for untyped λ-calculus, extending Curry's *basic functionality* with a new type constructor, *intersection*. This simple move makes it possible to express naturally and in a finitary way many *operational and denotational* properties of terms.

Intersection types have been originally introduced as a language for describing and capturing properties of λ-terms, which had escaped all previous typing disciplines. For instance, they were used in order to give the first type theoretic characterization of *strongly normalizable* terms, and later of *(persistently) normalizing terms*.

It was realized early on that intersection types also had a distinctive semantical flavour: they express at a syntactical level the fact that a term belongs to suitable compact open sets in a Scott domain. Building on this intuition, intersection types were used in Barendregt et al. [1983] to introduce filter models and give a proof of the completeness of the natural semantics of simple type assignment systems in applicative structures suggested in Scott [1972].

Since then, intersection types have been used as a powerful tool both for the analysis and the synthesis of λ-models. On the one hand, intersection type disciplines provide finitary inductive definitions of interpretation of λ-terms in models. On the other hand, they are suggestive for the shape the domain model has to have in order to exhibit certain properties.

Intersection types can be viewed also as a restriction of the domain theory in logical form, see Abramsky [1991], to the special case of modeling pure lambda calculus by means of ω-algebraic complete lattices. Many properties of these models can be proved using this paradigm, which goes back to Stone duality.

Chapter 14

An Exemplary System 16.10.2006:1032

There are several systems that assign intersection types to untyped lambda terms. These will be collectively denoted by $\lambda_\cap$. In this section we consider one particular system of this family, $\lambda_\cap^{\text{BCD}}$ in order to outline the concepts and related properties. Definitions and the statement of theorems will be given, but no proofs. These can be found in the next chapters of Part III.

One motivation for the system presented comes from trying to modify the system $\lambda_\to$ in such a way that not only subject reduction, but also subject expansion holds. The problem of subject expansion is the following. Suppose $\vdash_{\lambda_\to} M : A$ and that $M' \twoheadrightarrow_{\beta\eta} M$. Does one have $\vdash_{\lambda_\to} M' : A$? Let us focus on one β-step. So let $M \equiv (\lambda x.P)Q$ be a redex and suppose

$$\vdash_{\lambda_\to} P[x := Q] : A. \tag{1}$$

Do we have $\vdash_{\lambda_\to} (\lambda x.P)Q : A$? It is tempting to reason as follows. By assumption (1) also Q must have a type, say B. Then $(\lambda x.P)$ has a type $B{\to}A$ and therefore $\vdash_{\lambda_\to} (\lambda x.P)Q : A$. The mistake is that in (1) there may be several occurrences of Q, say $Q_1 \equiv Q_2 \equiv \ldots \equiv Q_n$, having as types respectively $B_1, \ldots, B_n$. It may be impossible to find a single type for all the occurrences of Q and this prevents us from finding a type for the redex. For example

$$\begin{array}{rcl} \vdash_{\lambda_\to} (\lambda x.\mathsf{I}(\mathsf{K}x)(\mathsf{I}x)) & : & A{\to}A, \\ \nvdash_{\lambda_\to} (\lambda xy.x(\mathsf{K}y)(xy))\mathsf{I} & : & A{\to}A. \end{array}$$

The system introduced in this chapter with intersection types assigned to untyped lambda terms remedies the situation. The idea is that if the several occurrences of Q have to have different types $B_1, \ldots, B_n$, we give them all of these types:

$$\vdash Q : B_1 \cap \ldots \cap B_n,$$

implying that for all i one has $Q : B_i$. Then we have

$$\begin{array}{rcll} \vdash (\lambda x.P) & : & B_1 \cap \ldots \cap B_n {\to} A & \quad\text{and} \\ \vdash ((\lambda x.P)Q) & : & A. & \end{array}$$

There is, however, a second problem. In the $\lambda\mathsf{K}$-calculus, with its terms $\lambda x.P$ such that $x \notin \text{FV}(P)$ there is the extra problem that Q may not be typable at all, as it may not occur in $P[x := Q]$! This is remedied by allowing $B_1 \cap \ldots \cap B_n$ also for $n = 0$ and writing this type as $\top$, to be considered as the universal type, i.e. assigned to all terms. Then in case $x \notin \mathrm{FV}(P)$ one has

$$\begin{array}{rcl} \vdash (\lambda x.P) & : & \top \to A \qquad \text{and} \\ \vdash ((\lambda x.P)Q) & : & A. \end{array}$$

This is the motivation to introduce a $\leq$ relation on types with largest element $\top$ and intersections such that $A \cap B \leq A$, $A \cap B \leq B$ and the extension of the type assignment by the sub-sumption rule $\Gamma \vdash M : A,\ A \leq B \ \Rightarrow\ \Gamma \vdash M : B$. It has as consequence that terms like $\lambda x.xx$ get as type $((A \to B) \cap A) \to B$, while $(\lambda x.xx)(\lambda x.xx)$ only gets $\top$ as type. Also we have subject conversion

$$\Gamma \vdash M : A \ \&\ M =_\beta N \ \Rightarrow\ \Gamma \vdash N : A.$$

This has as consequence that one can create a lambda model in which the meaning of a closed term consists of the collection of types it gets. In this way new lambda models will be obtained and new ways to study classical models as well.

The type assignement system $\lambda_\cap^{\mathrm{BCD}}$ will be introduced in Section 14.1 and the correspondig filter model in 14.2.

14.1. The system of type assignment $\lambda_\cap^{\mathrm{BCD}}$

A typical member of the family of intersection type assignment systems is $\lambda_\cap^{\mathrm{BCD}}$. This system is introduced in Barendregt et al. [1983] as an extension of the initial system in Coppo and Dezani-Ciancaglini [1980].

14.1.1. Definition. Let $\mathbb{A}$ be a set of type atoms.

(i) The *intersection type language* over $\mathbb{A}$, denoted by $\mathbb{T} = \mathbb{T}_\cap^{\mathbb{A}}$ is defined by the following abstract syntax.

$$\mathbb{T} = \mathbb{A} \mid \mathbb{T} \to \mathbb{T} \mid \mathbb{T} \cap \mathbb{T}$$

(ii) Write

$$\begin{array}{rcl} \mathbb{A}_\infty & = & \{\psi_0, \psi_1, \psi_2, \ldots\} \\ \mathbb{A}_\infty^\top & = & \mathbb{A}_\infty \cup \{\top\}, \end{array}$$

where the type atom $\top \notin \mathbb{A}_\infty$ is considered as a constant.

Notation. (i) A, B, C, D, E range over arbitrary types. When writing intersection types we shall use the following convention: the constructor $\cap$ takes precedence over the constructor $\to$ and it associates to the right. For example

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

(ii) $\alpha, \beta, \ldots$ range over $\mathbb{A}$.

14.1.2. Remark. In Part III the set of syntactic types will be formed as above; for many of these systems the set $\mathbb{A}$ will be finite. In this Chapter, however, we take $\mathbb{A} = \mathbb{A}_\infty^\top$.

The following deductive system has as intention to introduce an appropriate pre-order on $\mathbb{T}$, compatible with the operator $\to$, such that $A \cap B$ is a greatest lower bound of A and B, for each A, B.

14.1.3. Definition (Intersection type preorder). On $\mathbb{T} = \mathbb{T}_\cap^{\mathbb{A}_\infty^\top}$ a binary relation $\leq$ '*is subtype of*' is defined by the following axioms and rules.

(refl)	$A \leq A$
(incl$_L$)	$A \cap B \leq A$
(incl$_R$)	$A \cap B \leq B$
(glb)	$\dfrac{C \leq A \quad C \leq B}{C \leq A \cap B}$
(trans)	$\dfrac{A \leq B \quad B \leq C}{A \leq C}$
$(\top)$	$A \leq \top$
$(\top\to)$	$\top \leq \top\to\top$
$(\to\cap)$	$(A\to B) \cap (A\to C) \leq A\to(B \cap C)$
$(\to)$	$\dfrac{A' \leq A \quad B \leq B'}{(A\to B) \leq (A'\to B')}$

14.1.4. Definition. The intersection type theory BCD is the set of all judgements $A \leq B$ derivable from the axioms and rules in Definition 14.1.3. For $(A \leq B) \in \mathrm{BCD}$ we write $A \leq_{\mathrm{BCD}} B$ or $\vdash_{\mathrm{BCD}} A \leq B$ (or often just $A \leq B$).

14.1.5. Remark. All systems in Part III have the first five axioms and rules of Definition 14.1.3. They differ in the extra axioms and rules and the set of constants.

14.1.6. Definition. Write $A {=_{\mathrm{BCD}}} B$ (or $A = B$) for $A \leq_{\mathrm{BCD}} B$ & $B \leq_{\mathrm{BCD}} A$. In BCD we usually work with $\mathbb{T}$ modulo $=_{\mathrm{BCD}}$. By rule $(\to)$ one has

$$A = A' \ \& \ B = B' \ \Rightarrow \ (A\to B) = (A'\to B').$$

Moreover, $A \cap B$ becomes *the* glb of A, B.

14.1.7. Definition. (i) A *basis* is a finite set of statements of the shape $x{:}B$, where $B \in \mathbb{T}$, with all variables distinct.

(ii) The type assignment system $\lambda_\cap^{\mathrm{BCD}}$ for deriving statements of the form $\Gamma \vdash M : A$ with Γ a basis, $M \in \Lambda$ (the set of untyped lambda terms) and $A \in \mathbb{T}$

is defined by the following axioms and rules.

(Ax)	$\Gamma \vdash x{:}A$	if $(x{:}A) \in \Gamma$
$(\to\mathrm{I})$	$\dfrac{\Gamma,\ x{:}A \vdash M : B}{\Gamma \vdash (\lambda x.M) : (A\to B)}$	
$(\to\mathrm{E})$	$\dfrac{\Gamma \vdash M : (A \to B) \quad \Gamma \vdash N : A}{\Gamma \vdash (MN) : B}$	
$(\cap\mathrm{I})$	$\dfrac{\Gamma \vdash M : A \quad \Gamma \vdash M : B}{\Gamma \vdash M : (A \cap B)}$	
$(\leq)$	$\dfrac{\Gamma \vdash M : A}{\Gamma \vdash M : B}$	if $A \leq_{\mathrm{BCD}} B$
$(\top\text{-universal})$	$\Gamma \vdash M : \top$	

(iii) We say that a term M is *typable* from a given basis Γ, if there is a type $A \in \mathbb{T}$ such that the judgement $\Gamma \vdash M : A$ is derivable in $\lambda_\cap^{\mathrm{BCD}}$. In this case we write $\Gamma \vdash_{\cap\top}^{BCD} M : A$ or just $\Gamma \vdash M : A$, if there is little danger of confusion.

14.1.8. Remark. All systems of type assignment in Part III have the first five axioms and rules of Definition 14.1.7.

In the following Proposition we need the notions of admissible and derived rule. Let us first informally define these notions for the simple logical theory of propositional logic.

14.1.9. Definition. Let $\vdash$ denote provability in propositional logic. Consider the rule

$$\frac{\Gamma \vdash A}{\Gamma \vdash B} \qquad (R)$$

(i) R is called *admissible* if one has

$$\Gamma \vdash A \ \Rightarrow\ \Gamma \vdash B$$

(ii) R is called *derived* if one has

$$\Gamma \vdash A\to B$$

For example we have that

$$\frac{\Gamma \vdash A\to A\to B}{\Gamma \vdash A\to B}$$

is derived. Also that for propositional variables ϑ, ϱ

$$\frac{\vdash \vartheta}{\vdash \varrho}$$

is admissible, simply because $\vdash \vartheta$ does not hold, but not derived. A derived rule is always admissible and the example shows that the converse does not hold. If

$$\frac{\Gamma \vdash A}{\Gamma \vdash B}$$

is a derived rule, then for all $\Gamma' \supseteq \Gamma$ one has that

$$\frac{\Gamma' \vdash A}{\Gamma' \vdash B}$$

is also derived. Hence derived rules are closed under theory extension.

We will only be concerned with admissible and derived rules for theories of type assignment.

14.1.10. PROPOSITION. (i) *Notice that the rules* $(\cap E)$

$$\frac{\Gamma \vdash M : (A \cap B)}{\Gamma \vdash M : A} \qquad \frac{\Gamma \vdash M : (A \cap B)}{\Gamma \vdash M : B}$$

are derived in $\lambda_\cap^{\mathrm{BCD}}$.

(ii) *The following rules are admissible in the intersection type assignment system* $\lambda_\cap^{\mathrm{BCD}}$.

(*weakening*)	$\dfrac{\Gamma \vdash M : A \quad x \notin \Gamma}{\Gamma, x{:}B \vdash M : A}$
(*strengthening*)	$\dfrac{\Gamma, x{:}B \vdash M : A \quad x \notin FV(M)}{\Gamma \vdash M : A}$
(*cut*)	$\dfrac{\Gamma, x{:}B \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash (M[x := N]) : A}$
($\leq$-L)	$\dfrac{\Gamma, x{:}B \vdash M : A \quad C \leq B}{\Gamma, x{:}C \vdash M : A}$
($\to$L)	$\dfrac{\Gamma, y{:}B \vdash M : A \quad \Gamma \vdash N : C \quad x \notin \Gamma}{\Gamma, x{:}(C \to B) \vdash (M[y := xN]) : A}$
($\cap$L)	$\dfrac{\Gamma, x{:}A \vdash M : B}{\Gamma, x{:}(A \cap C) \vdash M : B}$

14.1.11. THEOREM. *In (i) assume* $A \neq \top$. *Then*

$$\begin{array}{rrcl}
\text{(i)} & \Gamma \vdash x : A & \Leftrightarrow & \exists B \in \mathbb{T}.[(x{:}B \in \Gamma \;\&\; B \leq A]. \\
\text{(ii)} & \Gamma \vdash (MN) : A & \Leftrightarrow & \exists B \in \mathbb{T}.[\Gamma \vdash M : (B \to A) \;\&\; \Gamma \vdash N : B]. \\
\text{(iii)} & \Gamma \vdash \lambda x.M : A & \Leftrightarrow & \exists n{>}0 \exists B_1, \dots, B_n, C_1, \dots, C_n \in \mathbb{T} \\
 & & & \forall i \in \{1, \dots, n\}.[\Gamma, x{:}B_i \vdash M : C_i \;\& \\
 & & & (B_1 \to C_1) \cap \dots \cap (B_n \to C_n) \leq A]. \\
\text{(iv)} & \Gamma \vdash \lambda x.M : B \to C & \Leftrightarrow & \Gamma, x{:}B \vdash M : C.
\end{array}$$

14.1.12. Definition. Let R be a notion of reduction. We introduce the following rules:

$$\boxed{\begin{array}{ll}(R\text{-}red) & \dfrac{\Gamma \vdash M : A \quad M \to_R N}{\Gamma \vdash N : A} \\[2ex] (R\text{-}exp) & \dfrac{\Gamma \vdash M : A \quad M \leftarrow_R N}{\Gamma \vdash N : A}\end{array}}$$

14.1.13. Proposition. *The rules (β-red), (β-exp) and (η-red) are admissible in $\lambda_\cap^{\mathrm{BCD}}$. The rule ($\eta$-exp) is not.*

The following result characterizes notions related to normalization in terms of type assignment in the system $\lambda_\cap^{\mathrm{BCD}}$. The notation $\top \notin A$ means that $\top$ does not occur in A.

14.1.14. Theorem. *Let $M \in \Lambda^\varnothing$.*

(i) *M has a head normal form* $\Leftrightarrow$ $\exists A \in \mathbb{T}.[A \neq_{\mathrm{BCD}} \top \;\&\; \vdash M : A]$.

(ii) *M has a normal form* $\Leftrightarrow$ $\exists A \in \mathbb{T}.[\top \notin A \;\&\; \vdash M : A]$.

Let M be a lambda term. For the notion 'approximant of M', see Barendregt [1984]. These are roughly obtained from the Böhm tree $\mathrm{BT}(M)$ of M by cutting of branches and replacing these by a new symbol $\perp$. The set of approximants of M is denoted by $\mathcal{A}(M)$. We have e.g. for the fixed-point combinator Y

$$\mathcal{A}(\mathsf{Y}) = \{\perp\} \cup \{\lambda f.f^n\perp \mid n{>}0\}.$$

Approximants are being typed by letting the typing rules be valid for approximants. For example one has

$$\begin{array}{l}\vdash \perp : \top \\ \vdash \lambda f.f\perp : (\top\to A_1)\to A_1 \\ \vdash \lambda f.f(f\perp) : (\top\to A_1)\cap(A_1\to A_2)\to A_2 \\ \ldots \\ \vdash \lambda f.f^n\perp : (\top\to A_1)\cap(A_1\to A_2)\cap\ldots\cap(A_{n-1}\to A_n)\to A_n \\ \ldots\end{array}$$

The set of types of a term M coincides with the union of the sets of types of its approximants $P \in \mathcal{A}(M)$. This will give an Approximation Theorem for the filter model of next section.

14.1.15. Theorem. $\Gamma \vdash M : A \;\Leftrightarrow\; \exists P \in \mathcal{A}(M).\Gamma \vdash P : A.$

For example since for all n $\lambda f.f^n\perp$ is an approximant of Y we have that all types of the shape $(\top\to A_1)\cap\ldots\cap(A_{n-1}\to A_n)\to A_n$ can be derived for Y.

Finally the question whether an intersection type is inhabited is undecidable.

14.1.16. Theorem. *The set $\{A \in \mathbb{T} \mid \exists M \in \Lambda^\varnothing \vdash M : A\}$ is undecidable.*

14.2. The filter model

14.2.1. Definition. (i) A *complete lattice* $(\mathcal{D}, \sqsubseteq)$ is a partial order which has arbitrary least upper bounds (sup's) (and hence has arbitrary inf's).

(ii) A subset $Z \subseteq \mathcal{D}$ is *directed* if $Z \neq \emptyset$ and

$$\forall x, y \in Z \exists z \in Z. x, y \sqsubseteq z.$$

(iii) An element $c \in \mathcal{D}$ is *compact* (in the literature also called *finite*) if for each directed $Z \subseteq \mathcal{D}$ one has

$$c \sqsubseteq \bigsqcup Z \;\Rightarrow\; \exists z \in Z. c \sqsubseteq z.$$

Let $\mathcal{K}(\mathcal{D})$ denote the set of compact elements of $\mathcal{D}$.

(iv) A complete lattice is ω-*algebraic* if $\mathcal{K}(\mathcal{D})$ is countable, and for each $d \in \mathcal{D}$, the set $\mathcal{K}(d) = \{c \in \mathcal{K}(\mathcal{D}) \mid c \sqsubseteq d\}$ is directed and $d = \bigsqcup \mathcal{K}(d)$.

(v) Let $(\mathcal{D}, \sqsubseteq)$ be an ω-*algebraic* complete lattice. The Scott topology on $\mathcal{D}$ contains as open sets the $U \subseteq \mathcal{D}$ such that

(1) $d \in U \;\&\; d \sqsubseteq e \;\Rightarrow\; e \in U$;

(2) if $Z \subseteq \mathcal{D}$ is directed then $\bigsqcup Z \in U \;\Rightarrow\; \exists z \in Z. z \in U$.

(vi) If $\mathcal{D}, \mathcal{E}$ are ω-algebraic complete lattices, then $[\mathcal{D} \to \mathcal{E}]$ denotes the set of continuous maps from $\mathcal{D}$ to $\mathcal{E}$. This set can be ordered pointwise

$$f \sqsubseteq g \;\Leftrightarrow\; \forall d \in \mathcal{D}. f(d) \sqsubseteq g(d)$$

and $\langle [\mathcal{D} \to \mathcal{E}], \sqsubseteq \rangle$ is again an ω-algebraic lattice.

(vii) The category **ALG** is the category whose objects are the ω-algebraic complete lattices and whose morphisms are the (Scott) continuous functions.

14.2.2. Definition. (i) A *filter* over $\mathbb{T} = \mathbb{T}_{\cap}^{\mathbb{A}_{\infty}^{\top}}$ is a non-empty set $X \subseteq \mathbb{T}$ such that

(1) $A \in X \;\&\; A \leq B \;\Rightarrow\; B \in X$;

(2) $A, B \in X \;\Rightarrow\; (A \cap B) \in X$.

(ii) $\mathcal{F}$ denotes the set of filters over $\mathbb{T}$.

14.2.3. Definition. (i) If $X \subseteq \mathbb{T}$ is non-empty, then the filter *generated by* X, notation $\uparrow X$, is the least filter containing X. Note that

$$\uparrow X = \{A \mid \exists n {\geq} 1 \, \exists B_1 \ldots B_n \in X. B_1 \cap \ldots \cap B_n \leq A\}.$$

(ii) A *principal* filter is of the form $\uparrow\{A\}$ for some $A \in \mathbb{T}$. We shall denote this simply by $\uparrow A$. Note that $\uparrow A = \{B \mid A \leq B\}$.

14.2.4. Proposition. (i) *$\mathcal{F} = \langle \mathcal{F}, \subseteq \rangle$ is an ω-algebraic complete lattice.*

(ii) *$\mathcal{F}$ has as bottom element $\uparrow \top$ and as top element $\mathbb{T}$.*

(iii) *The compact elements of $\mathcal{F}$ are exactly the principal filters.*

14.2.5. Definition. Let $\mathcal{D}$ be an ω-algebraic lattice and let

$$\begin{array}{rcl} F & : & \mathcal{D}\to[\mathcal{D}\to\mathcal{D}] \\ G & : & [\mathcal{D}\to\mathcal{D}]\to\mathcal{D} \end{array}$$

be Scott continuous. $\mathcal{D}$ is called a *reflexive* via F, G if $F \circ G = \mathtt{id}_{[\mathcal{D}\to\mathcal{D}]}$.

A reflexive element of **ALG** is also a λ-model in which the term interpretation is naturally defined as follows (see Barendregt [1984], Section 5.4).

14.2.6. Definition (Interpretation of terms). Let $\mathcal{D}$ be reflexive via F, G.

(i) A *term environment* in $\mathcal{D}$ is a map $\rho : \mathtt{Var}\to\mathcal{D}$.

(ii) If ρ is a term environment and $d \in \mathcal{D}$, then $\rho(x := d)$ is the term environment ρ' defined by

$$\begin{array}{rcll} \rho'(y) & = & \rho(y) & \text{if } y \not\equiv x; \\ \rho'(x) & = & d. & \end{array}$$

(iii) Given a term environment ρ, the interpretation $[\![\ ]\!]_\rho : \Lambda\to\mathcal{D}$ is defined as follows.

$$\begin{array}{rcl} [\![x]\!]^{\mathcal{D}}_\rho & = & \rho(x); \\ [\![MN]\!]^{\mathcal{D}}_\rho & = & F[\![M]\!]^{\mathcal{D}}_\rho[\![N]\!]^{\mathcal{D}}_\rho; \\ [\![\lambda x.M]\!]^{\mathcal{D}}_\rho & = & G(\boldsymbol{\lambda} d \in \mathcal{D}.[\![M]\!]^{\mathcal{D}}_{\rho(x:=d)}). \end{array}$$

(iv) The statement $M = N$, for M, N untyped lambda terms, is *true in* $\mathcal{D}$, notation $\mathcal{D} \models M = N$ iff

$$\forall \rho \in \mathsf{Env}_D.[\![M]\!]^{\mathcal{D}}_\rho = [\![N]\!]^{\mathcal{D}}_\rho.$$

14.2.7. Theorem. *Let $\mathcal{D}$ be reflexive via F, G. Then $\mathcal{D}$ is a λ-model, in particular for all $M, N \in \Lambda$*

$$\mathcal{D} \models (\lambda x.M)N = M[x: = N].$$

14.2.8. Proposition. *Define maps $F : \mathcal{F}\to[\mathcal{F}\to\mathcal{F}]$ and $G : [\mathcal{F}\to\mathcal{F}]\to\mathcal{F}$ by*

$$\begin{array}{rcl} F(X)(Y) & = & \uparrow\{B \mid \exists A \in Y.(A\to B) \in X\} \\ G(f) & = & \uparrow\{A\to B \mid B \in f(\uparrow A)\}. \end{array}$$

Then $\mathcal{F}$ is reflexive via F, G. Therefore $\mathcal{F}$ is a λ-model.

An important property of the λ-model $\mathcal{F}$ is that the meaning of a term is the set of types which are deducible for it.

14.2.9. Theorem. *For all λ-terms M one has*

$$[\![M]\!]^{\mathcal{F}}_\rho \quad = \quad \{A \mid \exists\Gamma \models \rho.\Gamma \vdash M : A\},$$

where $\Gamma \models \rho$ iff for all $(x{:}B) \in \Gamma$ one has $B \in \rho(x)$.

Lastly we notice that all continous functions are representable.

14.2.10. THEOREM.

$$[\mathcal{F}\to\mathcal{F}] = \{f : \mathcal{F}\to\mathcal{F} \mid f \textit{ is representable}\},$$

where $f \in \mathcal{F}\to\mathcal{F}$ *is called* representable *iff for some* $X \in \mathcal{F}$ *one has*

$$\forall Y \in \mathcal{F}.f(Y) = F(X)(Y).$$

14.3. Completeness of type assignment

14.3.1. DEFINITION (Interpretation of types). Let $\mathcal{D}$ be reflexive via F, G and hence a λ-model. For $F(d)(e)$ we also write (as usual) $d \cdot e$.

(i) A *type environment* in $\mathcal{D}$ is a map $\xi : \mathbb{A}_\infty \to \mathcal{P}(\mathcal{D})$.

(ii) For $X, Y \in \mathcal{P}(\mathcal{D})$ define

$$X\to Y = \{d \in \mathcal{D} \mid d \cdot X \subseteq Y\} = \{d \in \mathcal{D} \mid \forall x \in X.d \cdot x \in Y\}.$$

(iii) Given a type environment ξ, the interpretation $[\![\ ]\!]_\xi : \mathbb{T}\to\mathcal{P}(\mathcal{D})$ is defined as follows.

$$\begin{array}{rcll} [\![\top]\!]^{\mathcal{D}}_\xi & = & \mathcal{D}; & \\ [\![\alpha]\!]^{\mathcal{D}}_\xi & = & \xi(\alpha), & \text{for } \alpha \in \mathbb{A}_\infty; \\ [\![A\to B]\!]^{\mathcal{D}}_\xi & = & [\![A]\!]^{\mathcal{D}}_\xi \to [\![B]\!]^{\mathcal{D}}_\xi; & \\ [\![A \cap B]\!]^{\mathcal{D}}_\xi & = & [\![A]\!]^{\mathcal{D}}_\xi \cap [\![B]\!]^{\mathcal{D}}_\xi. & \end{array}$$

14.3.2. DEFINITION (Satisfaction). (i) Given a λ-model $\mathcal{D}$, a term environment ρ and a type environment ξ one defines the following.

$$\begin{array}{rcll} \mathcal{D}, \rho, \xi \models M : A & \Leftrightarrow & [\![M]\!]^{\mathcal{D}}_\rho \in [\![A]\!]^{\mathcal{D}}_\xi. & \\ \mathcal{D}, \rho, \xi \models \Gamma & \Leftrightarrow & \mathcal{D}, \rho, \xi \models x : B, & \text{for all } (x{:}B) \in \Gamma. \end{array}$$

(ii) $\Gamma \models M : A \Leftrightarrow \forall \mathcal{D}, \rho, \xi.[\mathcal{D}, \rho, \xi \models \Gamma \Rightarrow \rho, \xi \models M : A]$.

14.3.3. THEOREM (Soundness).

$$\Gamma \vdash M : A \Rightarrow \Gamma \models M : A.$$

14.3.4. THEOREM (Completeness).

$$\Gamma \models M : A \Rightarrow \Gamma \vdash M : A.$$

The completeness proof is an application of the λ-model $\mathcal{F}$, see Barendregt et al. [1983].

Chapter 15

The Systems $\lambda_\cap^{\mathcal{T}}$ and $\lambda_{\cap^\top}^{\mathcal{T}}$ 16.10.2006:1032

Intersection types are syntactic objects forming a free algebra $\mathbb{T}$, which is generated from a set of atoms $\mathbb{A}$, using the operators $\to$ and $\cap$. Postulating axioms and rules an *intersection type theory* results, which characterizes a preorder $\leq_{\mathcal{T}}$ on $\mathbb{T}$ with $\cap$ as set intersection, giving for two elements a greatest lower bound (glb). The class of these theories is abbreviated[1] as TT.

Taking into account the intuitive meaning of $\to$ as function space constructor one usually requires that the resulting equivalence relation $=_{\mathcal{T}}$ is a congruence. Then we speak of a *compatible* type theory, having a corresponding *type structure*

$$\langle \mathcal{S}, \leq, \cap, \to \rangle = \langle \mathbb{T}/{=_{\mathcal{T}}}, \leq, \cap, \to \rangle.$$

The collection of type structures is denoted by TS. Each type structure can be seen as coming from a compatible type theory and compatible type theories and type structures are basically the same. In the present Part III of this book both these syntactic and semantic aspects will be exploited.

$\mathrm{TT}^\top$ is a subset of TT, the set of *top type theories*, where the set of atoms $\mathbb{A}$ has a top element $\top$. Similarly a top intersection type structure $\mathrm{TS}^\top$ is of the form $\langle \mathcal{S}, \leq, \cap, \to, \top \rangle$.

The various type theories (and type structures) are introduced together in order to give reasonably uniform proofs of their properties as well of those of the corresponding type assignment systems and filter models.

Given a (top) type theory $\mathcal{T}$, one can define a corresponding type assignment system. These type assignment systems will be studied extensively in later chapters. We also introduce so-called *filters*, sets of types closed under intersection $\cap$ and preorder $\leq$. These play an important role in Chapter 17 to establish equivalences of categories and in Chapter 18 to build λ-models.

In Section 15.1 we define the notion of type theory and introduce 13 specific examples, including basic lemmas for these. In Section 15.2 the type assignment systems are defined. In Section 15.3 we discuss intersection type structures and introduce specific categories of lattices and type structures to accommodate these. Finally in Section 15.4 the filters are defined.

[1]Since all type theories in Part III of this book are using the intersection operator, we keep this implicit and often simply speak about *(top) type theories*, leaving 'intersection' implicit.

15.1. Type theories

As in Chapter 14 we will use as syntactic types $\mathbb{T} = \mathbb{T}_\cap^{\mathbb{A}}$ defined by

$$\mathbb{T} = \mathbb{A} \mid \mathbb{T}\to\mathbb{T} \mid \mathbb{T} \cap \mathbb{T}$$

as abstract syntax. This time we will use various sets of atoms $\mathbb{A}$.

15.1.1. Definition. (i) An *intersection type theory over a set of type atoms* $\mathbb{A}$ is a set of judgements $\mathcal{T}$ of the form $A \leq B$ (to be read: A is a subtype of B), with $A, B \in \mathbb{T}_\cap^{\mathbb{A}}$, satisfying the following axioms and rules.

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

This means that e.g. $(A \leq A) \in \mathcal{T}$ and $(A \leq B), (B \leq C) \in \mathcal{T} \Rightarrow (A \leq C) \in \mathcal{T}$, for all A, B, C.

(ii) A *top intersection type theory* is an intersection type theory with an element $\top \in \mathbb{T}$ for which one can derive

$$\boxed{(\top) \quad A \leq \top}$$

(iii) The notion '(top) intersection type theory' will be abbreviated as '(top) type theory', as the 'intersection' part is default.

(iv) TT stands for the set of type theories and $\mathrm{TT}^\top$ for that of top type theories.

(v) If $\mathcal{T} \in \mathrm{TT}^{(\top)}$ over $\mathbb{A}$, then we also write $\mathbb{T}^{\mathcal{T}}$ for $\mathbb{T}_\cap^{\mathbb{A}}$.

In this and the next section $\mathcal{T}$ ranges over elements of $\mathrm{TT}^{(\top)}$. Most of them have some extra axioms or rules, the above set being the minimum requirement. For example the theory BCD over $\mathbb{A} = \mathbb{A}_\infty^\top$, defined in Chapter 14 is a $\mathrm{TT}^\top$ and has the extra axioms $(\top\to)$ and $(\to\cap)$ and rule $(\to)$.

15.1.2. Notation. Let $\mathcal{T} \in \mathrm{TT}$. We write the following.

(i) $A \leq_{\mathcal{T}} B$ or $\vdash_{\mathcal{T}} A \leq B$ for $(A \leq B) \in \mathcal{T}$.

(ii) $A =_{\mathcal{T}} B$ for $A \leq_{\mathcal{T}} B \leq_{\mathcal{T}} A$.

(iii) $A <_{\mathcal{T}} B$ for $A \leq B \ \&\ A \neq_{\mathcal{T}} B$.

(iv) If there is little danger of confusion and $\mathcal{T}$ is clear from the context, then we will write $\leq, =, <$ for respectively $\leq_{\mathcal{T}}, =_{\mathcal{T}}, <_{\mathcal{T}}$.

(v) We write $A \equiv B$ for syntactic identity. E.g. $A \cap B \equiv A \cap B$, but $A \cap B \not\equiv B \cap A$.

15.1.3. Lemma. *For any $\mathcal{T}$ one has $A \cap B =_{\mathcal{T}} B \cap A$.*

Proof. By (incl_L), (incl_R) and (glb). ∎

15.1.4. Definition. $\mathcal{T}$ is called *compatible* iff the following rule holds.

$$\boxed{(\to^{=}) \quad \frac{A = A' \quad B = B'}{(A{\to}B) = (A'{\to}B')}}$$

This means $A =_{\mathcal{T}} A' \ \&\ B =_{\mathcal{T}} B' \ \Rightarrow\ (A{\to}B) =_{\mathcal{T}} (A'{\to}B')$. One way to insure this is to adopt $(\to^{=})$ as rule determining $\mathcal{T}$.

15.1.5. Remarks. (i) Let $\mathcal{T}$ be compatible. Then by Lemma 15.1.3 one has

$$(A \cap B){\to}C = (B \cap A){\to}C.$$

(ii) The rule (glb) implies that the following rule is admissible.

$$\boxed{(\mathrm{mon}) \quad \frac{A \leq A' \quad B \leq B'}{A \cap B \leq A' \cap B'}}$$

A $\mathcal{T} \in \mathrm{TT}$ can be seen as a structure with a pre-order

$$\mathcal{T} = \langle \mathbb{T}, \leq, \cap, \to \rangle.$$

This means that $\leq$ is reflexive and transitive, but not necessarily anti-symmetric

$$A \leq_{\mathcal{T}} B \ \&\ B \leq_{\mathcal{T}} A \ \not\Rightarrow\ A =_{\mathcal{T}} B.$$

If $\mathcal{T}$ is compatible one can go over to equivalence classes and obtain a type structure

$$\mathcal{T}/{=_{\mathcal{T}}} = \langle \mathbb{T}/{=_{\mathcal{T}}}, \leq, \cap, \to \rangle.$$

If moreover $\mathcal{T} \in \mathrm{TT}^{\top}$, then $\mathcal{T}/{=_{\mathcal{T}}}$ has top $[\top]$. In this structure $A \cap B$ is $\inf\{A, B\}$, the greatest lower bound of A and B. If $\mathcal{T}$ is also compatible, then $\to$ can be properly defined on the equivalence classes. This will be done in Section 15.3.

Specific intersection type theories

Now we will construct several, in total thirteen, type theories that will play an important role in later chapters, by introducing the following axiom schemes, rule schemes and axioms. Only two of them are non-compatible, so we obtain eleven type structures.

In the following φ, ω and $\top$ are distinct atoms differing from those in $\mathbb{A}_\infty$.

15.1.6. Notation. We introduce names for axiom(scheme)s and rule(scheme)s in Figure 15.1. Using these names a list of well-studied type structures can be specified in Figure 15.2 as the set of judgements axiomatized by mentioned rule(scheme)s and axiom(scheme)s.

Axioms	
(ω_{Scott})	$(\top \to \omega) = \omega$
(ω_{Park})	$(\omega \to \omega) = \omega$
$(\omega\varphi)$	$\omega \leq \varphi$
$(\varphi{\to}\omega)$	$(\varphi \to \omega) = \omega$
$(\omega{\to}\varphi)$	$(\omega \to \varphi) = \varphi$
(I)	$(\varphi \to \varphi) \cap (\omega \to \omega) = \varphi$
Axiom schemes	
$(\top)$	$A \leq \top$
$(\top{\to})$	$\top \leq (A \to \top)$
$(\top_{\text{lazy}})$	$(A \to B) \leq (\top \to \top)$
$({\to}\cap)$	$(A \to B) \cap (A \to C) \leq A \to B \cap C$
$({\to}\cap^{=})$	$(A \to B) \cap (A \to C) = A \to B \cap C$
Rule schemes	
$({\to})$	$\dfrac{A' \leq A \quad B \leq B'}{(A \to B) \leq (A' \to B')}$
$({\to}^{=})$	$\dfrac{A' = A \quad B = B'}{(A \to B) = (A' \to B')}$

Figure 15.1: Possible Axioms and Rules concerning $\leq$.

15.1.7. Definition. In Figure 15.2 a collection of TTs is defined. For each name $\mathcal{T}$ a set of atoms $\mathbb{A}^{\mathcal{T}}$ and a set of rules and axiom(scheme)s are given. The type theory $\mathcal{T}$ is the smallest set of judgements of the form $A \leq B$ with $A, B \in \mathbb{T}^{\mathcal{T}} = \mathbb{T}_\cap^{\mathbb{A}^{\mathcal{T}}}$ which is closed under the axiom(scheme)s and the rule(scheme)s of Definition 15.1.1 and the corresponding ones in Figure 15.2.

15.1.8. Remark. (i) Note that CDS and CD are non-compatible, while the other eleven are compatible.

(ii) The first ten type theories of Figure 15.2 belong clearly to $\mathrm{TT}^{\top}$. In Lemma 15.1.14(i) we will see that also $\mathrm{HL} \in \mathrm{TT}^{\top}$ with φ as top. Instead CDS and CD do not belong to $\mathrm{TT}^{\top}$, as shown in Lemma 15.1.14(ii) and (iii).

In this list the given order is logical, rather than historical, and some of the references define the models directly, others deal with the corresponding filter models (see Sections 17 and 18): Scott [1972], Park [1976], Coppo et al. [1987], Honsell and Ronchi Della Rocca [1992], Dezani-Ciancaglini et al. [2005], Barendregt et al. [1983], Abramsky and Ong [1993], Plotkin [1993], Engeler [1981], Coppo et al. [1979], Honsell and Lenisa [1999], Coppo et al. [1981], Coppo and Dezani-Ciancaglini [1980]. These theories are denoted by names (respectively acronymes) of the author(s) who have first considered the λ-model induced by such a theory.

$\mathcal{T}$	$\mathbb{A}^{\mathcal{T}}$	Rules	Axiom Schemes	Axioms
Scott	$\{\top,\omega\}$	$(\to)$	$(\to\cap),(\top),(\top\to)$	$(\omega_{\mathrm{Scott}})$
Park	$\{\top,\omega\}$	$(\to)$	$(\to\cap),(\top),(\top\to)$	(ω_{Park})
CDZ	$\{\top,\varphi,\omega\}$	$(\to)$	$(\to\cap),(\top),(\top\to)$	$(\omega\varphi),(\varphi\to\omega),(\omega\to\varphi)$
HR	$\{\top,\varphi,\omega\}$	$(\to)$	$(\to\cap),(\top),(\top\to)$	$(\omega\varphi),(\varphi\to\omega),(I)$
DHM	$\{\top,\varphi,\omega\}$	$(\to)$	$(\to\cap),(\top),(\top\to)$	$(\omega\varphi),(\omega\to\varphi),(\omega_{\mathrm{Scott}})$
BCD	$\mathbb{A}^{\top}_{\infty}$	$(\to)$	$(\to\cap),(\top),(\top\to)$	
AO	$\{\top\}$	$(\to)$	$(\to\cap),(\top)$	$(\top_{\mathrm{lazy}})$
Plotkin	$\{\top,\omega\}$	$(\to^{=})$	$(\top)$	–
Engeler	$\mathbb{A}^{\top}_{\infty}$	$(\to^{=})$	$(\to\cap^{=}),(\top),(\top\to)$	–
CDS	$\mathbb{A}^{\top}_{\infty}$	–	$(\top)$	–
HL	$\{\varphi,\omega\}$	$(\to)$	$(\to\cap)$	$(\omega\varphi),(\omega\to\varphi),(\varphi\to\omega)$
CDV	$\mathbb{A}_{\infty}$	$(\to)$	$(\to\cap)$	–
CD	$\mathbb{A}_{\infty}$	–	–	–

Figure 15.2: Various type theories

The expressive power of intersection types is remarkable. This will become apparent when we will use them as a tool for characterizing properties of λ-terms (see Sections 19.2 and 18.3), and for describing different λ-models (see Section 18). 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 modulo $=$, the structure of a meet semi-lattice with respect to $\cap$. This appears natural when we think of types as subsets of a domain of discourse D, which is endowed with a (partial) application $\cdot : D\times D\to D$, and interpret $\cap$ as set-theoretic intersection, $\leq$ as set inclusion, and give $\to$ the *realizability interpretation*.

$$
\begin{array}{rcl}
[\![A]\!] & \subseteq & D \\
A\leq B & \Leftrightarrow & [\![A]\!]\subseteq[\![B]\!] \\
[\![A\cap B]\!] & = & [\![A]\!]\cap[\![B]\!] \\
[\![A\to B]\!] & = & [\![A]\!]\to[\![B]\!]=\{d\in D \mid d\cdot[\![A]\!]\subseteq[\![B]\!]\}.
\end{array}
$$

This semantics, due to Scott, will be studied in Section 19.1.

The type $\top\to\top$ is the set of functions which applied to an arbitrary element return again an arbitrary element. In that case axiom $(\top\to)$ expresses the fact that all the objects in our domain of discourse are total functions, i.e. that $\top$ is equal to $A\to\top$, hence $A\to\top = B\to\top$ for all A, B (Barendregt et al. [1983]). 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 $(\top\to)$. One still may postulate the weaker property $(\top_{\mathrm{lazy}})$ to make all functions total (Abramsky and Ong [1993]). It simply says that an element which is a function, because it maps A into B, maps also the whole universe into itself.

In Figure 15.3 below consider $\vdash_{\cap\top}^{\mathcal{T}}$ for the ten type theories above the horizontal line and $\vdash_\cap^{\mathcal{T}}$ for the other three. Define $\mathcal{T}_1 \subseteq \mathcal{T}_2$ as

$$\forall \Gamma, M, A.\ [\Gamma \vdash^{\mathcal{T}_1} M : A \ \Rightarrow\ \Gamma \vdash^{\mathcal{T}_2} M : A].$$

If this is the case we have connected $\mathcal{T}_1$ with an edge towards the higher positioned $\mathcal{T}_2$. In Exercise 16.3.21 we will show that the edges denote strict inclusions.

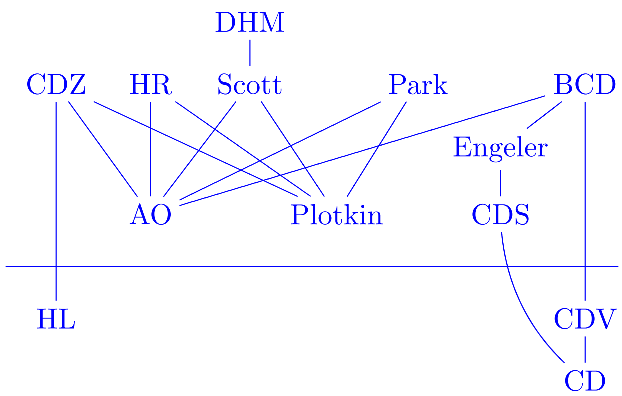

Figure 15.3: Inclusion among some intersection type theories.

The intended interpretation of arrow types also motivates axiom $(\to\cap)$, 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. into $B \cap C$), see Barendregt et al. [1983].

Rule $(\to)$ is again 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 Barendregt et al. [1983].

The rule $(\to\cap^=)$ is similar to the rule $(\to\cap)$. It capture properties of the graph models for the untyped lambda calculus, see Plotkin [1975] and Engeler [1981], as we shall discuss in Section 19.3.

In order to capture aspects of the λI-calculus we introduce TTs without an explicit mention of a top.

The remaining axioms express peculiar properties of D_∞-like inverse limit models, see Barendregt et al. [1983], Coppo et al. [1984], Coppo et al. [1987], Honsell and Ronchi Della Rocca [1992], Honsell and Lenisa [1993], Alessi, Dezani-Ciancaglini and Honsell [2004]. We shall discuss them in more detail in Section 19.3.

Some classes of type theories

Now we will consider some classes of TT. In order to do this, we list the relevant defining properties.

15.1.9. Definition. We define special subclasses of TT.

Class	Defining axiom(-scheme)(s) or rule
graph	$(\to^=), (\to\cap^=), (\top)$
lazy	$(\to), (\to\cap), (\top), (\top_{\text{lazy}})$
natural	$(\to), (\to\cap), (\top), (\top\to)$
proper	$(\to), (\to\cap)$

15.1.10. Notation. The sets of graph, lazy, natural and proper type theories are denoted by respectively $\mathrm{GTT}^\top$, $\mathrm{LTT}^\top$ $\mathrm{NTT}^\top$ and PTT.

15.1.11. Remark. The type theories of Figure 15.2 are classified as follows.

non compatible	CD, CDS
$\mathrm{GTT}^\top$	Plotkin, Engeler
$\mathrm{LTT}^\top$	AO
$\mathrm{NTT}^\top$	Scott, Park, CDZ, HR, DHM, BCD
PTT	CDV, HL

15.1.12. Remark. One has $\mathrm{NTT}^\top \subseteq \mathrm{LTT}^\top \subseteq \mathrm{GTT}^\top \subseteq \mathrm{TT}$ and $\mathrm{LTT}^\top \subseteq \mathrm{PTT} \subseteq \mathrm{TT}$. These inclusions are sharp.

Some properties about specific TTs

Results about proper type theories

15.1.13. Proposition. *Let $\mathcal{T}$ be a proper type theory. Then we have*

(i) $(A\to B)\cap(A'\to B') \leq (A\cap A')\to(B\cap B')$;

(ii) $(A_1\to B_1)\cap\ldots\cap(A_n\to B_n) \leq (A_1\cap\ldots\cap A_n)\to(B_1\cap\ldots\cap B_n)$;

(iii) $(A\to B_1)\cap\ldots\cap(A\to B_n) = A\to(B_1\cap\ldots B_n)$.

Proof. (i)
$$\begin{aligned}(A\to B)\cap(A'\to B') &\leq ((A\cap A')\to B)\cap((A\cap A')\to B')\\ &\leq (A\cap A')\to(B\cap B'),\end{aligned}$$
by respectively $(\to)$ and $(\to\cap)$.

(ii) Similarly (i.e. by induction on $n>1$, using (i) for the induction step).

(iii) By (ii) one has $(A\to B_1)\cap\ldots\cap(A\to B_n) \leq A\to(B_1\cap\ldots B_n)$. For $\geq$ use $(\to)$ to show that $A\to(B_1\cap\ldots\cap B_n) \leq (A\to B_i)$, for all i. ∎

It follows that the mentioned equality and inequalities hold for Scott, Park, CDZ, HR, DHM, BCD, AO, HL and CDV.

Results about the type theories of Figure 15.2

15.1.14. LEMMA. (i) *φ is the top and ω the bottom element in* HL*.*
(ii) CDV *has no top element.*
(iii) CD *has no top element.*

PROOF. (i) By induction on the generation of $\mathbb{T}^{\mathrm{HL}}$ one shows that $\omega \leq A \leq \varphi$ for all $A \in \mathbb{T}^{\mathrm{HL}}$.

(ii) If α is a fixed atom and

$$\mathcal{B}_\alpha := \alpha \mid \mathcal{B}_\alpha \cap \mathcal{B}_\alpha$$

and $A \in \mathcal{B}_\alpha$, then one can show by induction on the generation of $\leq_{\mathrm{CDV}}$ that $A \leq_{\mathrm{CDV}} B \;\Rightarrow\; A \in \mathcal{B}_\alpha$. Hence if $\alpha \leq_{\mathrm{CDV}} B$, then $B \in \mathcal{B}_\alpha$. Since $\mathcal{B}_{\alpha_1}$ and $\mathcal{B}_{\alpha_2}$ are disjoint when α_1 and α_2 are two different atoms, we conclude that CDV has no top element.

(iii) Similar to (ii). ∎

15.1.15. REMARK. By the above lemma φ turns out to be the top element in HL. But we will not use this and therefore denote it by the name φ and not $\top$.

In the following lemmas 15.1.16-15.1.20 we study the positions of the atoms ω, and φ in the compatible TTs introduced in Figure 15.2. The principal result is that $\omega < \varphi$ in HL and, as far as applicable,

$$\omega < \varphi < \top,$$

in the theories Scott, Park, CDZ, HR, DHM and Plotkin.

15.1.16. LEMMA. *Let $\mathcal{T} \in \{$Scott, Park, CDZ, HR, DHM, BCD, Engeler$\}$ be as defined in Figure 15.2. Define inductively the following collection of types*

$$\boxed{\mathcal{B} \;:=\; \top \mid \mathbb{T}^{\mathcal{T}} \to \mathcal{B} \mid \mathcal{B} \cap \mathcal{B}}$$

Then $\mathcal{B} = \{A \in \mathbb{T}^{\mathcal{T}} \mid A =_{\mathcal{T}} \top\}$.

PROOF. By induction on the generation of $A \leq_{\mathcal{T}} B$ one proves that $\mathcal{B}$ is closed upwards. This gives $\top \leq A \;\Rightarrow\; A \in \mathcal{B}$.

By induction on the definition of $\mathcal{B}$ one shows, using $(\top\!\to)$ and $(\to)$, that $A \in \mathcal{B} \;\Rightarrow\; \top \leq A$.

Therefore

$$A =_{\mathcal{T}} \top \;\Leftrightarrow\; \top \leq A \;\Leftrightarrow\; A \in \mathcal{B}. \;∎$$

15.1.17. LEMMA. *For $\mathcal{T} \in \{$AO, Plotkin$\}$ define inductively*

$$\boxed{\mathcal{B} \;:=\; \top \mid \mathcal{B} \cap \mathcal{B}}$$

Then $\mathcal{B} = \{A \in \mathbb{T}^{\mathcal{T}} \mid A =_{\mathcal{T}} \top\}$, hence $\top \to \top \neq_{\mathcal{T}} \top$.

Proof. Similar to the proof of 15.1.14, but easier. ■

15.1.18. Lemma. *For* $\mathcal{T} \in \{\text{CDZ}, \text{HR}, \text{DHM}\}$ *define by mutual induction*

$$\begin{array}{rcl} \mathcal{B} & = & \varphi \mid \top \mid \mathbb{T}^{\mathcal{T}} \to \mathcal{B} \mid \mathcal{H} \to \mathbb{T}^{\mathcal{T}} \mid \mathcal{B} \cap \mathcal{B} \\ \mathcal{H} & = & \omega \mid \mathcal{B} \to \mathcal{H} \mid \mathcal{H} \cap \mathbb{T}^{\mathcal{T}} \mid \mathbb{T}^{\mathcal{T}} \cap \mathcal{H} \end{array}$$

Then

$$\begin{array}{rcl} \varphi \leq B & \Rightarrow & B \in \mathcal{B}, \\ A \leq \omega & \Rightarrow & A \in \mathcal{H}. \end{array}$$

Proof. By induction on $\leq_{\mathcal{T}}$ one shows

$$A \leq B \;\Rightarrow\; (A \in \mathcal{B} \;\Rightarrow\; B \in \mathcal{B}) \;\Rightarrow\; (B \in \mathcal{H} \;\Rightarrow\; A \in \mathcal{H}).$$

From this the assertion follows immediately. ■

15.1.19. Lemma. *We work with the theory* HL.

(i) *Define by mutual induction*

$$\begin{array}{rcl} \mathcal{B} & = & \varphi \mid \mathcal{H} \to \mathcal{B} \mid \mathcal{B} \cap \mathcal{B} \\ \mathcal{H} & = & \omega \mid \mathcal{B} \to \mathcal{H} \mid \mathcal{H} \cap \mathbb{T} \mid \mathbb{T} \cap \mathcal{H} \end{array}$$

Then

$$\begin{array}{rcl} \mathcal{B} & = & \{A \in \mathbb{T}^{\text{HL}} \mid A =_{\text{HL}} \varphi\}; \\ \mathcal{H} & = & \{A \in \mathbb{T}^{\text{HL}} \mid A =_{\text{HL}} \omega\}. \end{array}$$

(ii) $\omega \neq_{\text{HL}} \varphi$ *and hence* $\omega <_{\text{HL}} \varphi$.

Proof. (i) By induction on $\leq_{\mathcal{T}}$ one shows

$$A \leq B \;\Rightarrow\; (A \in \mathcal{B} \;\Rightarrow\; B \in \mathcal{B}) \;\&\; (B \in \mathcal{H} \;\Rightarrow\; A \in \mathcal{H}).$$

This gives

$$(\varphi \leq B \;\Rightarrow\; B \in \mathcal{B}) \;\&\; (A \leq \omega \;\Rightarrow\; A \in \mathcal{H}).$$

By simultaneous induction on the generation of $\mathcal{B}$ and $\mathcal{H}$ one shows, using that ω is the bottom element of HL, by Lemma 15.1.14(i),

$$(B \in \mathcal{B} \;\Rightarrow\; B = \varphi) \;\&\; (A \in \mathcal{H} \;\Rightarrow\; A = \omega).$$

Now the assertion follows immediately.

(ii) By (i). ■

15.1.20. Proposition. *In* HL *we have* $\omega < \varphi$ *and as far as applicable we have for the other systems of Figure 15.2*

$$\omega < \varphi < \top.$$

More precisely,

(i) $\omega \leq \varphi$ *and* $\omega \neq \varphi$ *in* HL.

In all other systems

(ii) $\omega \leq \varphi$, $\omega \leq \top$, $\varphi \leq \top$;

(iii) $\omega \neq \varphi$, $\omega \neq \top$, $\varphi \neq \top$.

PROOF. (i) By ($\omega\varphi$) and Lemma 15.1.19.
(ii) By ($\omega\varphi$) and ($\top$).
(iii) By Lemmas 15.1.16-15.1.18. ■

15.2. Type assignment

Assignment of types from type theories

In this subsection we define for a $\mathcal{T}$ in TT a type assignment system $\lambda_\cap^{\mathcal{T}}$, that assigns to untyped lambda terms a (possibly empty set of) types in $\mathbb{T}^{\mathcal{T}}$. For a $\mathcal{T}$ in TT$^\top$ we also define a type assignment system $\lambda_{\cap\top}^{\mathcal{T}}$.

15.2.1. DEFINITION. (i) A $\mathcal{T}$-*statement* is of the form $M : A$ with the *subject* an untyped lambda term $M \in \Lambda$ and the *predicate* a type $A \in \mathbb{T}^{\mathcal{T}}$.
(ii) A $\mathcal{T}$-*declaration* is a $\mathcal{T}$-statement of the form $x : A$.
(iii) A $\mathcal{T}$-*basis* Γ is a finite set of $\mathcal{T}$-declarations, with all variables distinct.
(iv) A $\mathcal{T}$-*assertion* is of the form $\Gamma \vdash M : A$, where $M : A$ is a $\mathcal{T}$-statement and Γ is a $\mathcal{T}$-basis.

15.2.2. DEFINITION. (i) The *(basic) type assignment system* $\lambda_\cap^{\mathcal{T}}$ derives $\mathcal{T}$-assertions by the following axioms and rules.

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

Figure 15.4: Basic type assignment system $\lambda_\cap^{\mathcal{T}}$.

(ii) If $\mathcal{T}$ has a top element $\top$, then the $\top$-*type assignment system* $\lambda_{\cap\top}^{\mathcal{T}}$ is defined by adding the extra axiom to the basic system

($\top$-universal)	$\Gamma \vdash M : \top$

Figure 15.5: The extra axiom for the top assignment system $\lambda_{\cap\top}^{\mathcal{T}}$

15.2.3. NOTATION. (i) We write $\Gamma \vdash_{\cap\top}^{\mathcal{T}} M : A$ or $\Gamma \vdash_\cap^{\mathcal{T}} M : A$ if $\Gamma \vdash M : A$ is derivable in $\lambda_{\cap\top}^{\mathcal{T}}$ or $\lambda_\cap^{\mathcal{T}}$ respectively.
(ii) The assertion $\vdash_{\cap\top}^{\mathcal{T}}$ may also be written as $\vdash^{\mathcal{T}}$, $\vdash_{\cap\top}$ or simply $\vdash$ if by the context there is little danger of confusion. Similarly, $\vdash_\cap^{\mathcal{T}}$ may be written as $\vdash^{\mathcal{T}}$, $\vdash_\cap$ or $\vdash$.

(iii) $\lambda^{\mathcal{T}}_{\cap(\top)}$ may be denoted by $\lambda_{\cap(\top)}$.

15.2.4. Example. Let $\mathcal{T} \in \mathrm{TT}^{\top}$ with $A, B \in \mathbb{T}^{\mathcal{T}}$. Write $W \equiv (\lambda x.xx)$.

(i) $\vdash^{\mathcal{T}}_{\cap} W : A \cap (A{\to}B){\to}B$.

$\vdash^{\mathcal{T}}_{\cap\top} WW : \top$, but WW does not have a type in $\lambda^{\mathcal{T}}_{\cap}$.

(ii) Let $M \equiv \mathsf{KI}(WW)$. Then $\vdash M : (A{\to}A)$ in $\lambda^{\mathcal{T}}_{\cap\top}$.

(iii) (van Bakel) Let $M \equiv \lambda yz.\mathsf{K}z(yz)$ and $N \equiv \lambda yz.z$. Then $M \twoheadrightarrow_{\boldsymbol{\beta}} N$. We have $\vdash^{\mathcal{T}}_{\cap} N : B{\to}A{\to}A$, $\vdash^{\mathcal{T}}_{\cap\top} M : B{\to}A{\to}A$, but $\not\vdash^{\mathcal{T}}_{\cap} M : B{\to}A{\to}A$.

(iv) $\not\vdash^{\mathrm{CD}}_{\cap} \mathsf{I} : ((A \cap B){\to}C){\to}((B \cap A){\to}C)$.

In general the type assignment systems $\lambda^{\mathcal{T}}_{\cap\top}$ will be used for the the $\lambda\mathsf{K}$-calculus and $\lambda^{\mathcal{T}}_{\cap}$ for the $\lambda\mathsf{I}$-calculus.

15.2.5. Definition. Define the rules $(\cap\mathrm{E})$

$$\frac{\Gamma \vdash M : (A \cap B)}{\Gamma \vdash M : A} \qquad \frac{\Gamma \vdash M : (A \cap B)}{\Gamma \vdash M : B}$$

Notice that these rules are derived in $\lambda^{\mathcal{T}}_{\cap}$, $\lambda^{\mathcal{T}}_{\cap\top}$ for all $\mathcal{T}$.

15.2.6. Lemma. *In $\lambda^{\mathcal{T}}_{\cap}$ one has the following.*

(i) $\Gamma \vdash M : A \;\Rightarrow\; \mathrm{FV}(M) \subseteq \mathrm{dom}(\Gamma)$.

(ii) $\Gamma \vdash M : A \;\Rightarrow\; (\Gamma \restriction \mathrm{FV}(M)) \vdash M : A$.

(iii) *If in $\mathcal{T}$ with top $\top$ one has $\top = \top{\to}\top$, then*

$$\mathrm{FV}(M) \subseteq \mathrm{dom}(\Gamma) \;\Rightarrow\; \Gamma \vdash M : \top.$$

Proof. (i), (ii) By induction on the derivation.

(iii) By induction on M. ∎

Notice that $\Gamma \vdash M : A \Rightarrow \mathrm{FV}(M) \subseteq \mathrm{dom}(\Gamma)$ does not hold in $\lambda^{\mathcal{T}}_{\cap\top}$, since by axiom ($\top$ universal) we have $\vdash^{\mathcal{T}} M : \top$ for all $\mathcal{T}$ and all M.

15.2.7. Remark. For the type theories of Figure 15.2 with $\top$ we have defined the type assignment systems $\lambda^{\mathcal{T}}_{\cap}$. For those system having a top, there is also the type assignment system $\lambda^{\mathcal{T}}_{\cap\top}$. We will use for the type theories in Figure 15.2 only one of the two possibilities. For the first ten systems, i.e. Scott, Park, CDZ, HR, DHM, BCD, AO, Plotkin, Engeler and CDS, we only consider $\lambda^{\mathcal{T}}_{\cap\top}$. For the other 3 systems, i.e. HL, CDV and CD, we will only consider $\lambda^{\mathcal{T}}_{\cap}$. In fact by Lemma 15.1.14(ii) and (iii) we know that CDV and CD have no top at all. The system HL has a top, but we will not use it, as we do not know interesting properties of $\lambda^{\mathrm{HL}}_{\cap\top}$. So, for example, $\vdash^{\mathrm{Scott}}$ will be always $\vdash^{\mathrm{Scott}}_{\cap\top}$, whereas $\vdash^{\mathrm{HL}}$ will be always $\vdash^{\mathrm{HL}}_{\cap}$. The reader will be reminded of this. We do not know wether there exist TTs where the interplay of $\lambda^{\mathcal{T}}_{\cap}$ and $\lambda^{\mathcal{T}}_{\cap\top}$ yields results of interest.

Admissible rules

15.2.8. PROPOSITION. *The following rules are admissible in $\lambda_{\cap(\top)}^\mathcal{T}$.*

(*weakening*)	$\dfrac{\Gamma \vdash M : A \quad x \notin \Gamma}{\Gamma, x{:}B \vdash M : A}$;
(*strengthening*)	$\dfrac{\Gamma, x{:}B \vdash M : A \quad x \notin FV(M)}{\Gamma \vdash M : A}$;
(*cut*)	$\dfrac{\Gamma, x{:}B \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash (M[x := N]) : A}$;
($\leq$-L)	$\dfrac{\Gamma, x{:}B \vdash M : A \quad C \leq_\mathcal{T} B}{\Gamma, x{:}C \vdash M : A}$;
($\to$L)	$\dfrac{\Gamma, y{:}B \vdash M : A \quad \Gamma \vdash N : C \quad x \notin \Gamma}{\Gamma, x{:}(C \to B) \vdash (M[y := xN]) : A}$;
($\cap$L)	$\dfrac{\Gamma, x{:}A \vdash M : B}{\Gamma, x{:}(A \cap C) \vdash M : B}$.

Figure 15.6: Various admissible rules.

PROOF. By induction on the structure of derivations. ■

Proofs later on in Part III will freely use the rules of the above proposition.

As we remarked earlier, there are various equivalent alternative presentations of intersection type assignment systems. We have chosen a natural deduction presentation, where $\mathcal{T}$-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 15.2.8, see Barbanera et al. [1995], Barendregt and Ghilezan [n.d.]. Next to this we could have formulated the rules so that $\mathcal{T}$-bases "multiply". Notice that because of the presence of the type constructor $\cap$, a special notion of *multiplication of $\mathcal{T}$-bases* can be given.

15.2.9. DEFINITION (Multiplication of $\mathcal{T}$-bases).

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

Accordingly we define:

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

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

15.2.10. PROPOSITION. *The following rules are admissible in all* $\lambda^{\mathcal{T}}_{\cap(\top)}$.

(*multiple weakening*)	$\dfrac{\Gamma_1 \vdash M : A}{\Gamma_1 \uplus \Gamma_2 \vdash M : A}$
(*relevant* $\to$E)	$\dfrac{\Gamma_1 \vdash M : A \to B \quad \Gamma_2 \vdash N : A}{\Gamma_1 \uplus \Gamma_2 \vdash MN : B}$
(*relevant* $\cap$I)	$\dfrac{\Gamma_1 \vdash M : A \quad \Gamma_2 \vdash M : B}{\Gamma_1 \uplus \Gamma_2 \vdash M : A \cap B}$

PROOF. By induction on derivations. ∎

In Exercise 16.3.17, it will be shown that we can replace rule $(\leq)$ with other more perspicuous rules. This is possible as soon as we will have proved appropriate "inversion" theorems for $\lambda^{\mathcal{T}}_{\cap(\top)}$. For some very special theories, one can even omit altogether rule $(\leq)$, provided the remaining rules are reformulated "multiplicatively" with respect to $\mathcal{T}$-bases, see e.g. Di Gianantonio and Honsell [1993]. We shall not follow up this line of investigation.

In $\lambda^{\mathcal{T}}_{\cap(\top)}$, 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" judgements are derivable, (see Honsell and Ronchi Della Rocca [1992]). Rules like (*relevant* $\to$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 Girard et al. [1989]. We will not follow up this line of research either. See Honsell and Ronchi Della Rocca [1992].

Special type assignment for call-by-value λ-calculus

We will study later the type theory EHR with $\mathbb{A}^{\mathrm{EHR}} = \{\nu\}$ and the extra rule $(\to)$ and axioms $(\to\cap)$ and

$$A \to B \leq \nu.$$

The type assignment system $\lambda^{\mathrm{EHR}}_{\cap\nu}$ is defined by the axiom and rules of $\lambda^{\mathcal{T}}_{\cap}$ in Figure 15.4 with the extra axiom

(ν universal)	$\Gamma \vdash (\lambda x.M) : \nu.$

The type theory EHR has a top, namely ν, so one could consider it as an element of $\mathrm{TT}^{\top}$. This will not be done. Axiom (ν-universal) is different from ($\top$-universal) in Definition 15.2.2. This type assignment system has one particular application and will be studied in some exercises.

15.3. Type structures

Intersection type structures

Remember that a type algebra $\mathcal{A}$, see Definition **??**, is of the form $\mathcal{A} = \langle|\mathcal{A}|, \to\rangle$, i.e. just an arbitrary set $|\mathcal{A}|$ with a binary operation $\to$ on it.

15.3.1. Definition. (i) A *meet semi-lattice* is a structure

$$\mathcal{M} = \langle|\mathcal{M}|, \leq, \cap\rangle,$$

such that $\mathcal{M} = \langle|\mathcal{M}|, \leq, \cap\rangle$ is a partial order, for all $A, B \in |\mathcal{M}|$ the element $A \cap B$ (meet) is the greatest lower bound of A and B. MSL is the set of meet semi-lattices.

(ii) A *top meet semi-lattice* is a similar structure

$$\mathcal{M} = \langle|\mathcal{M}|, \leq, \cap, \top\rangle,$$

such that $\mathcal{M} = \langle|\mathcal{M}|, \leq, \cap\rangle$ is a MSL and $\top$ is the (unique) top of $\mathcal{M}$. $\mathrm{MSL}^\top$ is the set of top meet semi-lattices.

15.3.2. Definition. (i) An *(intersection) type structure* is a type algebra with the additional structure of a meet semi-lattice

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

TS is the set of type structures. The relation $\leq$ and the operation $\to$ have a priori no relation with each other, but in special structures this will be the case.

(ii) A *top type structure* is a type algebra that is also a top meet semi-lattice

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

$\mathrm{TS}^\top$ is the set of top type structures.

Notation. (i) As 'intersection' is everywhere in this Part III, we will omit this word and only speak about a *type structure*.

(ii) *Par abus de language* we also use $A, B, C, \ldots$ to denote arbitrary elements of type structures and we write $A \in \mathcal{S}$ for $A \in |\mathcal{S}|$.

If $\mathcal{T}$ is a type theory that is not compatible, like CD and CDS, then $\to$ cannot be defined on the equivalence classes. But if $\mathcal{T}$ is compatible, then one can work on the equivalence classes and obtain a type structure in which $\leq$ is a partial order.

15.3.3. Proposition. *Let $\mathcal{T}$ be a compatible type theory. Then $\mathcal{T}$ induces a type structure $\mathcal{T}/=$ defined as follows.*

$$\langle\mathbb{T}^{\mathcal{T}}/{=_{\mathcal{T}}}, \to, \leq, \cap\rangle,$$

by defining on the $=_\mathcal{T}$*-equivalence classes*

$$\begin{array}{rcl}[A]{\rightarrow}[B] & = & [A{\rightarrow}B]^2; \\ [A]\cap[B] & = & [A\cap B]; \\ [A]\leq[B] & \Leftrightarrow & A\leq_\mathcal{T} B.\end{array}$$

If moreover $\mathcal{T}$ *has a top* $\top$*, then* $\mathcal{T}/{=}$ *is a top type structure with* $[\top]$ *as top.*

PROOF. Here A, B, C range over $\mathbb{T}^\mathcal{T}$. Having realized this the rest is easy. Rule $(\rightarrow^=)$ is needed to ensure that $\rightarrow$ is well-defined. ∎

The (top) type structure $\mathcal{T}/=$, with $\mathcal{T}$ a type theory, is called a *syntactical* (top) type structure. In Proposition 15.3.6 we show that every type structure is isomorphic to a syntactical one.

Although essentially equivalent, type structures and type theories differ in the following. In the theories the types are freely generated from a fixed set of atoms and inequality can be controlled somewhat by choosing the right axioms and rules (this will be exploited in Section 19.3). In type structures one has the antisymmetric law $A\leq B\leq A \;\Rightarrow\; A=B$, which is in line with the common theory of partial orders (this will be exploited in Chapter 17).

Now the notion of type assignment will also be defined for intersection type structures. These structures arise naturally coming from algebraic lattices that are used towards obtaining a semantics for untyped lambda calculus.

15.3.4. DEFINITION. (i) Now let $\mathcal{S}\in\mathrm{TS}$. The notion of a $\mathcal{S}$-statement $M : A$, a $\mathcal{S}$-declaration $x : A$, a $\mathcal{S}$-basis and a $\mathcal{S}$-assertion $\Gamma\vdash M : A$ is as in Definition 15.2.1, now for $A\in\mathcal{S}$ an element of the type structure $\mathcal{S}$.

(ii) The notion $\Gamma\vdash^\mathcal{S}_\cap M : A$ is defined by the same set of axioms and rules as in Figure 15.4 where now $\leq_\mathcal{S}$ is the inequality of the structure $\mathcal{S}$. The assignment system $\lambda^\mathcal{S}_{\cap\top}$ has ($\top$-universal) as extra axiom.

The following result shows that for syntactic type structures type assignment is essentially the same as the one coming from the corresponding lambda theory.

15.3.5. PROPOSITION. *Let* $\mathcal{T}\in TT^{(\top)}$ *and let* $[\mathcal{T}] = \langle\mathcal{T}/=_\mathcal{T}, \leq, \cap, \rightarrow(,\top)\rangle$ *its corresponding (top) type structure. For a type* $A\in\mathcal{T}$ *write its equivalence class as* $[A]\in[\mathcal{T}]$*. For* $\Gamma = \{x_1 : B_1, \dots, x_n : B_n\}$ *a* $\mathcal{T}$*-basis write* $[\Gamma] = \{x_1 : [B_1], \dots, x_n : [B_n]\}$*, a* $[\mathcal{T}]$*-basis. Then*

$$\Gamma\vdash^\mathcal{T}_{\cap(\top)} M : A \;\Leftrightarrow\; [\Gamma]\vdash^{[\mathcal{T}]}_{\cap(\top)} M : [A].$$

PROOF. ($\Rightarrow$) By induction on the derivation of $\Gamma\vdash^\mathcal{T} M : A$. ($\Leftarrow$) Show by induction on the derivation of $[\Gamma]\vdash^{[\mathcal{T}]} M : [A]$ that for all $A'\in[A]$ and $\Gamma' = \{x_1 : B'_1, \dots, x_n : B'_n\}$, with $B'_i\in[B_i]$ for all $1\leq i\leq n$, one has

$$\Gamma'\vdash^\mathcal{T} M : A'. \; ∎$$

[2] Here we misuse notation in a suggestive way, by using the same notation $\rightarrow$ for equivalence classes as for types.

Using this result we could have defined type assignment first for type structures and then for compatible type theories via translation to the type assignment for its corresponding syntactical type structure, essentially by turning the previous result into a definition.

15.3.6. Proposition. *Every type structure is isomorphic to a syntactical one.*

Proof. For a type structure $\mathcal{S}$, define $\mathcal{T}_\mathcal{S}$ as follows. Take $\mathbb{A} = \{\underline{c} \mid c \in \mathcal{S}\}$. Define $\leq_{\mathcal{T}_\mathcal{S}}$ on $\mathbb{T} = \mathbb{T}_\cap^\mathbb{A}$ as follows. We make every element of $\mathbb{T}$ equal to an element of $\mathbb{A}$ by requiring

$$(\underline{a} \cap \underline{b}) =_{\mathcal{T}_\mathcal{S}} \underline{a \cap b}, \ \& \ (\underline{a} {\to} \underline{b}) =_{\mathcal{T}_\mathcal{S}} \underline{a {\to} b}.$$

This means of course $(\underline{a} \cap \underline{b}) \leq_{\mathcal{T}_\mathcal{S}} \underline{a \cap b}$, $(\underline{a} \cap \underline{b}) \geq_{\mathcal{T}_\mathcal{S}} \underline{a \cap b}$, etcetera. We moreover require

$$\frac{a \leq_\mathcal{S} b}{\underline{a} \leq_{\mathcal{T}_\mathcal{S}} \underline{b}}.$$

As a consequence $\underline{a} \leq_{\mathcal{T}_\mathcal{S}} \underline{\top}$ if $\mathcal{S}$ is a top type structure. The axioms and rules (refl), (trans), $(\to^=)$, (incl_L), (incl_R) and (glb) also hold automatically. Then $\mathcal{S} \cong \mathcal{T}_\mathcal{S}/=$. This can be seen as follows. Define $f : \mathcal{S} {\to} \mathcal{T}_\mathcal{S}/=$ by $f(a) = [\underline{a}]$. For the inverse, first define $g : \mathbb{T}_\cap^\mathbb{A} {\to} \mathcal{S}$ by

$$\begin{array}{rcl} g(\underline{c}) & = & c; \\ g(A {\to} B) & = & g(A) {\to} g(B); \\ g(A \cap B) & = & g(A) \cap g(B). \end{array}$$

Then show $A \leq_{\mathcal{T}_\mathcal{S}} B \ \Rightarrow \ g(A) \leq g(B)$. Finally set $f^{-1}([A]) = g(A)$, which is well defined. It is easy to show that f, f^{-1} constitute an isomorphism. ∎

15.3.7. Remark. Each of the eleven compatible type theories $\mathcal{T}$ in Figure 15.2 may be considered as the intersection type structure $\mathcal{T}/=$, also denoted as $\mathcal{T}$. For example Scott can be a name, a type theory or a type structure.

Categories of meet-semi lattices and type structures

For use in Chapter 17 we will introduce some categories related to given classes of type structures.

15.3.8. Definition. (i) The category **MSL** has as objects at most countable meet semi-lattices and as morphisms maps $f : \mathcal{M} {\to} \mathcal{M}'$, preserving $\leq, \cap$:

$$\begin{array}{rcl} A \leq B & \Rightarrow & f(A) \leq' f(B); \\ f(A \cap B) & = & f(A) \cap' f(B). \end{array}$$

(ii) The category $\mathbf{MSL}^\top$ is as **MSL**, but based on top meet semi-lattices. So now also $f(\top) = \top'$ for morphisms.

The difference between **MSL** and $\mathbf{MSL}^\top$ is that, in the **MSL** case, the top element is either missing or not relevant (not preserved by morphisms).

15.3.9. Definition. (i) The category **TS** has as objects the at most countable type structures and as morphisms maps $f:\mathcal{S}\to\mathcal{S}'$, preserving $\leq, \cap, \to$:

$$\begin{array}{rcl} A \leq B & \Rightarrow & f(A) \leq' f(B); \\ f(A\cap B) & = & f(A) \cap' f(B); \\ f(A\to B) & = & f(A) \to' f(B). \end{array}$$

(ii) The category $\mathbf{TS}^{\top}$ is as **TS**, but based on top type structures. Now also

$$f(\top) = \top'$$

for morphisms.

15.3.10. Definition. We define four full subcategories of **TS** by specifying in each case the objects.
(i) $\mathbf{GTS}^{\top}$ with as objects the graph top type structures.
(ii) $\mathbf{LTS}^{\top}$ with as objects the lazy top type structures.
(iii) $\mathbf{NTS}^{\top}$ with as objects the natural top type structures.
(iv) **PTS** with as objects the proper type structures.

15.4. Filters

15.4.1. Definition. (i) Let $\mathcal{T}\in \mathrm{TT}$ and $X \subseteq \mathbb{T}^{\mathcal{T}}$. Then X is a *filter* over $\mathcal{T}$ if the following hold.

(1) X is non-empty;

(2) $A\in X\ \&\ A\leq B \ \Rightarrow\ B\in X$;

(3) $A, B\in X \ \Rightarrow\ A\cap B\in X$.

(ii) Write $\mathcal{F}^{\mathcal{T}} = \{X\subseteq \mathcal{T} \mid X \text{ is a filter over } \mathcal{T}\}$.

We loosely say that filters are non-empty sets of types closed under $\leq$ and $\cap$.

15.4.2. Definition. Let $\mathcal{T}\in\mathrm{TT}$.
(i) For $A\in\mathbb{T}^{\mathcal{T}}$ write $\uparrow A = \{B\in\mathbb{T}^{\mathcal{T}} \mid A\leq B\}$.
(ii) For a non-empty $X\subseteq\mathbb{T}^{\mathcal{T}}$ define $\uparrow X$ to be the least filter over $\mathbb{T}^{\mathcal{T}}$ containing X; it can be described explicitly by

$$\uparrow X = \{B\in\mathbb{T}^{\mathcal{T}} \mid \exists n\geq 1\, \exists A_1,\ldots,A_n\in X. A_1\cap\ldots\cap A_n \leq B\}.$$

15.4.3. Remark. $C\in\ \uparrow\{B_i \mid i\in\mathcal{I}\neq\emptyset\} \ \Leftrightarrow\ \exists I\subseteq_{\mathrm{fin}}\mathcal{I}.[I\neq\emptyset\ \&\ \bigcap_{i\in I} B_i \leq C]$.

15.4.4. Proposition. *Let* $\mathcal{T}\in\mathrm{TT}^{\top}$.
(i) $\mathcal{F}^{\mathcal{T}} = \langle\mathcal{F}^{\mathcal{T}}, \subseteq\rangle$ *is a complete lattice, with for* $\mathcal{X}\subseteq\mathcal{F}^{\mathcal{T}}$ *the sup is*

$$\begin{array}{rcll} \bigsqcup\mathcal{X} & = & \uparrow(\cup\mathcal{X}), & \textit{if } X\neq\emptyset, \\ \bigsqcup\mathcal{X} & = & \{\top\}, & \textit{else}. \end{array}$$

(ii) *For* $A\in\mathbb{T}^{\mathcal{T}}$ *one has* $\uparrow A = \uparrow\{A\}$ *and* $\uparrow A\in\mathcal{F}^{\mathcal{T}}$.

(iii) *For* $A,B\in\mathbb{T}^{\mathcal{T}}$ *one has* $\uparrow A\sqcup\uparrow B=\uparrow(A\cap B)$.
(iv) *For* $X\in\mathcal{F}^{\mathcal{T}}$ *one has*

$$\begin{aligned} X &= \bigsqcup\{\uparrow A \mid A\in X\}\\ &= \bigsqcup\{\uparrow A \mid \uparrow A\subseteq X\}\\ &= \bigcup\{\uparrow A \mid A\in X\}\\ &= \bigcup\{\uparrow A \mid \uparrow A\subseteq X\}. \end{aligned}$$

(v) $\{\uparrow A \mid A\in\mathbb{T}^{\mathcal{T}}\}$ *is the set of finite elements of* $\mathcal{F}^{\mathcal{T}}$.

Proof. Easy. ■

15.4.5. Definition. Let $\mathcal{T}\in\mathrm{TT}$. Then $\mathcal{F}_s^{\mathcal{T}}=\mathcal{F}^{\mathcal{T}}\cup\{\emptyset\}$ is the extension of $\mathcal{F}^{\mathcal{T}}$ with the emptyset.

15.4.6. Proposition. *Let* $\mathcal{T}\in\mathrm{TT}$.
(i) $\mathcal{F}_s^{\mathcal{T}}=\langle\mathcal{F}_s^{\mathcal{T}},\subseteq\rangle$ *is a complete lattice, with for* $\mathcal{X}\subseteq\mathcal{F}_s^{\mathcal{S}}$ *the sup is*

$$\bigsqcup\mathcal{X}=\begin{cases}\emptyset, & \text{if } \mathcal{X}=\emptyset \text{ or } \mathcal{X}=\{\emptyset\},\\ \uparrow(\cup\mathcal{X}), & \text{else}.\end{cases}$$

(ii) *For* $A\in\mathbb{T}^{\mathcal{T}}$ *one has* $\uparrow A=\uparrow\{A\}$ *and* $\uparrow A\in\mathcal{F}_s^{\mathcal{T}}$.
(iii) *For* $A,B\in\mathbb{T}^{\mathcal{T}}$ *one has* $\uparrow A\sqcup\uparrow B=\uparrow(A\cap B)$.
(iv) *For* $X\in\mathcal{F}_s^{\mathcal{T}}$ *one has*

$$\begin{aligned} X &= \bigsqcup\{\uparrow A \mid A\in X\} &&= \bigsqcup\{\uparrow A \mid \uparrow A\subseteq X\}\\ &= \bigcup\{\uparrow A \mid A\in X\} &&= \bigcup\{\uparrow A \mid \uparrow A\subseteq X\}. \end{aligned}$$

(v) $\{\uparrow A \mid A\in\mathbb{T}^{\mathcal{T}}\}\cup\{\emptyset\}$ *is the set of finite elements of* $\mathcal{F}_s^{\mathcal{T}}$.

Proof. Immediate. ■

15.4.7. Remark. The items 15.1.9-15.2.10 and 15.4.1-15.4.6 are about type theories, but can be translated immediately to structures and if no $\to$ are involved to meet-semi lattices. For example Proposition 15.1.13 also holds for a proper type structure, hence it holds for Scott, Park, CDZ, HR, DHM, BCD, AO, HL and CDV considered as type structures. Also 15.1.14-15.1.20 immediately yield corresponding valid statements for the corresponding type structures, though the proof for the type theories cannot be translated to proofs for the type structures because they are by induction on the syntactic generation of $\mathbb{T}$ or $\leq$. Also 15.2.4-15.2.10 hold for type structures, as follows immediately from Propositions 15.3.5 and 15.3.6. Finally 15.4.1-15.4.6 can be translated immediately to type structures and meet semi-lattices. Therefore in the following chapters everywhere the type theories may be translated to type structures (or if no $\to$ is involved to meet semi-lattices). In Chapter 17 we work directly with meet semi-lattices and type structures and not with type theories, because there a partial order is needed.

15.5. Exercises 16.10.2006:1032

15.5.1. Show that $\Gamma, x{:}\top \vdash^{\mathcal{T}}_{\cap\top} M : A \;\Rightarrow\; \Gamma \vdash^{\mathcal{T}}_{\cap\top} M : A$.

15.5.2. Define the system $\mathtt{K}$ and the type assignment system $\lambda^{\mathtt{K}}_{\cap}$ of Krivine [1990] are CD and $\lambda^{\mathrm{CD}}_{\cap}$, but with rule $(\leq)$ replaced by

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

Similarly $\mathtt{K}^{\top}$ and $\lambda^{\mathtt{K}^{\top}}_{\cap\top}$ are CDS and $\lambda^{\mathrm{CDS}}_{\cap\top}$, with $(\leq)$ replaced by $(\cap\mathrm{E})$. Show that

(i) $\Gamma \vdash^{\mathtt{K}} M : A \quad\Leftrightarrow\quad \Gamma \vdash^{\mathrm{CD}}_{\cap} M : A$.

(ii) $\Gamma \vdash^{\mathtt{K}^{\top}} M : A \quad\Leftrightarrow\quad \Gamma \vdash^{\mathrm{CDS}}_{\cap\top} M : A$.

15.5.3. (i) Show that $\lambda x.xxx$ and $(\lambda x.xx)\mathsf{I}$ are typable in system $\mathtt{K}$.
(ii) Show that all closed terms in normal forms are typable in system $\mathtt{K}$.

15.5.4. Show the following:

(i) $\vdash^{\mathtt{K}} \lambda z.\mathsf{KI}(zz) : (A{\to}B) \cap A{\to}C{\to}C$.

(ii) $\vdash^{\mathtt{K}^{\top}} \lambda z.\mathsf{KI}(zz) : \top{\to}C{\to}C$.

(iii) $\vdash^{\mathrm{BCD}}_{\cap\top} \lambda z.\mathsf{KI}(zz) : \top{\to}(A{\to}B \cap C){\to}A{\to}B$.

15.5.5. For $\mathcal{T}$ a type theory, $M, N \in \Lambda$ and $x \notin \mathrm{dom}(\Gamma)$ show

(i) $\Gamma \vdash^{\mathcal{T}}_{\cap} M : A \quad\Rightarrow\quad \Gamma \vdash^{\mathcal{T}}_{\cap} M[x{:}= N] : A$;
(ii) $\Gamma \vdash^{\mathcal{T}}_{\cap\top} M : A \quad\Rightarrow\quad \Gamma \vdash^{\mathcal{T}}_{\cap\top} M[x{:}= N] : A$.

15.5.6. Show that

$$M \text{ is a closed term } \Rightarrow \vdash^{\mathrm{Park}}_{\cap\top} M : \omega.$$

Later we will show the converse (Theorem 18.3.22).

15.5.7. Prove that for all types $A \in \mathbb{T}^{\mathrm{AO}}$ there is an n such that

$$\top^{n}{\to}\top \leq_{\mathrm{AO}} A.$$

15.5.8. Prove that if $(\omega\varphi), (\varphi \to \omega)$ and $(\omega \to \varphi)$ are axioms in $\mathcal{T}$, then for all M in normal form $\{x_1 : \omega, \ldots, x_n : \omega\} \vdash^{\mathcal{T}} M : \varphi$, where $\{x_1, \ldots, x_n\} \supseteq \mathrm{FV}(M)$.

15.5.9. Let $\mathcal{D} = \langle D, \cdot\rangle$ be an applicative structure, i.e. a set with an arbitrary binary operation on it. For $X, Y \subset D$ define

$$X \to Y = \{d \in D \mid \forall e \in X.d \cdot e \in Y\}.$$

Consider $(\mathcal{P}(D), \to, \subseteq, \cap, D)$, where $\mathcal{P}(D)$ is the power set of D, $\subseteq$ and $\cap$ are the usual set theoretic notions and D is the top of $\mathcal{P}(D)$. Show

- $(\mathcal{P}(\mathcal{D}), \to, \subseteq, \cap)$ is a proper type structure.
- $\mathcal{D} = \mathcal{D} \to \mathcal{D}$.
- $(\mathcal{P}(\mathcal{D}), \to, \subseteq, \cap, \mathcal{D})$ is a natural type structure.

Chapter 16

Basic Properties 16.10.2006:1032

This Chapter us on type theories but, by Remark 15.4.7, applies as well to type structures. That is, everywhere $\mathcal{T}$, TT and TT$^{\top}$ may be replaced by $\mathcal{S}$, TS and TS$^{\top}$, respectively.

Let $\mathcal{T}$ be a type theory. We derive properties of $\vdash^{\mathcal{T}}$, where $\vdash^{\mathcal{T}}$ stands for $\vdash^{\mathcal{T}}_{\cap}$ or $\vdash^{\mathcal{T}}_{\cap^{\top}}$. Whenever we need to require extra properties about $\mathcal{T}$, this will be stated explicitly. Often $\mathcal{T}$ will be one of the theories from Figure 15.2.

The properties that will be studied are inversion theorems that will make it possible to predict when statements

$$\Gamma \vdash^{\mathcal{T}} M : A \tag{1}$$

are derivable, in particular from what other statments. This will be done in Section 16.1. Building upon this, in Section 16.2 conditions are given when type assignment statements remain valid after reducing or expanding the M according to $\boldsymbol{\beta}$ or $\boldsymbol{\eta}$-rules.

16.1. Inversion theorems

In the style of Coppo et al. [1984] and Alessi et al. [2003], [2005] we shall isolate special properties which allow to 'reverse' some of the rules of the type assignment system $\vdash^{\mathcal{T}}_{\cap}$, thereby achieving some form of 'generation' and 'inversion' properties. These state necessary and sufficient conditions when an assertion $\Gamma \vdash^{\mathcal{T}} M : A$ holds depending on the form of M and A, see Theorems 16.1.1 and 16.1.10.

16.1.1. Theorem (Inversion Theorem I). *If $\vdash$ is $\vdash^{\mathcal{T}}_{\cap}$, then the following statements hold unconditionally; if it is $\vdash^{\mathcal{T}}_{\cap^{\top}}$, then they hold under the assumption that*

$A \neq \top$ in (i) and (ii).

$$
\begin{array}{lrcl}
\text{(i)} & \Gamma \vdash x : A & \Leftrightarrow & \Gamma(x) \leq A. \\
\text{(ii)} & \Gamma \vdash MN : A & \Leftrightarrow & \exists k \geq 1\, \exists B_1, \ldots, B_k, C_1, \ldots, C_k \\
 & & & [C_1 \cap \ldots \cap C_k \leq A \;\&\; \forall i \in \{1, \ldots, k\} \\
 & & & \Gamma \vdash M : B_i \to C_i \;\&\; \Gamma \vdash N : B_i]. \\
\text{(iii)} & \Gamma \vdash \lambda x.M : A & \Leftrightarrow & \exists k \geq 1\, \exists B_1, \ldots, B_k, C_1, \ldots, C_k \\
 & & & [(B_1 \to C_1) \cap \ldots \cap (B_k \to C_k) \leq A \\
 & & & \;\&\; \forall i \in \{1, \ldots, k\}.\Gamma, x{:}B_i \vdash M : C_i].
\end{array}
$$

PROOF. We only treat ($\Rightarrow$) in (i)-(iii), as ($\Leftarrow$) is trivial. Let first $\vdash$ be $\vdash^{\mathcal{T}}_{\cap}$.

(i) By induction on derivations. We reason according which axiom or rule has been used in the last step. Only axiom (Ax), and rules ($\cap$I), ($\leq$) could have been applied. In the first case one has $\Gamma(x) \equiv A$. In the other two cases the induction hypothesis applies.

(ii) By induction on derivations. By assumption on A and the shape of the term the last applied step has to be rule ($\to$E), ($\leq$) or ($\cap$I). In the first case the last applied rule is

$$(\to\mathrm{E}) \quad \frac{\Gamma \vdash M : D \to A \quad \Gamma \vdash N : D}{\Gamma \vdash MN : A}.$$

We can take $k = 1$ and $C_1 \equiv A$ and $B_1 \equiv D$. In the second case the last rule applied is

$$(\leq) \quad \frac{\Gamma \vdash MN : B \quad B \leq A}{\Gamma \vdash MN : A}$$

and the induction hypothesis applies. In the last case $A \equiv A_1 \cap A_2$ and the last applied rule is

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

By the induction hypothesis there are B_i, C_i, D_j, E_j, with $1 \leq i \leq k,\ 1 \leq j \leq k'$, such that

$$
\begin{array}{ll}
\Gamma \vdash M : B_i \to C_i, & \Gamma \vdash N : B_i, \\
\Gamma \vdash M : D_j \to E_j, & \Gamma \vdash N : D_j, \\
C_1 \cap \ldots \cap C_k \leq A_1, & E_1 \cap \ldots \cap E_{k'} \leq A_2.
\end{array}
$$

Hence we are done, as $C_1 \cap \ldots \cap C_k \cap E_1 \cap \ldots \cap E_{k'} \leq A$.

(iii) Again by induction on derivations. We only treat the case $A \equiv A_1 \cap A_2$ and the last applied rule is ($\cap$I):

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

By the induction hypothesis there are B_i, C_i, D_j, E_j with $1 \leq i \leq k,\ 1 \leq j \leq k'$ such that

$$
\begin{array}{ll}
\Gamma, x{:}B_i \vdash M : C_i, & (B_1 \to C_1) \cap \ldots \cap (B_k \to C_k) \leq A_1, \\
\Gamma, x{:}D_j \vdash M : E_j, & (D_1 \to E_1) \cap \ldots \cap (D_{k'} \to E_{k'}) \leq A_2.
\end{array}
$$

We are done, since $(B_1 \to C_1) \cap \ldots \cap (B_k \to C_k) \cap (D_1 \to E_1) \cap \ldots \cap (D_{k'} \to E_{k'}) \leq A$.

Now we prove ($\Rightarrow$) in (i)-(iii) for $\lambda^{\mathcal{T}}_{\cap\top}$.

(i) The condition $A \neq \top$ implies that axiom ($\top$ universal) cannot have been used in the last step. Hence the reasoning above suffices.

(ii), (iii) The only interesting rule is ($\cap$I). Condition $A \neq \top$ implies that we cannot have $A_1 = A_2 = \top$. In case $A_1 \neq \top$ and $A_2 \neq \top$ the result follows as above. The other cases are more easy. ■

Notice that as a consequence of this theorem the *subformula property* holds for all $\lambda^{\mathcal{T}}_{\cap(\top)}$.

16.1.2. Corollary (Subformula property). *Assume $\Gamma \vdash^{\mathcal{T}}_{\cap(\top)} M : A$ and let N be a subterm of M. Then N is typable in an extension $\Gamma^{+} = \Gamma, x_1{:}B_1, \ldots, x_n{:}B_n$ in which also the variables $\{x_1, \ldots, x_n\} = \mathrm{FV}(N) - \mathrm{FV}(M)$ get a type assigned.*

Proof. If we have rule ($\top$-universal) the statement is trivial. Otherwise if N is a subterm of M, then we can write $M \equiv C[N]$. The statement is proved by induction on the structure of $C[\;\;]$. ■

16.1.3. Proposition. *We have for fresh y ($\notin \mathrm{dom}(\Gamma)$) the following.*

$$\begin{array}{l}\exists B\,[\Gamma \vdash N : B \;\&\; \Gamma \vdash M[x{:}= N] : A] \;\Rightarrow \\ \qquad \exists B\,[\Gamma \vdash N : B \;\&\; \Gamma, y{:}B \vdash M[x{:}= y] : A].\end{array}$$

Proof. By induction on the structure of M. ■

Under some conditions (that will hold for many TTs, notably the ones introduced in Section 15.1), the Inversion Theorem can be restated in a more memorable form. This will be done in Theorem 16.1.10.

16.1.4. Definition. $\mathcal{T}$ is called *$\boldsymbol{\beta}$-sound* if

$\forall k{\geq}1 \forall A_1, \ldots, A_k, B_1, \ldots, B_k, C, D.$

$$\left.\begin{array}{l}(A_1{\to}B_1) \cap \ldots \cap (A_k{\to}B_k) \leq (C{\to}D) \;\&\; D \neq \top \;\Rightarrow \\ C \leq A_{i_1} \cap \ldots \cap A_{i_p} \;\&\; B_{i_1} \cap \ldots \cap B_{i_p} \leq D, \\ \text{for some } p \geq 1 \text{ and } 1 \leq i_1, \ldots, i_p \leq k.\end{array}\right\} \qquad (*)$$

This definition immediately translates to type structures. The notion of $\boldsymbol{\beta}$-soundness is introduced to prove invertibility of the rule ($\to$I), which is important for the next section.

16.1.5. Lemma. *Let $\mathcal{T}$ satisfy ($\top$) and ($\top\!\to$). Suppose moreover that $\mathcal{T}$ is $\boldsymbol{\beta}$-sound. Then for all A, B*

$$A{\to}B = \top \;\Leftrightarrow\; B = \top.$$

Proof. ($\Rightarrow$) $\top{\to}\top \leq \top = A{\to}B$, by assumption; hence $\top \leq B$ ($\leq \top$), by $\boldsymbol{\beta}$-soundness. ($\Leftarrow$) By rule ($\top\!\to$). ■

Let $\mathcal{T}$ be $\boldsymbol{\beta}$-sound. Then $A{\to}B \leq A'{\to}B' \;\Rightarrow\; A' \leq A \;\;\&\;\; B \leq B'$ if B' is not the top element (but not in general).

In 16.1.6-16.1.8 we will show that all $\mathcal{T}$'s of Figures 15.2 are $\boldsymbol{\beta}$-sound. In these items ψ, ψ_i denote atoms.

16.1.6. Remark. Note that in a TT every type A can be written uniquely, apart from the order, as

$$A \equiv \alpha_1 \cap \ldots \cap \alpha_n \cap (B_1{\to}C_1) \cap \ldots \cap (B_k{\to}C_k) \qquad (+),$$

i.e. an intersection of atoms ($\alpha_i \in \mathbb{A}$) and arrow types.

For some of our $\mathcal{T}$ the shape (+) in Remark 16.1.6 can be simplified.

16.1.7. Definition. For the type theories $\mathcal{T}$ of Figure 15.2 we define for each $A \in \mathbb{T}^{\mathcal{T}}$ its *canonical form*, notation $\mathrm{cf}(A)$, as follows.

(i) If $\mathcal{T} \in \{\mathrm{BCD}, \mathrm{AO}, \mathrm{Plotkin}, \mathrm{Engeler}, \mathrm{CDS}, \mathrm{CDV}, \mathrm{CD}\}$, then

$$\mathrm{cf}(A) \equiv A.$$

(ii) If $\mathcal{T} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{HR}, \mathrm{DHM}, \mathrm{HL}\}$ then the definition is by induction on A. For an atom α the canonical form $\mathrm{cf}(\alpha)$ depends on the type theory in question; moreover the mapping cf preserves $\to, \cap$ and $\top$.

System $\mathcal{T}$	A	$\mathrm{cf}(A)$
Scott	ω	$\top{\to}\omega$
Park	ω	$\omega{\to}\omega$
CDZ, HL	ω	$\varphi{\to}\omega$
	φ	$\omega{\to}\varphi$
HR	ω	$\varphi{\to}\omega$
	φ	$(\omega{\to}\omega) \cap (\varphi{\to}\varphi)$
DHM	φ	$\omega{\to}\varphi$
	ω	$\top{\to}\varphi$
All systems except HL	$\top$	$\top$
All systems	$B{\to}C$	$B{\to}C$
All systems	$B \cap C$	$\mathrm{cf}(B) \cap \mathrm{cf}(C)$

16.1.8. Theorem. *All theories $\mathcal{T}$ of Figure 15.2 are $\boldsymbol{\beta}$-sound.*

Proof. We prove the following stronger statement (induction loading). Let

$$\begin{array}{rcl} A & \leq & A', \\ \mathrm{cf}(A) & \equiv & \alpha_1 \cap \ldots \cap \alpha_n \cap (B_1{\to}C_1) \cap \ldots \cap (B_k{\to}C_k), \\ \mathrm{cf}(A') & \equiv & \alpha'_1 \cap \ldots \cap \alpha'_{n'} \cap (B'_1{\to}C'_1) \cap \ldots \cap (B'_{k'}{\to}C'_{k'}). \end{array}$$

Then

$$\forall j \in \{1, k'\}.[C_{j'} \neq \top \;\Rightarrow$$
$$\exists p{\geq}1 \exists i_1, \ldots i_p \in \{1, k\}.[B'_j \leq B_{i_1} \cap \ldots \cap B_{i_p} \;\&\; C_{i_1} \cap \ldots \cap C_{i_p} \leq C'_j]].$$

The proof of the statement is by induction on the generation of $A \leq A'$. From it $\boldsymbol{\beta}$-soundness follows easily. ∎

16.1.9. Remark. From the Theorem it follows immediately that for the compatible theories of Fig. 15.2 the corresponding type structures are $\boldsymbol{\beta}$-sound.

16.1.10. Theorem (Inversion Theorem II). *Of the following properties (i) holds in general, (ii) provided that $\mathcal{T}$ is proper and $A \neq \top$ if $\vdash$ is $\vdash^{\mathcal{T}}_{\cap\top}$ and (iii) provided that $\mathcal{T}$ is $\boldsymbol{\beta}$-sound.*

$$\begin{array}{rrcl} \text{(i)} & \Gamma, x{:}A \vdash x : B & \Leftrightarrow & A \leq B. \\ \text{(ii)} & \Gamma \vdash (MN) : A & \Leftrightarrow & \exists B\,[\Gamma \vdash M : (B{\to}A)\ \&\ \Gamma \vdash N : B]. \\ \text{(iii)} & \Gamma \vdash (\lambda x.M) : (B{\to}C) & \Leftrightarrow & \Gamma, x{:}B \vdash M : C. \end{array}$$

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

(i) If $B \neq \top$, then the conclusion follows from Theorem 16.1.1(i). If $B = \top$, then the conclusion holds trivially.

(ii) Suppose $\Gamma \vdash MN : A$. Then by Theorem 16.1.1(ii) there are $B_1, \ldots, B_k$, $C_1, \ldots, C_k$, with $k \geq 1$, such that $C_1 \cap \ldots \cap C_k \leq A$, $\Gamma \vdash M : B_i{\to}C_i$ and $\Gamma \vdash N : B_i$ for $1 \leq i \leq k$. Hence $\Gamma \vdash N : B_1 \cap \ldots \cap B_k$ and

$$\begin{array}{rl} \Gamma \vdash M : & (B_1{\to}C_1) \cap \ldots \cap (B_k{\to}C_k) \\ & \leq (B_1 \cap \ldots \cap B_k){\to}(C_1 \cap \ldots \cap C_k) \\ & \leq (B_1 \cap \ldots \cap B_k){\to}A, \end{array}$$

by Lemma 15.1.13. So we can take $B \equiv (B_1 \cap \ldots \cap B_k)$.

(iii) Suppose $\Gamma \vdash (\lambda x.M) : (B{\to}C)$. Then Theorem 16.1.1(iii) applies and we have for some $k \geq 1$ and $B_1, \ldots, B_k, C_1, \ldots, C_k$

$$\begin{array}{r} (B_1{\to}C_1) \cap \ldots \cap (B_k{\to}C_k) \leq B{\to}C, \\ \Gamma, x{:}B_i \vdash M : C_i \text{ for all } i. \end{array}$$

If $C = \top$, then the assertion holds trivially, so let $C \neq \top$. Then by $\boldsymbol{\beta}$-soundness there are $1 \leq i_1, \ldots, i_p \leq k$, $p \geq 1$ such that

$$\begin{array}{c} B \leq B_{i_1} \cap \ldots \cap B_{i_p}, \\ C_{i_1} \cap \ldots \cap C_{i_p} \leq C. \end{array}$$

Applying ($\leq$-L) we get

$$\begin{array}{c} \Gamma, x{:}B \vdash M : C_{i_j},\ 1 \leq j \leq p, \\ \Gamma, x{:}B \vdash M : C_{i_1} \cap \ldots \cap C_{i_p} \leq C. \ \blacksquare \end{array}$$

We give a simple example which shows that in general rule ($\to$E) cannot be reversed, i.e. that if $\Gamma \vdash MN : B$, then it is not always true that there exists A such that $\Gamma \vdash M : A{\to}B$ and $\Gamma \vdash N : A$.

16.1.11. Example. Let $\mathcal{T}$ = Engeler, one of the intersection type theories of Figure 15.2. Let $\Gamma = \{x{:}(\varphi_0{\to}\varphi_1) \cap (\varphi_2{\to}\varphi_3), y{:}(\varphi_0 \cap \varphi_2)\}$. Then one has $\Gamma \vdash^{\mathcal{T}}_{\cap\top} xy : \varphi_1 \cap \varphi_3$. Nevertheless, it is not possible to find a type B such that $\Gamma \vdash^{\mathcal{T}}_{\cap\top} x : B{\to}(\varphi_1 \cap \varphi_3)$ and $\Gamma \vdash^{\mathcal{T}}_{\cap\top} y : B$. See Exercise ??.

16.1.12. Remark. In general

$$\Gamma \vdash^{\mathcal{T}} (\lambda x.M) : A \not\Rightarrow \exists B, C.A = (B \to C) \ \&\ \Gamma, x{:}B \vdash^{\mathcal{T}} M : C.$$

A counterexample is $\vdash^{\mathrm{BCD}} \mathsf{I} : (\alpha_1 \to \alpha_1) \cap (\alpha_2 \to \alpha_2)$, with α_1, α_2 atomic.

16.1.13. Proposition. *For $\mathcal{T} \in \{$Scott, Park, CDZ, HR, DHM, BCD, AO$\}$ the properties (i), (ii) and (iii) of Theorem 16.1.10 hold for $\vdash^{\mathcal{T}}_{\cap\top}$, provided that in (ii) $A \neq \top$ for $\mathcal{T} = \mathrm{AO}$. For $\mathcal{T} \in \{\mathrm{HL}, \mathrm{CDV}\}$ the properties hold unconditionally for $\vdash^{\mathcal{T}}_{\cap}$.*

Proof. For these $\mathcal{T}$ Theorem 16.1.10 applies since they are proper and $\boldsymbol{\beta}$-sound (by Theorem 16.1.8). Moreover, by axiom $(\to\top)$ we have $\Gamma \vdash^{\mathcal{T}}_{\cap\top} M : \top \to \top$ for all Γ, M, hence we do not need to assume $A \neq \top$ for $\mathcal{T} \in \{$Scott, Park, CDZ, HR, DHM, BCD$\}$. ■

Results for type structures

Now let $\mathcal{S}$ be a type structure. Again $\vdash$ stands for $\vdash^{\mathcal{S}}_{\cap}$ or $\vdash^{\mathcal{S}}_{\cap\top}$.

16.1.14. Theorem (Inversion Theorem I). *If $\vdash$ is $\vdash^{\mathcal{S}}_{\cap}$, then the following statements hold unconditionally; if it is $\vdash^{\mathcal{S}}_{\cap\top}$, then they hold under the assumption that $A \neq \top$ in (i) and (ii).*

$$\begin{array}{lrcl}
\text{(i)} & \Gamma \vdash x : A & \Leftrightarrow & \Gamma(x) \leq A. \\
\text{(ii)} & \Gamma \vdash MN : A & \Leftrightarrow & \exists k \geq 1\, \exists B_1, \ldots, B_k, C_1, \ldots, C_k \\
 & & & [C_1 \cap \ldots \cap C_k \leq A \ \&\ \forall i \in \{1, \ldots, k\} \\
 & & & \Gamma \vdash M : B_i \to C_i \ \&\ \Gamma \vdash N : B_i]. \\
\text{(iii)} & \Gamma \vdash \lambda x.M : A & \Leftrightarrow & \exists k \geq 1\, \exists B_1, \ldots, B_k, C_1, \ldots, C_k \\
 & & & [(B_1 \to C_1) \cap \ldots \cap (B_k \to C_k) \leq A \\
 & & & \ \&\ \forall i \in \{1, \ldots, k\}.\Gamma, x{:}B_i \vdash M : C_i].
\end{array}$$

Proof. As for Theorem 16.1.1. An alternative proof is to use Propositions 15.3.6 and 15.3.5 and to apply Theorem 16.1.1. ■

16.1.15. Theorem (Inversion Theorem II). *Of the following properties (i) holds in general, (ii) provided that $\mathcal{S}$ is proper and $A \neq \top$ if $\vdash$ is $\vdash^{\mathcal{S}}_{\cap\top}$ and (iii) provided that $\mathcal{S}$ is $\boldsymbol{\beta}$-sound.*

$$\begin{array}{lrcl}
\text{(i)} & \Gamma, x{:}A \vdash x : B & \Leftrightarrow & A \leq B. \\
\text{(ii)} & \Gamma \vdash (MN) : A & \Leftrightarrow & \exists B\, [\Gamma \vdash M : (B \to A) \ \&\ \Gamma \vdash N : B]. \\
\text{(iii)} & \Gamma \vdash (\lambda x.M) : (B \to C) & \Leftrightarrow & \Gamma, x{:}B \vdash M : C.
\end{array}$$

Proof. Similarly. ■

Proposition 16.1.13 immediately gives the following result.

16.1.16. Proposition. *For $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{HR}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}\}$ considered as type structures the properties (i), (ii) and (iii) of Theorem 16.1.15 hold for $\lambda^{\mathcal{S}}_{\cap\top}$ provided that in (ii) $A \neq \top$ for $\mathcal{S} = \mathrm{AO}$. For $\mathcal{S} \in \{\mathrm{HL}, \mathrm{CDV}\}$ they hold unconditionally for $\lambda^{\mathcal{S}}_{\cap}$.*

Comment: Because of the Remark at the beginning of this Chapter.

16.2. Subject reduction and expansion

Various subject reduction and expansion properties are proved, for the classical $\boldsymbol{\beta}$, $\boldsymbol{\beta}\mathsf{I}$ and $\boldsymbol{\eta}$ notions of reduction. Other results can be found in Alessi et al. [2003], Alessi et al. [2006]. We consider the following rules.

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

$$(R\text{-}exp) \quad \frac{M\,{}_R\!\leftarrow N \quad \Gamma \vdash M : A}{\Gamma \vdash N : A}$$

where R is a notion of reduction, notably $\boldsymbol{\beta}$-, $\boldsymbol{\beta}\mathsf{I}$, or $\boldsymbol{\eta}$-reduction. If one of these rules holds in $\lambda_{\cap(\top)}^{\mathcal{T}}$, we write $\lambda_{\cap(\top)}^{\mathcal{T}} \models (R\text{-}\{exp, red\})$, respectively. If both hold we write $\lambda_{\cap(\top)}^{\mathcal{T}} \models (R\text{-}cnv)$. These properties will be crucial in Chapters 17, 18 and 19, where we will discuss (untyped) λ-models induced by these systems.

Recall that $(\lambda x.M)N$ is a $\boldsymbol{\beta}\mathsf{I}$-*redex* if $x \in \mathrm{FV}(M)$, Curry and Feys [1958].

$\boldsymbol{\beta}$-conversion

We first investigate when $\lambda_{\cap(\top)}^{\mathcal{T}} \models (\boldsymbol{\beta}(\mathsf{I})\text{-}red)$.

16.2.1. Proposition. (i) $\lambda_{\cap(\top)}^{\mathcal{T}} \models (\boldsymbol{\beta}\mathsf{I}\text{-}red) \iff$

$$[\Gamma \vdash^{\mathcal{T}} (\lambda x.M) : (B{\to}A) \mathbin{\&} x \in \mathrm{FV}(M) \;\Rightarrow\; \Gamma, x{:}B \vdash^{\mathcal{T}} M : A].$$

(ii) $\lambda_{\cap(\top)}^{\mathcal{T}} \models (\boldsymbol{\beta}\text{-}red) \iff [\Gamma \vdash^{\mathcal{T}} (\lambda x.M) : (B{\to}A) \;\Rightarrow\; \Gamma, x{:}B \vdash^{\mathcal{T}} M : A]$.

Proof. (i) ($\Rightarrow$) Assume $\Gamma \vdash \lambda x.M : B{\to}A$ & $x \in \mathrm{FV}(M)$, which implies $\Gamma, y{:}B \vdash (\lambda x.M)y : A$, by weakening and rule $({\to}\mathrm{E})$ for a fresh y. Now rule $(\boldsymbol{\beta}\mathsf{I}\text{-}red)$ gives us $\Gamma, y{:}B \vdash M[x{:=}y] : A$. Hence $\Gamma, x{:}B \vdash M : A$.

($\Leftarrow$) Suppose $\Gamma \vdash (\lambda x.M)N : A$ & $x \in \mathrm{FV}(M)$, in order to show that $\Gamma \vdash M[x{:=}N] : A$. We may assume $A \neq \top$. Then Theorem 16.1.1(ii) implies $\Gamma \vdash \lambda x.M : B_i{\to}C_i$, $\Gamma \vdash N : B_i$ and $C_1 \cap \ldots \cap C_k \leq A$, for some $B_1, \ldots, B_k, C_1, \ldots, C_k$. By assumption $\Gamma, x{:}B_i \vdash M : C_i$. Hence by rule (*cut*), Proposition 15.2.8, one has $\Gamma \vdash M[x{:=}N] : C_i$. Therefore $\Gamma \vdash M[x{:=}N] : A$, using rules $(\cap\mathrm{I})$ and $(\leq)$.

(ii) Similarly. ∎

16.2.2. Corollary. *Let $\mathcal{T}$ be $\boldsymbol{\beta}$-sound. Then $\lambda_{\cap(\top)}^{\mathcal{T}} \models (\boldsymbol{\beta}\text{-}red)$.*

Proof. Using Theorem 16.1.10(iii). ∎

16.2.3. Corollary. (i) *Let $\mathcal{T} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{HR}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Plotkin}, \mathrm{Engeler}, \mathrm{CDS}\}$. Then $\lambda_{\cap\top}^{\mathcal{T}} \models (\boldsymbol{\beta}\text{-}red)$.*

(ii) *Let $\mathcal{T} \in \{\mathrm{HL}, \mathrm{CD}, \mathrm{CDV}\}$. Then $\lambda_{\cap}^{\mathcal{T}} \models (\boldsymbol{\beta}\text{-}red)$.*

Proof. By Corollary 16.2.2 and Theorem 16.1.8. ∎

In Definition 18.2.22 we will introduce a type theory that is not $\boldsymbol{\beta}$-sound, but nevertheless induces a type assignment system satisfying ($\boldsymbol{\beta}$-*red*).
Now we investigate when $\lambda^{\mathcal{T}}_{\cap(\top)} \models (\boldsymbol{\beta}\text{-}exp)$. As a warm-up, suppose that $\Gamma \vdash M[x{:=}N] : A$. Then we would like to conclude that N has a type, as it seems to be a subformula, and therefore $\Gamma \vdash (\lambda x.M)N : A$. There are two problems: N may occur several times in $M[x{:=}N]$, so that it has (should have) in fact several types. In the system $\lambda_{\to}$ this problem causes the failure of rule ($\boldsymbol{\beta}$-*exp*). But in the intersection type theories one has $N : B_1 \cap \ldots \cap B_k$ if $N : B_1, \ldots, N : B_k$. Therefore $(\lambda x.M)N$ has a type if $M[x{:=}N]$ has one. The second problem arises if N does not occur at all in $M[x{:=}N]$, i.e. if the redex is a $\lambda\mathsf{K}$-redex. We would like to assign as type to N the intersection over an empty sequence, i.e. the top $\top$. This makes ($\boldsymbol{\beta}$-*exp*) invalid in $\lambda^{\mathcal{T}}_{\cap}$, but valid in systems $\lambda^{\mathcal{T}}_{\cap\top}$.

16.2.4. Proposition. (i) *Suppose* $\Gamma \vdash^{\mathcal{T}} M[x{:=}N] : A$. *Then*

$$\Gamma \vdash^{\mathcal{T}} (\lambda x.M)N : A \ \Leftrightarrow\ N \textit{ is typable in context } \Gamma.$$

(ii) $\lambda^{\mathcal{T}}_{\cap(\top)} \models (\boldsymbol{\beta}\text{-}exp) \ \Leftrightarrow\ \forall \Gamma, M, N, A$

$$[\Gamma \vdash^{\mathcal{T}} M[x{:=}N] : A \ \Rightarrow\ N \textit{ is typable in context } \Gamma].$$

(iii) $\lambda^{\mathcal{T}}_{\cap(\top)} \models (\boldsymbol{\beta}\mathsf{I}\text{-}exp) \ \Leftrightarrow\ \forall \Gamma, M, N, A \textit{ with } x \in \mathrm{FV}(M)$

$$[\Gamma \vdash^{\mathcal{T}} M[x{:=}N] : A \ \Rightarrow\ N \textit{ is typable in context } \Gamma].$$

Proof. (i) ($\Rightarrow$) By Theorem 16.1.1(ii). ($\Leftarrow$) Let $\Gamma \vdash M[x{:=}N] : A$ and suppose N is typable in context Γ. By proposition 16.1.3 for some B and a fresh y one has $\Gamma \vdash N : B \;\&\; \Gamma, y{:}B \vdash M[x := y] : A$. Then $\Gamma \vdash \lambda x.M : (B{\to}A)$ and hence $\Gamma \vdash (\lambda x.M)N : A$.

(ii) ($\Rightarrow$) Assume $\Gamma \vdash M[x{:=}N] : A$. Then $\Gamma \vdash (\lambda x.M)N : A$, by ($\boldsymbol{\beta}$-*exp*), hence by (i) we are done. ($\Leftarrow$) Assume $\Gamma \vdash L' : A$, with $L \to_{\boldsymbol{\beta}} L'$. By induction on the generation of $L \to_{\boldsymbol{\beta}} L'$ we get $\Gamma \vdash L : A$ from (i) and Theorem 16.1.1.

(iii) Similar to (ii). ∎

16.2.5. Corollary. (i) $\lambda^{\mathcal{T}}_{\cap\top} \models (\boldsymbol{\beta}\text{-}exp)$.

(ii) $\lambda^{\mathcal{T}}_{\cap} \models (\boldsymbol{\beta}\mathsf{I}\text{-}exp)$.

Proof. (i) Trivial, since every term has type $\top$.

(ii) By the subformula property (Corollary 16.1.2). ∎

Now we can harvest results towards closure under $\boldsymbol{\beta}$-conversion.

16.2.6. Theorem. *Let* $\mathcal{T} \in \mathrm{TT}$ *be* $\boldsymbol{\beta}$-*sound.*

(i) *Let* $\mathcal{T} \in \mathrm{TT}^{\top}$. *Then* $\lambda^{\mathcal{T}}_{\cap\top} \models (\boldsymbol{\beta}\text{-}cnv)$.

(ii) $\lambda^{\mathcal{T}}_{\cap} \models (\boldsymbol{\beta}\mathsf{I}\text{-}cnv)$.

Proof. (i) By Corollaries 16.2.2 and 16.2.5(i).

(ii) By Corollaries 16.2.2 and 16.2.5(ii). ∎

16.2.7. Corollary. (i) *Let* $\mathcal{T} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{HR}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Plotkin}, \mathrm{Engeler}, \mathrm{CDS}\}$. *Then* $\lambda_{\cap^\top}^{\mathcal{T}} \models (\boldsymbol{\beta}\text{-}cnv)$.

(ii) *Let* $\mathcal{T} \in \{\mathrm{HL}, \mathrm{CDV}, \mathrm{CD}\}$. *Then* $\lambda_{\cap}^{\mathcal{T}} \models (\boldsymbol{\beta}\mathsf{I}\text{-}cnv)$.

Proof. (i) By Theorem 16.2.6(i).

(ii) By Theorem 16.2.6(ii). ■

$\boldsymbol{\eta}$-conversion

First we give necessary and sufficient conditions for a system $\lambda_{\cap^{(\top)}}^{\mathcal{T}}$ to satisfy the rule $(\boldsymbol{\eta}\text{-}red)$.

16.2.8. Theorem. (i) *Let* $\mathcal{T} \in \mathrm{TT}^\top$. *Then*

$$\lambda_{\cap^\top}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}red) \;\Leftrightarrow\; \mathcal{T} \text{ is natural.}$$

(ii) *Let* $\mathcal{T} \in \mathrm{TT}$. *Then*

$$\lambda_{\cap}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}red) \;\Leftrightarrow\; \mathcal{T} \text{ is proper.}$$

Proof. (i) ($\Rightarrow$) Assume $\lambda_{\cap^\top}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}red)$ towards $(\to\cap)$, $(\to)$ and $(\top\to)$.

As to $(\to\cap)$, one has

$$x{:}(A\to B) \cap (A\to C), y{:}A \vdash xy : B \cap C,$$

hence by $(\to\mathrm{I})$ it follows that $x{:}(A\to B) \cap (A\to C) \vdash \lambda y.xy : A\to(B \cap C)$. Therefore $x{:}(A\to B) \cap (A\to C) \vdash x : A\to(B \cap C)$, by $(\boldsymbol{\eta}\text{-}red)$. By Theorem 16.1.10(i) one can conclude $(A\to B) \cap (A\to C) \leq A\to(B \cap C)$.

As to $(\to)$, suppose that $A \leq B$ and $C \leq D$, in order to show $B\to C \leq A\to D$. One has $x{:}B\to C, y{:}A \vdash xy : C \leq D$, so $x{:}B\to C \vdash \lambda y.xy : A\to D$. Therefore by $(\boldsymbol{\eta}\text{-}red)$ it follows that $x{:}B\to C \vdash x : A\to D$ and we are done as before.

As to $\top \leq \top\to\top$, notice that $x{:}\top, y{:}\top \vdash xy : \top$, so we have $x{:}\top \vdash \lambda y.xy : \top\to\top$. Therefore $x{:}\top \vdash x : \top\to\top$ and again we are done.

($\Leftarrow$) Let $\mathcal{T}$ be natural. Assume that $\Gamma \vdash \lambda x.Mx : A$, with $x \notin \mathrm{FV}(M)$, in order to show $\Gamma \vdash M : A$. If $A = \top$, we are done. Otherwise,

$$\begin{array}{rcl} \Gamma \vdash \lambda x.Mx : A & \Rightarrow & \Gamma, x{:}B_i \vdash Mx : C_i, 1 \leq i \leq k, \;\& \\ & & (B_1\to C_1) \cap \ldots \cap (B_k\to C_k) \leq A, \\ & & \text{for some } B_1, \ldots, B_k, C_1, \ldots, C_k, \end{array}$$

by Theorem 16.1.1(iii). By Lemma 16.1.5 we omit the i such that $C_i = \top$. There is at least one $C_i \neq \top$, since otherwise $A \geq (B_1\to\top) \cap \ldots \cap (B_k\to\top) = \top$, again by Lemma 16.1.5, and we would have $A = \top$. Hence by Theorem 16.1.10(ii)

$$\begin{array}{lll} \Rightarrow & \Gamma, x{:}B_i \vdash M : D_i\to C_i \text{ and} & \\ & \Gamma, x{:}B_i \vdash x : D_i, & \text{for some } D_1, \ldots, D_k, \\ \Rightarrow & B_i \leq D_i, & \text{by Theorem 16.1.10(i)}, \\ \Rightarrow & \Gamma \vdash M : (B_i\to C_i), & \text{by } (\leq\text{-L}) \text{ and } (\to), \\ \Rightarrow & \Gamma \vdash M : ((B_1\to C_1) \cap \ldots \cap (B_k\to C_k)) \leq A. & \end{array}$$

(ii) Similarly, but simpler. ■

16.2.9. Corollary. (i) *Let* $\mathcal{T} \in \{\text{Scott}, \text{Park}, \text{CDZ}, \text{HR}, \text{DHM}, \text{BCD}\}$. *Then* $\lambda_{\cap^\top}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}red)$.

(ii) *Let* $\mathcal{T} \in \{\text{HL}, \text{CDV}\}$. *Then* $\lambda_{\cap}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}red)$.

In order to characterize the admissibility of rule $(\boldsymbol{\eta}\text{-}exp)$, we need to introduce a further condition on type theories. This condition is necessary and sufficient to derive from the basis $x{:}A$ the same type A for $\lambda y.xy$, as we will show in the proof of Theorem 16.2.11.

16.2.10. Definition. Let $\mathcal{T} \in \text{TT}$.
(i) $\mathcal{T}$ is called *$\boldsymbol{\eta}$-sound* iff for all A there are $k \geq 1, m_1, \ldots, m_k \geq 1$ and $B_1, \ldots, B_k$, $C_1, \ldots, C_k$,

$$\begin{pmatrix} D_{11} \ldots D_{1m_1} \\ \ldots \\ D_{k1} \ldots D_{km_k} \end{pmatrix} \text{ and } \begin{pmatrix} E_{11} \ldots E_{1m_1} \\ \ldots \\ E_{k1} \ldots E_{km_k} \end{pmatrix}$$

with

$$\begin{array}{lll} (B_1{\to}C_1) \cap \ldots \cap (B_k{\to}C_k) & \leq & A \\ \& \; A & \leq & (D_{11}{\to}E_{11}) \cap \ldots \cap (D_{1m_1}{\to}E_{1m_1}) \cap \\ & & \ldots \\ & & (D_{k1}{\to}E_{k1}) \cap \ldots \cap (D_{km_k}{\to}E_{km_k}) \\ \& \; B_i \leq D_{i1} \cap \ldots \cap D_{im_i} \;\&\; E_{i1} \cap \ldots \cap E_{im_i} \leq C_i, & & \\ \quad \text{for } 1 \leq i \leq k. & & \end{array}$$

(ii) Let $\mathcal{T} \in \text{TT}^\top$. Then $\mathcal{T}$ is called *$\boldsymbol{\eta}^\top$-sound* iff for all $A \neq \top$ at least one of the following two conditions holds.

(1) There are types $B_1, \ldots, B_n$ with $(B_1{\to}\top) \cap \ldots \cap (B_n{\to}\top) \leq A$;
(2) There are $k \geq 1$, $m_1, \ldots, m_k \geq 1$ and $B_1, \ldots, B_k$, $C_1, \ldots, C_k$,

$$\begin{pmatrix} D_{11} \ldots D_{1m_1} \\ \ldots \\ D_{k1} \ldots D_{km_k} \end{pmatrix} \text{ and } \begin{pmatrix} E_{11} \ldots E_{1m_1} \\ \ldots \\ E_{k1} \ldots E_{km_k} \end{pmatrix}$$

with

$$\begin{array}{lll} (B_1{\to}C_1) \cap \ldots \cap (B_k{\to}C_k) \cap & & \\ \cap (B_{k+1}{\to}\top) \cap \ldots \cap (B_n{\to}\top) & \leq & A \\ \& \; A & \leq & (D_{11}{\to}E_{11}) \cap \ldots \cap (D_{1m_1}{\to}E_{1m_1}) \cap \\ & & \ldots \\ & & (D_{k1}{\to}E_{k1}) \cap \ldots \cap (D_{km_k}{\to}E_{km_k}) \\ \& \; B_i \leq D_{i1} \cap \ldots \cap D_{im_i} \;\&\; E_{i1} \cap \ldots \cap E_{im_i} \leq C_i, & & \\ \quad \text{for } 1 \leq i \leq k. & & \end{array}$$

This definition immediately translates to type structures. The validity of $\boldsymbol{\eta}$-expansion can be given as follows.

16.2.11. Theorem (Characterization of $\boldsymbol{\eta}$-*exp*).

$$\begin{array}{llll} \text{(i)} & \lambda_{\cap}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}exp) & \Leftrightarrow & \mathcal{T} \textit{ is } \boldsymbol{\eta}\textit{-sound.} \\ \text{(ii)} & \lambda_{\cap^\top}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}exp) & \Leftrightarrow & \mathcal{T} \textit{ is } \boldsymbol{\eta}^\top\textit{-sound.} \end{array}$$

PROOF. (i) ($\Rightarrow$) Assume $\lambda_\cap^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-exp})$. As $x{:}A \vdash x : A$, by assumption we have $x{:}A \vdash \lambda y.xy : A$. From Theorem 16.1.1(iii) it follows that $x{:}A, y{:}B_i \vdash xy : C_i$ and $(B_1 \to C_1) \cap \ldots \cap (B_k \to C_k) \leq A$ for some B_i, C_i. By Theorem 16.1.1(ii) for each i there exist D_{ij}, E_{ij}, such that for each j one has $x{:}A, y{:}B_i \vdash x : (D_{ij} \to E_{ij})$, $x{:}A, y{:}B_i \vdash y : D_{ij}$ and $E_{i1} \cap \ldots \cap E_{im_i} \leq C_i$. Hence by Theorem 16.1.1(i) we have $A \leq (D_{ij} \to E_{ij})$ and $B_i \leq D_{ij}$ for all i and j. Therefore we obtain the condition of 16.2.10(i).

($\Leftarrow$) Suppose that $\Gamma \vdash M : A$ in order to show $\Gamma \vdash \lambda x.Mx : A$, with x fresh. By assumption A satisfies the condition of Definition 16.2.10(i).

$$
\begin{array}{l}
(B_1 \to C_1) \cap \ldots \cap (B_k \to C_k) \;\;\leq\;\; A \\
\&\;\; A \;\;\leq\;\; (D_{11} \to E_{11}) \cap \ldots \cap (D_{1m_1} \to E_{1m_1}) \cap \\
\qquad\qquad \ldots \\
\qquad\qquad (D_{k1} \to E_{k1}) \cap \ldots \cap (D_{km_k} \to E_{km_k}) \\
\&\;\; B_i \;\leq\; D_{i1} \cap \ldots \cap D_{im_i} \;\&\; E_{i1} \cap \ldots \cap E_{im_i} \;\leq\; C_i, \\
\qquad \text{for } 1 \leq i \leq k.
\end{array}
$$

By rule ($\leq$) for all i, j we have $\Gamma \vdash M : D_{ij} \to E_{ij}$ and so $\Gamma, x{:}D_{ij} \vdash Mx : E_{ij}$ by rule ($\to$E). From ($\leq$ L), ($\cap$I) and ($\leq$) we get $\Gamma, x{:}B_i \vdash Mx : C_i$ and this implies $\Gamma \vdash \lambda x.Mx : B_i \to C_i$, using rule ($\to$I). So we can conclude by ($\cap$I) and ($\leq$) that $\Gamma \vdash \lambda x.Mx : A$.

(ii) The proof is nearly the same as for (i). ($\Rightarrow$) Again we get $x{:}A, y{:}B_i \vdash xy : C_i$ and $(B_1 \to C_1) \cap \ldots \cap (B_k \to C_k) \leq A$ for some B_i, C_i. If all $C_i = \top$, then A satisfies the first condition of Definition 16.2.10(ii). Otherwise, consider the i such that $C_i \neq \top$ and reason as in the proof of ($\Rightarrow$) for (i).

($\Leftarrow$) Suppose that $\Gamma \vdash M : A$ in order to show $\Gamma \vdash \lambda x.Mx : A$, with x fresh. If A satisfies the first condition of Definition 16.2.10(ii), that is $(B_1 \to \top) \cap \ldots \cap (B_n \to \top) \leq A$, then by rule ($\top$) it follows that $\Gamma, x{:}B_i \vdash Mx : \top$, hence $\Gamma \vdash \lambda x.Mx : (B_1 \to \top) \cap \ldots \cap (B_n \to \top) \leq A$. Now let A satisfy the second condition. Then the proof is similar to that for ($\Leftarrow$) in (i). ∎

For most intersection type theories of interest the condition of $\boldsymbol{\eta}(^\top)$-soundness is deduced from the following proposition.

16.2.12. PROPOSITION. *Let $\mathcal{T} \in \mathrm{TT}$ with atoms $\mathbb{A}$ be proper.*

(i) *$\mathcal{T}$ is $\boldsymbol{\eta}$-sound* $\;\Leftrightarrow\;$ $\forall A \in \mathbb{A}\, \exists B_1, \ldots, B_k, C_1, \ldots, C_k\, \exists n \geq 1$
$A = (B_1 \to C_1) \cap \ldots \cap (B_k \to C_k)$.

(ii) *Let $\mathcal{T} \in \mathrm{TT}^\top$. Then*

$$
\begin{array}{lcl}
\mathcal{T} \textit{ is } \boldsymbol{\eta}^\top\textit{-sound} & \Leftrightarrow & \forall A \in \mathbb{A} [\top \to \top \leq A \vee \exists B_1, \ldots, B_k, C_1, \ldots, C_k \\
\exists k \geq 1 & & [(B_1 \to C_1) \cap \ldots \cap (B_k \to C_k) \cap (\top \to \top) \leq A \\
 & & \& \; A \leq (B_1 \to C_1) \cap \ldots \cap (B_k \to C_k)]].
\end{array}
$$

(iii) *Let $\mathcal{T} \in NTT^\top$. Then*

$$\mathcal{T} \textit{ is } \boldsymbol{\eta}^\top\textit{-sound} \;\Leftrightarrow\; \mathcal{T} \textit{ is } \boldsymbol{\eta}\textit{-sound.}$$

PROOF. (i) ($\Rightarrow$) Suppose $\mathcal{T}$ is $\boldsymbol{\eta}$-sound. Let $A \in \mathbb{A}$. Then A satisfies the condition of Definition 16.2.10(i), for some $B_1, \ldots, B_k$, $C_1, \ldots, C_k$, $D_{11}, \ldots, D_{1m_1}, \ldots, D_{k1}, \ldots, D_{km_1}$, $E_{11}, \ldots, E_{1m_1}, \ldots, E_{k1}, \ldots, E_{km_k}$. By ($\rightarrow\cap$) and ($\rightarrow$), using Proposition 15.1.13, it follows that

$$\begin{array}{rcl} A & \leq & (D_{11} \cap \ldots \cap D_{1m_1} \rightarrow E_{11} \cap \ldots \cap E_{1m_1}) \cap \ldots \cap \\ & & (D_{k1} \cap \ldots \cap D_{km_k} \rightarrow E_{k1} \cap \ldots \cap E_{km_k}) \\ & \leq & (B_1 \rightarrow C_1) \cap \ldots \cap (B_k \rightarrow C_k), \end{array}$$

hence $A =_{\mathcal{T}} (B_1 \rightarrow C_1) \cap \ldots \cap (B_k \rightarrow C_k)$.

($\Leftarrow$) By induction on the generation of A one can show that A satisfies the condition of $\boldsymbol{\eta}$-soundness. The case $A_1 \rightarrow A_2$ is trivial and the case $A \equiv A_1 \cap A_2$ follows by the induction hypothesis and Rule (mon).

(ii) Similarly. Note that $(\top \rightarrow \top) \leq (B \rightarrow \top)$ for all B.

(iii) Immediately by (ii) using rule ($\top\rightarrow$). ■

16.2.13. COROLLARY. (i) *Let* $\mathcal{T} \in \{\text{Scott}, \text{Park}, \text{CDZ}, \text{HR}, \text{DHM}, \text{AO}\}$. *Then* $\mathcal{T}$ *is* $\boldsymbol{\eta}^{\top}$*-sound.*

(ii) HL *is* $\boldsymbol{\eta}$*-sound.*

PROOF. Easy. For AO in (i) one applies (ii) of the Proposition. ■

16.2.14. COROLLARY. (i) *Let* $\mathcal{T} \in \{\text{Scott}, \text{Park}, \text{CDZ}, \text{HR}, \text{DHM}, \text{AO}\}$. *Then*

$$\lambda_{\cap^{\top}}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}exp).$$

(ii) *Let* $\mathcal{T} = \text{HL}$*, then*

$$\lambda_{\cap}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}exp).$$

PROOF. By the previous Corollary and Theorem 16.2.11. ■

Exercise 16.3.15 shows that the remaining systems of Figure 15.2 do not satisfy ($\boldsymbol{\eta}$-*exp*).

Now we can harvest results towards closure under $\boldsymbol{\eta}$-conversion.

$\mathcal{T}$	$\boldsymbol{\beta}$-red	$\boldsymbol{\beta}\mathsf{I}$-red	$\boldsymbol{\beta}$-exp	$\boldsymbol{\beta}\mathsf{I}$-exp	$\boldsymbol{\eta}$-red	$\boldsymbol{\eta}$-exp
Scott	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Park	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
CDZ	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
HR	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
DHM	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
BCD	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\not\surd$
AO	$\surd$	$\surd$	$\surd$	$\surd$	$\not\surd$	$\surd$
Plotkin	$\surd$	$\surd$	$\surd$	$\surd$	$\not\surd$	$\not\surd$
Engeler	$\surd$	$\surd$	$\surd$	$\surd$	$\not\surd$	$\not\surd$
CDS	$\surd$	$\surd$	$\surd$	$\surd$	$\not\surd$	$\not\surd$
HL	$\surd$	$\surd$	$\not\surd$	$\surd$	$\surd$	$\surd$
CDV	$\surd$	$\surd$	$\not\surd$	$\surd$	$\surd$	$\not\surd$
CD	$\surd$	$\surd$	$\not\surd$	$\surd$	$\not\surd$	$\not\surd$

Figure 16.1: Type theories versus reduction and expansion

16.2.15. Theorem. (i) *Let* $\mathcal{T} \in \mathrm{TT}^{\top}$. *Then*

$$\lambda_{\cap^\top}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}cnv) \;\Leftrightarrow\; \mathcal{T} \textit{ is natural and } \boldsymbol{\eta}^{\top}\textit{-sound.}$$

(ii) *Let* $\mathcal{T} \in \mathrm{TT}$. *Then*

$$\lambda_{\cap}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}cnv) \;\Leftrightarrow\; \mathcal{T} \textit{ is proper and } \boldsymbol{\eta}\textit{-sound.}$$

Proof. (i) By Theorems 16.2.11(ii) and 16.2.8(i).
(ii) By Theorems 16.2.11(i) and 16.2.8(ii). ∎

16.2.16. Theorem. (i) *For* $\mathcal{T} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{HR}, \mathrm{DHM}\}$ *one has*

$$\lambda_{\cap^\top}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}cnv).$$

(ii) *For* $\mathcal{T} = \mathrm{HL}$ *one has*

$$\lambda_{\cap}^{\mathcal{T}} \models (\boldsymbol{\eta}\text{-}cnv).$$

Proof. (i) By Corollaries 16.2.9(i) and 16.2.14(i).
(ii) By Corollaries 16.2.9(ii) and 16.2.14(ii). ∎

Figure 16.1 summarises the results of this section and of the exercises in the following section for the type theories of Figure 15.2. The symbol '$\surd$' stands for "the property holds" and '$\not\surd$' for "the property fails".

16.3. Exercises

16.3.1. Show that for each number $n \in \mathbb{N}$ there is a type $A_n \in \mathbb{T}^{\mathrm{CD}}$ such that for the Church numerals $\mathbf{c}_n$ one has $\Gamma \vdash_{\cap}^{\mathrm{CD}} \mathbf{c}_{n+1} : A_n$, but $\Gamma \nvdash_{\cap}^{\mathrm{CD}} \mathbf{c}_n : A_n$.

16.3.2. Show that $\mathsf{S}(\mathsf{KI})(\mathsf{II})$ and $(\lambda x.xxx)\mathsf{S}$ are typable in $\vdash_{\cap}^{\mathrm{CD}}$.

16.3.3. Derive $\vdash_{\cap^\top}^{\mathrm{CDZ}} (\lambda x.xxx)\mathsf{S} : \varphi$ and $y{:}\omega, z{:}\omega \vdash_{\cap^\top}^{\mathrm{CDZ}} (\lambda x.xxx)(\mathsf{S}yz) : \omega$.
Comment: Moved here as they need lemmas from 16.

16.3.4. Find the relation between the following three types w.r.t. $\leq_{\mathrm{CDZ}}$.

$$(\omega{\to}(\varphi{\to}\varphi){\to}\omega) \cap ((\varphi{\to}\varphi){\to}\varphi), (\omega{\to}\omega){\to}\omega \text{ and } \varphi{\to}(\omega{\to}\omega){\to}\varphi.$$

16.3.5. Using the Inversion Theorems show the following.

(i) $\nvdash_{\cap}^{\mathrm{CD}} \mathsf{1} : \alpha{\to}\alpha$, where α is any constant.

(ii) $\nvdash_{\cap}^{\mathrm{HL}} \mathsf{K} : \omega$.

(iii) $\nvdash_{\cap^\top}^{\mathrm{Scott}} \mathsf{I} : \omega$.

(iv) $\nvdash_{\cap^\top}^{\mathrm{Plotkin}} \mathsf{I}x : \omega$.

16.3.6. We say that M and M' have the same types in Γ, notation $M \sim_\Gamma M'$ if

$$\forall A\, [\Gamma \vdash M : A \;\Leftrightarrow\; \Gamma \vdash M' : A].$$

Prove that $M \sim_\Gamma M' \;\Rightarrow\; M\vec{N} \sim_\Gamma M'\vec{N}$, .

16.3.7. Show that $\mathcal{T}$ = Plotkin is $\boldsymbol{\beta}$-sound by checking that it satisfies the following stronger condition.

$$\begin{aligned}&(A_1{\to}B_1) \cap \ldots \cap (A_n{\to}B_n) \leq C{\to}D \;\Rightarrow\\ &\exists k \neq 0\, \exists i_1, \ldots, i_k. 1 \leq i_j \leq n \;\&\; C = A_{i_j} \;\&\; B_{i_1} \cap \ldots \cap B_{i_k} = D.\end{aligned}$$

16.3.8. Show that $\mathcal{T}$ = Engeler is $\boldsymbol{\beta}$-sound by checking that it satisfies the following stronger condition:

$$\begin{aligned}&(A_1{\to}B_1) \cap \ldots \cap (A_n{\to}B_n) \leq C{\to}D \& D \neq \top \;\Rightarrow\\ &\exists k \neq 0\, \exists i_1, \ldots, i_k. 1 \leq i_j \leq n \;\&\; C = A_{i_j} \;\&\; B_{i_1} \cap \ldots \cap B_{i_k} = D.\end{aligned}$$

16.3.9. Let $\mathbb{A}^{\mathcal{T}} = \{\top, \omega\}$ and $\mathcal{T}$ be defined by the axioms and rules of the theories Scott and Park together. Show that $\mathcal{T}$ is not $\boldsymbol{\beta}$-sound [Hint: show that $\top \neq \omega$].

16.3.10. Prove that Theorem 16.1.10(ii) still holds if the condition of properness is replaced by the following two conditions

$$A \leq_{\mathcal{T}} B \;\Rightarrow\; C{\to}A \leq_{\mathcal{T}} C{\to}B$$

$$(A{\to}B) \cap (C{\to}D) \leq_{\mathcal{T}} A \cap C{\to}B \cap D.$$

16.3.11. Show that the following condition

$$A \to B =_{\mathcal{T}} \top \to \top \;\Rightarrow\; B =_{\mathcal{T}} \top$$

is necessary for the admissibility of rule ($\boldsymbol{\beta}$-*red*) in $\lambda_\cap^{\mathcal{T}}$. [Hint: Use Proposition 16.2.1(ii).]

16.3.12. Remember that the systems $\lambda_\cap^{\mathsf{K}}$ and $\lambda_{\cap\top}^{\mathsf{K}^\top}$ are defined in Exercise 15.5.2.

(i) Show that rules ($\boldsymbol{\beta}$-*red*) and ($\boldsymbol{\beta}\mathsf{l}$-*exp*) are admissible in $\lambda_\cap^{\mathsf{K}}$, while ($\boldsymbol{\beta}$-*exp*) is not admissible.

(ii) Show that rules ($\boldsymbol{\beta}$-*red*) and ($\boldsymbol{\beta}$-*exp*) are admissible in $\lambda_{\cap\top}^{\mathsf{K}^\top}$.

16.3.13. (i) Show that for $\mathcal{T} \in \{\mathrm{AO}, \mathrm{Engeler}, \mathrm{Plotkin}, \mathrm{CDS}\}$ one has

$$\lambda_{\cap\top}^{\mathcal{T}} \not\models (\boldsymbol{\eta}\text{-red}).$$

(ii) Show that for $\mathcal{T} = \mathrm{CD}$ one has

$$\lambda_{\cap}^{\mathcal{T}} \not\models (\boldsymbol{\eta}\text{-red}).$$

16.3.14. Verify the following.

(i) $\boldsymbol{\eta}$-soundness implies $\boldsymbol{\eta}\top$-soundness. **Comment:** Wil: Is this interesting?

(ii) Let $\mathcal{T} \in \{\mathrm{BCD}, \mathrm{Plotkin}, \mathrm{Engeler}, \mathrm{CDS}\}$. Then $\mathcal{T}$ is not $\boldsymbol{\eta}\top$-sound.

(iii) Let $\mathcal{T} \in \{\mathrm{AO}, \mathrm{CDV}, \mathrm{CD}\}$. Then $\mathcal{T}$ is not $\boldsymbol{\eta}$-sound.

Notice that AO is $\boldsymbol{\eta}\top$-sound (Corollary 16.2.13).

16.3.15. (i) Show that for $\mathcal{T} \in \{\mathrm{BCD}, \mathrm{Engeler}, \mathrm{Plotkin}, \mathrm{CDS}\}$ one has

$$\lambda_{\cap\top}^{\mathcal{T}} \not\models (\boldsymbol{\eta}\text{-exp}).$$

(ii) Show that for $\mathcal{T} \in \{\mathrm{CDV}, \mathrm{CD}\}$ one has

$$\lambda_{\cap}^{\mathcal{T}} \not\models (\boldsymbol{\eta}\text{-exp}).$$

16.3.16. Show that rules ($\boldsymbol{\eta}$-*red*) and ($\boldsymbol{\eta}$-*exp*) are not admissible in the systems $\lambda_\cap^{\mathsf{K}}$ and $\lambda_{\cap\top}^{\mathsf{K}^\top}$ as defined in Exercises 15.5.2.

16.3.17. Let $\vdash$ denote derivability in the system obtained from the system $\lambda_\cap^{\mathrm{CDV}}$ by replacing rule $(\leq)$ by the rules $(\cap\mathrm{E})$, see Definition 15.2.5, and adding the rule

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

Show that $\Gamma \vdash_\cap^{\mathrm{CDV}} M : A \;\Leftrightarrow\; \Gamma \vdash M : A$.

16.3.18. (Barendregt et al. [1983]) Let $\vdash$ denote derivability in the system obtained from $\lambda_{\cap\top}^{\mathrm{BCD}}$ by replacing rule $(\leq)$ by the rules $(\cap\mathrm{E})$ and adding $(R\eta)$ as defined in Exercise 16.3.17. Verify that

$$\Gamma \vdash_{\cap\top}^{\mathrm{BCD}} M : A \;\Leftrightarrow\; \Gamma \vdash M : A.$$

16.3.19. Let Δ be a basis that is allowed to be infinite. We define $\Delta \vdash M : A$ iff there exists a finite basis $\Gamma \subset \Delta$ such that $\Gamma \vdash M : A$.

(i) Show that all the typability rules are derivable except possibly for ($\to$I).

(ii) Suppose dom(Δ) is the set of all the variables. Show that the rule ($\to$I) is derivable if it is reformulated as

$$\Delta_x, x{:}A \vdash M : B \;\Rightarrow\; \Delta \vdash (\lambda x.M) : (A \to B),$$

with Δ_x the result of removing any $x{:}C$ from Δ.

(iii) Reformulate and prove Propositions 15.2.8, 15.2.10, Theorems 16.1.1 and 16.1.10 for infinite bases.

16.3.20. A *multi-basis* Γ is a set of declarations, in which the requirement that

$$x{:}A, y{:}B \in \Gamma \;\Rightarrow\; x \equiv y \;\Rightarrow\; A \equiv B$$

is dropped. Let Δ be a (possibly infinite) multi-basis. We define $\Delta \vdash M : A$ iff there exists a singled (only one declaration per variable) basis $\Gamma \subset \Delta$ such that $\Gamma \vdash M : A$.

(i) Show that $x : \alpha_1, x : \alpha_2 \not\vdash^{\mathrm{CD}} x : \alpha_1 \cap \alpha_2$.

(ii) Show that $x : \alpha_1 \to \alpha_2, x : \alpha_1 \not\vdash^{\mathrm{CD}} xx : \alpha_2$.

(iii) Consider $\Delta = \{x : \alpha_1 \cap \alpha_2, x : \alpha_1\}$;
$A = \alpha_2$;
$B = (\alpha_1 \to \alpha_2 \to \alpha_3) \to \alpha_3$;
$M = \lambda y.yxx$.
Show that $\Delta, x : A \vdash^{\mathrm{CD}} M : B$, but $\Delta \not\vdash^{\mathrm{CD}} (\lambda x.M) : (A \to B)$.

(iv) We say that a multi-basis is closed under $\cap$ if for all $x \in \mathrm{dom}(\Delta)$ the set $\mathcal{X} = \Delta(x)$ is closed under $\cap$, i.e. $A, B \in \mathcal{X} \;\Rightarrow\; A \cap B \in \mathcal{X}$, up to equality of types in the TT under consideration.
Show that all the typability rules of Figures 15.4 and **??**, except for ($\to$I), are derivable for (possibly infinite) multi-bases that are closed under $\cap$.

(v) Let Δ be closed under $\cap$. We define

$$\Delta[x := X] = \{y : \Delta(y) \mid y \neq x\} \cup \{x : A \mid A \in X\}.$$

Prove that the following reformulation of ($\to$I) using principal filters is derivable

$$\frac{\Delta[x := \uparrow B] \vdash N : C}{\Delta \vdash \lambda x.N : B \to C}.$$

(vi) Prove Propositions 15.2.8, 15.2.10, Theorems 16.1.1 and 16.1.10 for (possible infinite) multi-bases reformulating the statements whenever it is necessary.

(vii) Prove that if $\Delta(x)$'s are filters then $\{A \mid \Delta \vdash x : A\} = \Delta(x)$.

16.3.21. Show that the inclusions suggested in 15.3 are strict.

Chapter 17

Type and Lambda Structures

This Chapter is on meet semi-lattices and type structures. The pre-order of type theories is not sufficient, we need a partial order.

There are many models of the untyped lambda calculus that 'live' in the category **ALG** of ω-complete algebraic lattices: posets with arbitrary sups and countably many compact elements. For example $P\omega$ and $\mathcal{D}_\infty$ are enriched **ALG** objects. Each object $\mathcal{D}$ of **ALG**, that by definition is algebraic, is determined by its compact (finite) elements $\mathcal{K}(\mathcal{D})$. On these compact elements there is a natural structure of a meet semi-lattice: for $d, e \in \mathcal{K}(\mathcal{D})$

$$d \leq_{\mathcal{K}(\mathcal{D})} e \;\Leftrightarrow\; e \sqsubseteq_{\mathcal{D}} d.$$

The reason for turning around the order is as follows. An element $d \in \mathcal{K}(\mathcal{D})$ is said to approximate an $x \in \mathcal{D}$ iff $d \sqsubseteq x$. Now if $d \sqsubseteq e$, then d approximates more elements of $\mathcal{D}$, than e. Therefore it will turn out to be natural to write $e \leq d$. Moreover, if d, e are compact, then so is $d \sqcup_{\mathcal{D}} e$. With respect to the new ordering this element is the meet

$$d \cap_{\mathcal{K}(\mathcal{D})} e.$$

In this way it is natural to relate the objects of **ALG** with those of $\mathbf{MSL}^{\top}$, the category of meet semi-lattices with top. In fact $\mathcal{D}$ can be reconstructed from $\mathcal{K}(\mathcal{D})$ by considering *filters* of compact elements: $\mathcal{D} \cong \mathcal{F}^{\mathcal{K}(\mathcal{D})}$. It turns out that a subcategory $\mathbf{ALG}_a$ of **ALG**, which has the same objects but less morphisms, is equivalent with $\mathbf{MSL}^{\top}$.

So far so good, but the obtained structures are too poor for interpreting lambda calculus. Therefore the story is repeated for so-called *algebraically applicative structures* (*zip structures*), objects $\mathcal{D}$ of **ALG** enriched with a kind of 'pairing operator' Z on $\mathcal{K}(\mathcal{D})$. The category obtained in this way is called **ZS**. Using this pairing operator one can define application and abstraction (not yet satisfying the $\boldsymbol{\beta}$-rule) on $\mathcal{D}$. Corresponding to **ZS** there is now the category $\mathbf{TS}^{\top}$ of intersection type structures with top, that consists of meet semi-lattices with top enriched with an operator $\rightarrow$.

Finally we will show that zip structures correspond with lambda structures, i.e. triples $\langle \mathcal{D}, F, G\rangle$, where $\mathcal{D} \in \mathbf{ALG}$, $F : [\mathcal{D}\rightarrow[\mathcal{D}\rightarrow\mathcal{D}]]$ and $G : [[\mathcal{D}\rightarrow\mathcal{D}]\rightarrow\mathcal{D}]$, such that $\langle G, F\rangle$ is a Galois connection.

Each time we also consider the so-called strict case, in which the objects of the category may miss a top element or where the top element is treated in a special way.

In Section 17.1 we introduce the categories $\mathbf{ALG}$, $\mathbf{ALG}_a$ and $\mathbf{ALG}_a^s$ (strict case) of complete lattices, $\mathbf{ZS}$ and $\mathbf{ZS}^s$ of zip structures and lazy ($\mathbf{LZS}$), natural ($\mathbf{NZS}$) and proper strict ($\mathbf{PZS}^s$) subcategories of the latter. We also derive some basic lemmas.

In Section 17.2 we show the equivalences (of the categories) $\mathbf{MSL}^\top \cong \mathbf{ALG}_a$ and of $\mathbf{MSL} \cong \mathbf{ALG}_a^s$.

In Section 17.3 we treat the equivalences $\mathbf{TS}^\top \cong \mathbf{ZS}$ and also of the lazy and natural subcategories. After that the strict cases are treated.

Finally in Section 17.4 we show that zip structures correspond with lambda structures. For the general case of zip structures the corresponcence is not perfect, but it is in the form of an equivalence between categories, for the lazy, natural and proper strict zip structures.

Summarizing, there are three groups of categories of structures: the type, lambda and zip structures. The type structures are expansions of meet semi-lattices with the extra operator $\to$; the lambda and zip structures are expansions of algebraic lattices D with respectively the extra structure of a pair $\langle F, G\rangle$ with $F : D{\to}D{\to}D$ and $G : [D{\to}D]{\to}D$ or a map Z merging ('zipping') two compact elements into one. Within each group there are five or three relevant items, explained above. The categories equivalences are displayed in Figures 17.1, 17.2.

$$\begin{tikzcd}[arrows=-]
\text{meet semi-lattices} & \mathbf{MSL}^\top \arrow[r, "\cong"] \arrow[d] & \mathbf{ALG} \arrow[d] & \text{algebraic lattices} \\
& \mathbf{MSL} \arrow[r, "\cong"] & \mathbf{ALG}^s &
\end{tikzcd}$$

Figure 17.1: Equivalences proved in Section 17.2

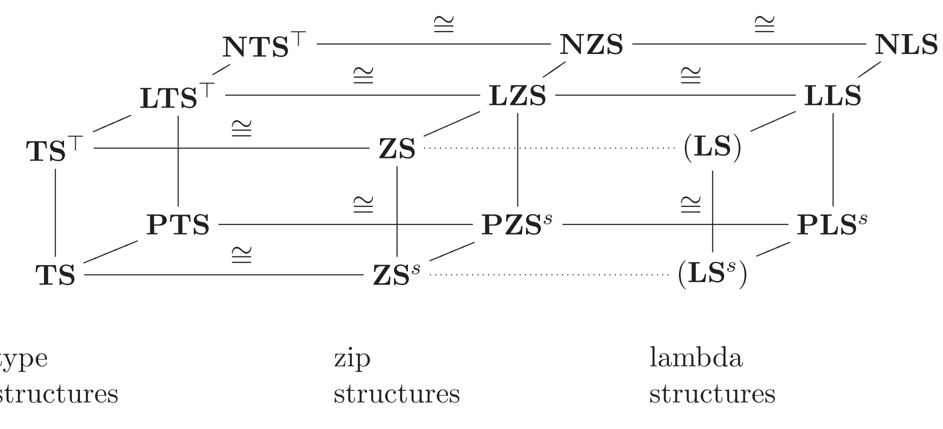

Figure 17.2: Equivalences proved in Sections 17.3 and 17.4

17.1. Algebraic lattices and zip structures

Algebraic lattices

Comment:

- for sure all the content of this subsection is not new, so references are mandatory. We do not know this. Can you put pointers? Me too I am not very familiar, and the references become old very quickly today, I propose to do this work when you will establish a FIXED publication date.
- I do not have a good suggestion for the top of D, both Ω and $\top_D$ were used, I put the macro topD. Is OK It is not OK to call the top of a lattice exactly as the top type, please change back the macro!
- all new macros are in the file partIII.tex, some of them are red (by ma) Some I changed, marking them with 'hb'. For the ones I agreed I removed the 'ma'
- indexes are denoted either by I or by $\mathcal{I}$, you must choose to give some meaning to this notational difference (I finite and $\mathcal{I}$ possibly infinite?), state and respect it, or use the same notation.

The following has already been given in Definition 14.2.1, but now we treat in in greater detail.

Comment: D is a type, I propose $\mathtt{D}$ (macro dD)

17.1.1. Definition. (i) A *complete lattice* is a poset $\mathcal{D} = (\mathtt{D}, \sqsubseteq)$ such that for arbitrary $X \subseteq \mathtt{D}$ the supremum $\bigsqcup X \in \mathtt{D}$ exists. Then one has also a *top* element $\top_{\mathcal{D}} = \bigsqcup \mathtt{D}$, a *bottom* element $\bot_{\mathcal{D}} = \bigsqcup \emptyset$, arbitrary *infima*

$$\bigsqcap X = \bigsqcup \{z \mid \forall x \in X. z \sqsubseteq x\}$$

and the *sup* and *inf* of two elements

$$x \sqcup y = \bigsqcup\{x, y\}, \quad x \sqcap y = \bigsqcap\{x, y\}.$$

(ii) A subset $Z \subseteq \mathtt{D}$ is called *directed* if Z is non-empty and

$$\forall x, y \in Z\ \exists z \in Z. x \sqsubseteq z \ \&\ y \sqsubseteq z.$$

(iii) An element $d \in \mathtt{D}$ is called *compact* (also sometimes called *finite* in the literature) if for every directed $Z \subseteq \mathtt{D}$ one has

$$d \sqsubseteq \bigsqcup Z \ \Rightarrow\ \exists z \in Z. d \sqsubseteq z.$$

Note that if d, e are compact, then so is $d \sqcup e$[1].

(iv) $\mathcal{K}(\mathcal{D}) = \{d \in \mathtt{D} \mid d \text{ is compact}\}$.

[1] In general it is not true that if $d \sqsubseteq e \in \mathcal{K}(\mathcal{D})$, then $d \in \mathcal{K}(\mathcal{D})$; take for example $\omega + 1$ in the ordinal $\omega + \omega = \{0, 1, 2, \ldots\ \omega, \omega + 1, \omega + 2, \ldots\}$. It is compact, but $\omega\ \ (\sqsubseteq \omega + 1)$ is not.

(v) $\mathcal{K}^s(\mathcal{D}) = \mathcal{K}(\mathcal{D}) - \{\bot_{\mathcal{D}}\}$.
(vi) $\mathcal{D}$ is called an *algebraic lattice* if

$$\forall x \in \mathtt{D}. x = \bigsqcup\{e \in \mathcal{K}(\mathcal{D}) \mid e \sqsubseteq x\}.$$

$\mathcal{D}$ is called an *ω-algebraic lattice* if moreover $\mathcal{K}(\mathcal{D})$ is countable (finite or countably infinite).

Instead of $d \in \mathtt{D}$ or $X \subseteq \mathtt{D}$ we often write $d \in \mathcal{D}$ or $X \subseteq \mathcal{D}$, respectively. When useful we will decorate $\sqsubseteq, \bigsqcup, \bigsqcap, \bot, \top, \sqcup$ and $\sqcap$ with $\mathcal{D}$, e.g. $\sqsubseteq_{\mathcal{D}}$ etcetera.

The following connects the notion of a compact element to the notion of compact subset of a topological space.

17.1.2. Lemma. *Let $\mathcal{D}$ be a complete lattice. Then $d \in \mathcal{D}$ is compact iff*

$$\forall Z \subseteq \mathcal{D}.[d \sqsubseteq Z \;\Rightarrow\; \exists Z_0 \subseteq Z.[Z_0 \textit{ is finite } \;\&\; d \sqsubseteq \bigsqcup Z_0]].$$

Proof. ($\Rightarrow$) Suppose $d \in \mathcal{D}$ is compact. Given $Z \subseteq \mathcal{D}$, let

$$Z^+ = \{\bigsqcup Z_0 \mid Z_0 \subseteq Z \;\&\; Z_0 \text{ finite}\}.$$

Then $Z \subseteq Z^+, \bigsqcup Z_0 = \bigsqcup Z$ and Z^+ is directed. Hence

$$\begin{array}{rcl} d \sqsubseteq \bigsqcup Z & \Rightarrow & d \sqsubseteq \bigsqcup Z^+ \\ & \Rightarrow & \exists z^+ \in Z^+. d \sqsubseteq z^+ \\ & \Rightarrow & \exists Z_0 \subseteq Z. d \sqsubseteq \bigsqcup Z_0 \;\&\; Z_0 \text{ is finite.} \end{array}$$

($\Leftarrow$) Suppose $d \sqsubseteq \bigsqcup Z$ with $Z \subseteq \mathcal{D}$ directed. By the condition $d \sqsubseteq \bigsqcup Z_0$ for some finite $Z_0 \subseteq Z$. If Z_0 is non-empty, then by the directedness of Z there exists a $z \in Z$ such that $z \sqsupseteq \bigsqcup Z_0 \sqsupseteq d$. If Z_0 is empty, then $d = \bot$ and we can take an arbitrary element z in the non-empty Z satisfying $d \sqsubseteq z$. ∎

17.1.3. Notation. Let $\mathcal{D}$ be an ω-algebraic lattice. For $x \in \mathcal{D}$, write

$$K(x) = \{d \in \mathcal{K}(\mathcal{D}) \mid d \sqsubseteq x\}.$$

In this Chapter $a, b, c, d \ldots$ always denote compact elements in lattices. Generic elements are denoted by $x, y, z \ldots$. **Comment:** I do not agree to cancel this, it helps the reader!

17.1.4. Definition. Let $\mathcal{D}, \mathcal{E}$ be complete lattices and $f : \mathcal{D} \to \mathcal{E}$.
(i) f is called *(Scott) continuous* iff for all directed $X \subseteq \mathcal{D}$ one has

$$f(\bigsqcup X) = \bigsqcup f(X) \; (= \bigsqcup\{f(x) \mid x \in X\}).$$

(ii) $[\mathcal{D} \to \mathcal{E}] = \{f : \mathcal{D} \to \mathcal{E} \mid f \text{ is Scott continuous functions}\}$.
(iii) f is called *strict* iff $f(\bot) = \bot$.
(iv) Write $[\mathcal{D} \to_s \mathcal{E}]$ for the collection of continuous strict maps.

17.1.5. Proposition. *Let $\mathcal{D}, \mathcal{E}$ be algebraic lattices.*

(i) *Let* $f \in [\mathcal{D} \to \mathcal{E}]$. *Then for* $x \in \mathcal{D}$

$$f(x) = \bigsqcup\{f(e) \mid e \sqsubseteq x \ \&\ e \in \mathcal{K}(\mathcal{D})\}.$$

(ii) *Let* $f, g \in [\mathcal{D} \to \mathcal{E}]$. *Suppose* $f \restriction \mathcal{K}(\mathcal{D}) = g \restriction \mathcal{K}(\mathcal{D})$. *Then* $f = g$.

PROOF. (i) Use that $x = \bigsqcup\{e \mid e \sqsubseteq x\}$ is a directed sup and that f is continuous.

(ii) By (i). ∎

17.1.6. DEFINITION. The category **ALG** has as objects the ω-algebraic complete lattices and as morphisms the continuous maps.

Comment: *Rightarrow* is used for implication, so we cannot use it for step function (macro step), the lemma is unreadable!

17.1.7. DEFINITION. (i) $[\mathcal{D} \to \mathcal{D}']$ is partially ordered pointwise as follows.

$$f \sqsubseteq g \iff \forall x \in \mathcal{D}. f(x) \sqsubseteq g(x).$$

(ii) If $a \in \mathcal{D}$, $a' \in \mathcal{D}'$, then $a \mapsto a'$ is the *step function* defined by

$$\begin{array}{rcll} (a \mapsto a')(d) & = & a', & \text{if } a \sqsubseteq d; \\ & = & \bot_{\mathcal{D}'}, & \text{else.} \end{array}$$

17.1.8. LEMMA. $[\mathcal{D} \to \mathcal{D}']$ *is a complete lattice with*

$$(\bigsqcup_{f \in X} f)(d) = \bigsqcup_{f \in X} f(d).$$

17.1.9. LEMMA. *For* $a, b \in \mathcal{D}$, $a', b' \in \mathcal{D}'$ *and* $f \in [\mathcal{D} \to \mathcal{D}']$ *one has*

(i) a *compact* $\Rightarrow$ $a \mapsto a'$ *is continuous.*
(ii) $a \mapsto a'$ *is continuous and* $a' \neq \bot$ $\Rightarrow$ a *is compact.*
(iii) a' *compact* $\Leftrightarrow$ $a \mapsto a'$ *compact.*
(iv) $a' \sqsubseteq f(a)$ $\Leftrightarrow$ $(a \mapsto a') \sqsubseteq f$.
(v) $b \sqsubseteq a \ \&\ a' \sqsubseteq b'$ $\Rightarrow$ $(a \mapsto a') \sqsubseteq (b \mapsto b')$.
(vi) $(a \mapsto a') \sqcup (b \mapsto b') \sqsubseteq (a \sqcap b) \mapsto (a' \sqcup b')$.

PROOF. Easy. ∎

17.1.10. LEMMA. *For all* $b, a_1, \dots, a_n \in \mathcal{D}$, $b', a'_1, \dots, a'_n \in \mathcal{D}'$, *and* $f \in [\mathcal{D} \to \mathcal{D}']$

$$\begin{array}{l} (b \mapsto b') \sqsubseteq (a_1 \mapsto a'_1) \sqcup \dots \sqcup (a_n \mapsto a'_n) \iff \\ \qquad\qquad \iff \exists I \subseteq \{1, \dots, n\} \, [\sqcup_{i \in I} a_i \sqsubseteq b \ \&\ b' \sqsubseteq \sqcup_{i \in I} a'_i]. \end{array}$$

Clearly in $(\Rightarrow)$ *we have* $I \neq \emptyset$ *if* $d \neq \bot_{\mathcal{D}}$.

PROOF. Easy. ∎

17.1.11. PROPOSITION. *Let* $\mathcal{D}, \mathcal{D}' \in$ **ALG**.

(i) *For* $f \in [\mathcal{D} \to \mathcal{D}']$ *one has* $f = \bigsqcup\{a \Rightarrow a' \mid a' \sqsubseteq f(a),\ a \in \mathcal{K}(\mathcal{D}), a' \in \mathcal{K}(\mathcal{D}')\}$.

(ii) *Let* $\mathcal{D}\in\mathbf{ALG}$ *and let* $f:[\mathcal{D}\to\mathcal{D}']$ *be compact. Then*

$$f=(a_1\mapsto a_1')\sqcup\ldots\sqcup(a_n\mapsto a_n'),$$

for some $a_1,\ldots,a_n\in\mathcal{K}(\mathcal{D}),a_1',\ldots,a_n'\in\mathcal{K}(\mathcal{D}')$.

(iii) $[\mathcal{D}\to\mathcal{D}']\in\mathbf{ALG}$.

Proof. (i) It suffices to show that RHS and LHS are equal when applied to an arbitrary element $d\in\mathcal{D}$.

$$\begin{array}{rcl} f(d) &=& f(\bigsqcup\{a\mid a\sqsubseteq d\ \&\ a\in\mathcal{K}(\mathcal{D})\})\\ &=& \bigsqcup\{f(a)\mid a\sqsubseteq d\ \&\ a\in\mathcal{K}(\mathcal{D})\}\\ &=& \bigsqcup\{a'\mid a'\sqsubseteq f(a)\ \&\ a\sqsubseteq d\ \&\ a\in\mathcal{K}(\mathcal{D}),a'\in\mathcal{K}(\mathcal{D}')\}\\ &=& \bigsqcup\{(a\mapsto a')(d)\mid a'\sqsubseteq f(a)\ \&\ a\sqsubseteq d\ \&\ a\in\mathcal{K}(\mathcal{D}),a'\in\mathcal{K}(\mathcal{D}')\}\\ &=& \bigsqcup\{(a\mapsto a')(d)\mid a'\sqsubseteq f(a)\ \&\ a\in\mathcal{K}(\mathcal{D}),a'\in\mathcal{K}(\mathcal{D}')\}\\ &=& (\bigsqcup\{(a\mapsto a')\mid a'\sqsubseteq f(a)\ \&\ a\in\mathcal{K}(\mathcal{D}),a'\in\mathcal{K}(\mathcal{D}')(d)\}. \end{array}$$

(ii) For f compact one has $f=\bigsqcup\{a\mapsto a'\mid a'\sqsubseteq f(a)\ \&\ a\in\mathcal{K}(\mathcal{D}),a'\in\mathcal{K}(\mathcal{D}')\}$, by (i). Hence by Lemma 17.1.2 for some $a_1,\ldots,a_n\in\mathcal{K}(\mathcal{D}),a_1',\ldots,a_n'\in\mathcal{K}(\mathcal{D}')$

$$f=(a_1\mapsto a_1')\sqcup\ldots\sqcup(a_n\mapsto a_n'). \tag{17.1}$$

(iii) It remains to show that there are only countably many compact elements in $[\mathcal{D}\to\mathcal{D}]$. Since $\mathcal{K}(\mathcal{D})$ is countable, there are only countably many expressions in the RHS of (17.1). (The cardinality is $\le\Sigma_n n.\aleph_0^2=\aleph_0$.) Therefore there are countable many compact $f\in[\mathcal{D}\to\mathcal{D}]$. (There may be more expressions on the RHS for one f, but this results in less compact elements.) ∎

17.1.12. Definition. (i) The category $\mathbf{ALG}_a$ has the same objects as $\mathbf{ALG}$ and as morphisms $\mathbf{ALG}_a(\mathcal{D},\mathcal{D}')$ maps $f:\mathcal{D}\to\mathcal{D}'$ that satisfy the properties 'compactness preserving' and 'additive':

$$\begin{array}{ll} \text{(cmp-pres)} & \forall a\in\mathcal{K}(\mathcal{D}).f(a)\in\mathcal{K}(\mathcal{D}');\\ \text{(add)} & \forall X\subseteq\mathcal{D}.f(\bigsqcup X)=\bigsqcup f(X). \end{array}$$

(ii) The category $\mathbf{ALG}_a^s$ has the same objects as $\mathbf{ALG}_a$ and as morphisms $\mathbf{ALG}_a^s(\mathcal{D},\mathcal{D}')$ maps $f:\mathcal{D}\to\mathcal{D}'$ satisfying (cmp-pres), (add) and

$$\text{(s)}\quad \forall d\in\mathcal{D}.[f(d)=\bot_{\mathcal{D}'}\ \Rightarrow\ d=\bot_{\mathcal{D}}].$$

17.1.13. Remark. (i) Note that the requirement (add) implies that a morphism f is continuous (only required to preserve sups for directed subsets X) and strict, i.e. $f(\bot_{\mathcal{D}})=\bot_{\mathcal{D}'}$.

(ii) Remember that $\mathcal{K}^s(\mathcal{D})=\mathcal{K}(\mathcal{D})-\{\bot_{\mathcal{D}}\}$. Note that $\mathbf{ALG}_a^s(\mathcal{D},\mathcal{D}')$ consists of maps satisfying

$$\begin{array}{ll} (\text{cmp-pres}^s) & \forall a\in\mathcal{K}^s(\mathcal{D}).f(a)\in\mathcal{K}^s(\mathcal{D}');\\ \text{(add)} & \forall X\subseteq\mathcal{D}.f(\bigsqcup X)=\bigsqcup f(X). \end{array}$$

17.1.14. Remark. By contrast to Proposition 17.1.11(iii) $\mathbf{ALG}_a(\mathcal{D},\mathcal{D}')\notin\mathbf{ALG}_a$, because from (i) of that Proposition it follows that the set of compactness preserving functions is not closed under taking the supremum.

Zip structures

The idea of a *zip structure* is that we have an element of **ALG** such that the information of two compact elements is 'zipped' together, not necessarily in such a way that the constituents can be found back.

17.1.15. Definition. (i) A *zip structure* is a pair $\mathcal{D}_Z = \langle \mathcal{D}, Z \rangle$ with $\mathcal{D} \in \mathbf{ALG}$ and $Z : \mathcal{K}(\mathcal{D}) \times \mathcal{K}(\mathcal{D}) \to \mathcal{K}(\mathcal{D})$ an arbitrary map.

(ii) The category **ZS** has zip structures as objects and maps $f : \mathcal{D} \to \mathcal{D}'$ such that ($a, b, c, \ldots$ ranging over $\mathcal{K}(\mathcal{D})$)

$$\begin{array}{ll} \text{(cmp-pres)} & \forall a. f(a) \in \mathcal{K}(\mathcal{D}'); \\ \text{(add)} & \forall X \subseteq D. f(\bigsqcup X) = \bigsqcup f(X); \\ (Z\text{-comm}) & \forall a, b. f(Z(a, b)) = Z'(f(a), f(b)) \end{array}$$

as morphisms $\mathbf{ZS}(\langle \mathcal{D}, Z \rangle, \langle \mathcal{D}', Z' \rangle)$. The second requirement implies that a morphism f is continuous (only required to preserve sups for directed sets X) and strict, i.e. $f(\bot_{\mathcal{D}}) = \bot_{\mathcal{D}'}$.

17.1.16. Lemma. *Let $f : \mathcal{D} \to \mathcal{D}'$ be a continuous function with $\mathcal{D}, \mathcal{D}' \in \mathbf{ALG}$. Then, for any $X \subseteq \mathcal{D}$, $b' \in \mathcal{K}(\mathcal{D}')$,*

$$b' \sqsubseteq f(\bigsqcup X) \ \Leftrightarrow\ \exists Z \subseteq_{fin} X \cap \mathcal{K}(\mathcal{D}). b' \sqsubseteq f(\bigsqcup Z).$$

Proof. Note that $\Xi = \{\bigsqcup Z \mid Z \subseteq_{\text{fin}} X \cap \mathcal{K}(\mathcal{D})\}$ is a directed set and $\bigsqcup \Xi = \bigsqcup X$. Moreover, by monotonicity of f, also the set $\{f(\bigsqcup Z) \mid Z \subseteq_{\text{fin}} X \cap \mathcal{K}(\mathcal{D})\}$ is directed. Therefore

$$\begin{array}{rcll} b' \sqsubseteq f(\bigsqcup X) & \Leftrightarrow & b' \sqsubseteq f(\bigsqcup \Xi) & \\ & \Leftrightarrow & b' \sqsubseteq \bigsqcup f(\Xi), & \text{since } f \text{ is continuous,} \\ & \Leftrightarrow & b' \sqsubseteq \bigsqcup \{f(\bigsqcup Z) \mid Z \subseteq_{\text{fin}} X \cap \mathcal{K}(\mathcal{D})\}, & \text{by definition of } \Xi, \\ & \Leftrightarrow & \exists Z \subseteq_{\text{fin}} X \cap \mathcal{K}(\mathcal{D}). b' \sqsubseteq f(\bigsqcup Z), & \text{since } b' \text{ is compact.} \ \blacksquare \end{array}$$

17.1.17. Corollary. *A map $f : \mathcal{D} \to \mathcal{D}'$ satisfies* (add) *iff f is Scott continuous and*

$$\forall X \subseteq_{fin} \mathcal{K}(\mathcal{D}). f(\bigsqcup X) = \bigsqcup f(X). \tag{1}$$

Proof. The non-trivial direction is to show, assuming f is Scott continuous and satisfies (1), that f is additive. By monotonicity of f we only need to show for all $X \subseteq \mathcal{D}$

$$f(\bigsqcup X) \sqsubseteq \bigsqcup f(X).$$

As $\mathcal{D}'$ is algebraic, it suffices to assume $b' \sqsubseteq f(\bigsqcup X)$ and conclude $b' \sqsubseteq \bigsqcup f(X)$. By the Lemma $\exists Z \subseteq_{\text{fin}} X \cap \mathcal{K}(\mathcal{D}). b' \sqsubseteq f(\bigsqcup Z) = \bigsqcup f(Z)$, so $b' \sqsubseteq \bigsqcup f(X)$. $\blacksquare$

We now specialize this general framework to special zip structures.

17.1.18. Definition. Let $\mathcal{D}_Z = (\mathcal{D}, Z)$ be a zip structure.

(i) Then $\mathcal{D}_Z$ is a *lazy zip structure* if the following holds.

(1) (Z-contr) $a \sqsubseteq a' \ \&\ b' \sqsubseteq b \ \Rightarrow\ Z(a',b') \sqsubseteq Z(a,b)$;

(2) (Z-add) $Z(a, b_1 \sqcup b_2) = Z(a,b_1) \sqcup Z(a,b_2)$;

(3) (Z-lazy) $Z(\bot_{\mathcal{D}}, \bot_{\mathcal{D}}) \sqsubseteq Z(a,b)$.

(ii) **LZS** is the full subcategory of **ZS** consisting of lazy zip structures.

17.1.19. Definition. Let $\mathcal{D}_Z = (\mathcal{D}, Z) \in \mathbf{ZS}$.

(i) Then $\mathcal{D}_Z$ is a *natural zip structure* if $\mathcal{D}_Z \in \mathbf{LZS}$ and moreover

$$(Z\text{-bot}) \quad Z(\bot_{\mathcal{D}}, \bot_{\mathcal{D}}) = \bot_{\mathcal{D}}.$$

(ii) **NZS** is the full subcategory of **LZS** consisting of natural zip structures.

17.1.20. Remark. Since condition (Z-bot) is stronger than (Z-lazy), D is natural if it satisfies (Z-contr), (Z-add) and (Z-bot). In fact, (Z-bot) corresponds to ($\top\!\rightarrow$), and (Z-lazy) to the weaker ($\top_{\rm lazy}$).

Strict zip structures

17.1.21. Definition. Let $\mathcal{D}$ be an ω-algebraic lattice.

(i) Let $Z : (\mathcal{K}^s(\mathcal{D}) \times \mathcal{K}^s(\mathcal{D})) \to \mathcal{K}^s(\mathcal{D})$. Then $\mathcal{D}_Z = \langle \mathcal{D}, Z\rangle$ is called a *strict zip structure*.

(ii) If we write $Z(a,b)$, then it is always understood that $(a,b) \in \mathrm{dom}(Z)$.

(iii) The category $\mathbf{ZS}^s$ consists of strict zip structures as objects and as morphisms maps f satisfying

$$\begin{array}{ll} (\text{cmp-pres}^{\rm s}) & \forall a \in \mathcal{K}^{\rm s}(\mathcal{D}).f(a) \in \mathcal{K}^{\rm s}(\mathcal{D}'); \\ (\text{add}) & \forall X \subseteq D.f(\bigsqcup X) = \bigsqcup f(X); \\ (Z\text{-comm}) & \forall a,b.f(Z(a,b)) = Z'(f(a), f(b)). \end{array}$$

17.1.22. Definition. (i) Let $\mathcal{D}_Z = \langle \mathcal{D}, Z\rangle \in \mathbf{ZS}^s$. Then $\mathcal{D}_Z$ is called a *proper strict zip structure*, if it satisfies the following conditions.

$$\begin{array}{ll} (Z\text{-contr}) & a \sqsubseteq a' \ \&\ b' \sqsubseteq b \ \Rightarrow\ Z(a',b') \sqsubseteq Z(a,b); \\ (Z\text{-add}) & Z(a, b_1 \sqcup b_2) = Z(a,b_1) \sqcup Z(a,b_2). \end{array}$$

(ii) $\mathbf{PLS}^s$ is the full subcategory of $\mathbf{ZS}^s$ consisting of proper strict zip structures.

17.1.23. Remark. In Section 15.3 we introduced the names $\mathrm{MSL}^\top$, MSL, $\mathrm{TS}^\top$, TS for collections of meet semi-lattices and type structures. The superscript $\top$ in these names denotes that the structures in the relevant collections have a top element.

In the present Section we introduced the names $\mathbf{ALG}_a, \mathbf{ALG}_a^s, \mathbf{ZS}, \mathbf{ZS}^s$. The superscript s, to be read as 'strict', concerns the top element of $\mathcal{K}(\mathcal{D})_{\leq}$ (see Definition 17.2.5) and suggests that we are not interested in it.

In the next Section we will establish equivalences of categories like

$$\begin{array}{rcl} \mathbf{MSL}^\top & \cong & \mathbf{ALG}_a; \\ \mathbf{MSL} & \cong & \mathbf{ALG}_a^s; \\ \mathbf{TS}^\top & \cong & \mathbf{ZS}; \\ \mathbf{TS} & \cong & \mathbf{ZS}^s. \end{array}$$

Note that under this correspondence there is a sort of adjunction in the superscripts of the names due to the fact that in the lefthand side of this table the presence of a top is given explicitly, whereas in the right hand side the name indicates when we are *not* interested in the top (of $\mathcal{K}(\mathcal{D})_{\leq}$). In particular, note that **TS** does not correspond with **ZS**.

17.2. Meet-semi lattices and algebraic lattices

Comment:

- X, Y are filters and subsets of $\mathcal{D}$.
- $\mathcal{X}$ is a set of filters.
- $\mathcal{F}^{\mathcal{S}}$ is the set of all filters over $\mathcal{S}$
- $\tilde{\mathcal{F}}^{\mathcal{S}} = \langle \mathcal{F}^{\mathcal{S}}, \subseteq \rangle$ (the macro is FFL) I use coercion.

The main results of this section are concerned with equivalences between categories of type structures and lambda structures. The proto-type result is Corollary 17.2.15, stating the equivalence between the categories $\mathbf{MSL}^{\top}$ and $\mathbf{ALG}_a$. We start introducing some categories.

Categories of meet-semi lattices and type structures

For use in Chapter 17 we will introduce some categories related to given classes of type structures.

17.2.1. Definition. (i) The category **MSL** has as objects at most countable meet semi-lattices and as morphisms maps $f : \mathcal{M} \to \mathcal{M}'$, preserving $\leq, \cap$:

$$\begin{array}{rcl} A \leq B & \Rightarrow & f(A) \leq' f(B); \\ f(A \cap B) & = & f(A) \cap' f(B). \end{array}$$

(ii) The category $\mathbf{MSL}^{\top}$ is as **MSL**, but based on top meet semi-lattices. So now also $f(\top) = \top'$ for morphisms.

The difference between **MSL** and $\mathbf{MSL}^{\top}$ is that, in the **MSL** case, the top element is either missing or not relevant (not preserved by morphisms).

17.2.2. Definition. (i) The category **TS** has as objects the at most countable type structures and as morphisms maps $f : \mathcal{S} \to \mathcal{S}'$, preserving $\leq, \cap, \to$:

$$\begin{array}{rcl} A \leq B & \Rightarrow & f(A) \leq' f(B); \\ f(A \cap B) & = & f(A) \cap' f(B); \\ f(A \to B) & = & f(A) \to' f(B). \end{array}$$

(ii) The category $\mathbf{TS}^{\top}$ is as **TS**, but based on top type structures. Now also

$$f(\top) = \top'$$

for morphisms.

Remember that in Definition 15.3.10 we defined four full subcategories of **TS** by specifying in each case the objects.

(i) $\mathbf{GTS}^{\top}$ with as objects the graph top type structures.
(ii) $\mathbf{LTS}^{\top}$ with as objects the lazy top type structures.
(iii) $\mathbf{NTS}^{\top}$ with as objects the natural top type structures.
(iv) **PTS** with as objects the proper type structures.

We define now the functors establishing the equivalences between categories of type structures and lambda structures.

17.2.3. Remark. If $\mathcal{T}$ is countable, then $\mathcal{F}^{\mathcal{T}} \in \mathbf{ALG}_a$.

17.2.4. Definition. We define a map $\mathsf{Flt} : \mathbf{MSL}^{\top} \to \mathbf{ALG}_a$, that will turn out to be a functor, as follows.

(i) On objects $\mathcal{S} \in \mathbf{MSL}$ one defines

$$\mathsf{Flt}(\mathcal{S}) = \langle \mathcal{F}^{\mathcal{S}}, \subseteq \rangle.$$

(ii) On morphisms $f : \mathcal{S} \to \mathcal{S}'$ one defines $\mathsf{Flt}(f) : \mathcal{F}^{\mathcal{S}} \to \mathcal{F}^{\mathcal{S}'}$ by

$$\mathsf{Flt}(f)(X) = \{s' \mid \exists s \in X. f(s) \leq s'\}.$$

17.2.5. Definition. Let $\mathcal{D} \in \mathbf{ALG}_a$. On $\mathcal{K}(\mathcal{D})$ define

$$d \leq e \;\Leftrightarrow\; e \sqsubseteq d.$$

17.2.6. Proposition. $(\mathcal{K}(\mathcal{D}), \leq) \in \mathbf{MSL}^{\top}$.

Proof. Immediate noticing that

$$\begin{array}{rcl} d \sqcap_{\mathcal{K}(\mathcal{D})} e & = & d \sqcup_{\mathcal{D}} e; \\ \perp_{\mathcal{K}(\mathcal{D})} & = & \top_{\mathcal{D}}; \\ \top_{\mathcal{K}(\mathcal{D})} & = & \perp_{\mathcal{D}}. \blacksquare \end{array}$$

Instead of $\sqcap_{\leq}$ we often write $\cap_{\leq}$ or simply $\cap$.

17.2.7. Definition. We define a map $\mathsf{Cmp} : \mathbf{ALG}_a \to \mathbf{MSL}^{\top}$, that will turn out to be a functor, as follows.

(i) On objects $\mathcal{D} \in \mathbf{ALG}_a$ one defines Cmp by

$$\mathsf{Cmp}(\mathcal{D}) = (\mathcal{K}(\mathcal{D}), \leq).$$

(ii) On morphisms $f \in \mathbf{ALG}_a(\mathcal{D}, \mathcal{D}')$ one defines $\mathsf{Cmp}(f)$ by

$$\mathsf{Cmp}(f)(d) = f(d).$$

17.2.8. Lemma. *The map* Cmp *is a functor.*

PROOF. Let $f \in \mathbf{ALG}_a(\mathcal{D}, \mathcal{D}')$. Note that $\mathsf{Cmp}(f) = f \restriction \mathcal{K}(\mathcal{D}) : \mathcal{K}(\mathcal{D}) \rightarrow \mathcal{K}(\mathcal{D}')$, by (cmp-pres). By the fact that f is additive one has $f(\perp_{\mathcal{D}}) = \perp_{\mathcal{D}'}$, which is $f(\top_{\mathcal{K}(\mathcal{D})}) = \top_{\mathcal{K}(\mathcal{D}')}$, and

$$f(a \cap_{\mathcal{K}(\mathcal{D})} b) = f(a \sqcup_{\mathcal{D}} b) = f(a) \sqcup_{\mathcal{D}'} f(b) = f(a) \cap_{\mathcal{K}(\mathcal{D})} f(b).$$

Also f is monotonic as it is continuous. ∎

17.2.9. LEMMA. *Let $\mathcal{S} \in \mathbf{MSL}^{\top}$ be a meet semi-lattice, $I \neq \emptyset$ and $s, t, s_i \in \mathcal{S}$. Then*

(i) *In $\mathcal{F}^{\mathcal{S}}$ we have*

$$\bigsqcup_{i \in I} \uparrow s_i = \uparrow \bigcap_{i \in I} s_i.$$

(ii) *In $\mathcal{K}(\mathcal{F}^{\mathcal{S}})$ we have*

$$\bigcap_{i \in I} \uparrow s_i = \uparrow \bigcap_{i \in I} s_i,$$

where the bigcap denote the infima in respectively $\mathcal{K}(\mathcal{S})$ and $\mathcal{S}$.

(iii) $\uparrow s \leq_{\mathcal{K}(\mathcal{F}^{\mathcal{S}})} \uparrow t \;\Leftrightarrow\; s \leq_{\mathcal{S}} t.$

PROOF. From previous definitions. ∎

In the remainder of the present section we write $\leq$ instead of $\leq_{\mathcal{K}(\mathcal{F}^{\mathcal{S}})}$.

17.2.10. LEMMA. *Let $f \in \mathbf{MSL}^{\top}(\mathcal{S}, \mathcal{S}')$, $s \in \mathcal{S}$. Then $\mathsf{Flt}(f)(\uparrow s) = \uparrow f(s)$.*

PROOF.

$$\begin{array}{rcll} \mathsf{Flt}(f)(\uparrow s) & = & \{s' \mid \exists t \in \uparrow s. f(t) \leq s'\}, & \\ & = & \{s' \mid \exists t \geq s. f(t) \leq s'\}, & \\ & = & \{s' \mid f(s) \leq s'\}, & \text{since } f \text{ is monotone,} \\ & = & \uparrow f(s). & \end{array}$$

17.2.11. PROPOSITION. *Flt is a functor from $\mathbf{MSL}^{\top}$ to $\mathbf{ALG}_a$.*

PROOF. We have to prove that Flt transforms a morphism in $\mathbf{MSL}^{\top}$ into a morphism in $\mathbf{ALG}_a$. Let $f \in \mathbf{MSL}^{\top}(\mathcal{S}, \mathcal{S}')$, $\uparrow s \in K(\mathcal{F}^{\mathcal{S}})$. By Lemma 17.2.10 $\mathsf{Flt}(f)(\uparrow s) = \uparrow f(s)$, which is compact in $\mathcal{F}^{\mathcal{S}'}$, hence $\mathsf{Flt}(f)$ satisfies (cmp-pres).

$\mathsf{Flt}(f)$ satisfies (add). Indeed, by Corollary 17.1.17 and the fact that $\mathsf{Flt}(f)$ is trivially Scott continuous, it is enough to prove that it commutes with finite joins of compact elements. Let I be non-empty. We have

$$\begin{array}{rcll} \mathsf{Flt}(f)(\bigsqcup_{i \in I} \uparrow s_i) & = & \mathsf{Flt}(f)(\uparrow \bigcap_{i \in I} s_i), & \text{by Lemma 17.2.9(ii),} \\ & = & \uparrow f(\bigcap_{i \in I} s_i), & \text{by Lemma 17.2.10,} \\ & = & \uparrow \bigcap_{i \in I} f(s_i), & \text{since } f \text{ commutes with } \cap, \\ & = & \bigsqcup_{i \in I} \uparrow f(s_i), & \text{by Lemma 17.2.9(ii),} \\ & = & \bigsqcup_{i \in I} \mathsf{Flt}(f)(\uparrow s_i) & \text{by Lemma 17.2.10.} \end{array}$$

If I is empty, and $\top, \top'$ are respectively the bottoms of $\mathcal{S}, \mathcal{S}'$, then

$$\begin{array}{rcll}
\mathsf{Flt}(f)(\bigsqcup_\emptyset \uparrow s_i) & = & \mathsf{Flt}(f)(\uparrow \top), & \\
& = & \uparrow f(\top), & \text{by Lemma 17.2.10,} \\
& = & \uparrow \top', & \text{since } f \text{ preserves tops,} \\
& = & \bigsqcup_\emptyset \mathsf{Flt}(f)(\uparrow s_i). &
\end{array}$$

So $\mathsf{Flt}(f)$ satisfies (add). ■

It is possible to leave out conditions (cmp-pres) and (add), obtaining the category **ALG**. Then one needs to consider *approximable maps* as morphisms in the category $\mathbf{MSL}^\top$. See Exercise ?? the Exercise is to be written by Fabio? Or did we have it?

Now we will show that the functors Flt and Cmp establish an equivalence between the categories $\mathbf{MSL}^\top$ and $\mathbf{ALG}_a$. First we need another Lemma.

17.2.12. Lemma. (i) *Let $\mathcal{X} \subseteq \mathcal{F}^{\mathcal{K}(\mathcal{D})}$. Then taking sups in $\mathcal{D}$ one has*

$$\bigsqcup(\bigcup \mathcal{X}) = \bigsqcup_{X \in \mathcal{X}} (\bigsqcup X).$$

(ii) *Let $\theta \subseteq \mathcal{K}(\mathcal{D})$ be non-empty. Then taking the sups in $\mathcal{D}$ one has*

$$\bigsqcup(\uparrow\theta) = \bigsqcup \theta.$$

Proof. (i) Realising that a sup (in $\mathcal{D}$) of a union of $\{Y_i\}_{i \in I} \subseteq \mathcal{K}(\mathcal{D})$ is the sup of the sups $\bigsqcup Y_i$, one has

$$\bigsqcup(\bigcup_{i \in I} Y_i) = \bigsqcup_{i \in I} (\bigsqcup Y_i).$$

The result follows by making an α-conversion $[i := X]$ and taking $I = \mathcal{X}$ and $Y_X = X$.

(ii) $\uparrow\theta$ is obtained from θ by taking extensions and intersections in $\mathcal{K}(\mathcal{D})$. Now the order in this $\mathbf{MSL}^\top$is the reverse of the one induced by $\mathcal{D}$, therefore $\uparrow\theta$ is obtained by taking smaller elements and unions (in $\mathcal{D}$). But then taking the big union the result is the same. ■

Comment: the maps where ξ, τ, σ (ξ is a type environment in chapter 14), now I used the macros maps, mapt and mapx

17.2.13. Proposition. (i) *Let $\mathcal{S} \in \mathbf{MSL}^\top$. Then $\circledast = \circledast_\mathcal{S} : \mathcal{S} \to \mathcal{K}(\mathcal{F}^\mathcal{S})$ defined by $\circledast_\mathcal{S}(s) = \uparrow s$ is an $\mathbf{MSL}^\top$ isomorphism.*

$$\begin{tikzcd}
\mathcal{S} \arrow[rr, dotted, "\mathsf{Flt}"] \arrow[dr, "\circledast=\uparrow"'] & & \mathcal{F}^\mathcal{S} \arrow[dl, dotted, "\mathsf{Cmp}"] \\
& \mathcal{K}(\mathcal{F}^\mathcal{S}) &
\end{tikzcd}$$

(ii) *Let* $\mathcal{D}\in\mathbf{ALG}_a$. *Then* $\ominus=\ominus_{\mathcal{D}}:\mathcal{F}^{\mathcal{K}(\mathcal{D})}\to\mathcal{D}$ *defined by* $\ominus(X)=\bigsqcup X$, *is an* $\mathbf{ALG}_a$ *isomorphism with inverse* $\odot:\mathcal{D}\to\mathcal{F}^{\mathcal{K}(\mathcal{D})}$ *defined by* $\odot(x)=\{c\in\mathcal{K}(\mathcal{D})\mid c\sqsubseteq x\}$.

$$\begin{tikzcd}
\mathcal{K}(\mathcal{D}) \arrow[dr, dotted, "\mathsf{Flt}"'] & & \mathcal{D} \arrow[ll, dotted, "\mathsf{Cmp}"'] \\
& \mathcal{F}^{\mathcal{K}(\mathcal{D})} \arrow[ur, "\ominus=\bigsqcup"'] &
\end{tikzcd}$$

Proof. (i) By Proposition 15.4.4(v) $\circledast$ is a surjection. It is also 1-1, since $\uparrow s=\uparrow t\;\Rightarrow\; s\le t\le s\;\Rightarrow\; s=t$. Moreover, $\circledast$ preserves $\le$:

$$\begin{array}{rcll}
s\le t & \Leftrightarrow & \uparrow t\subseteq\uparrow s & \\
& \Leftrightarrow & \uparrow s\le\uparrow t, & \text{by definition of } \le \text{ on } \mathcal{K}(\mathcal{F}^{\mathcal{S}}).
\end{array}$$

Also $\circledast$ preserves $\cap$:

$$\begin{array}{rcll}
\circledast(s\cap t) & = & \uparrow(s\cap t) & \\
& = & \uparrow s\sqcup\uparrow t & \text{in } \mathcal{F}^{\mathcal{S}}, \text{ by 15.4.4(iii)},\\
& = & \uparrow s\cap\uparrow t & \text{in } \mathcal{K}(\mathcal{F}^{\mathcal{S}}), \text{ by definition of } \le \text{ on } \mathcal{K}(\mathcal{F}^{\mathcal{S}}),\\
& = & \circledast(s)\cap\circledast(t). &
\end{array}$$

Then $\circledast(\top_{\mathcal{S}})=\uparrow\top_{\mathcal{S}}=\top_{\mathcal{K}(\mathcal{F}^{\mathcal{S}})}$, since $\{\top_{\mathcal{S}}\}$ is the $\subseteq$-smallest filter; hence $\circledast$ preserves tops.

Finally $\circledast^{-1}$ preserves $\le,\sqcap_{\mathcal{K}(\mathcal{F}^{\mathcal{S}})}$ and $\top_{\mathcal{K}(\mathcal{F}^{\mathcal{S}})}$, as by Lemma 15.4.4(v) an element $c\in\mathcal{K}(\mathcal{F}^{\mathcal{S}})$ is of the form $c=\uparrow s$, with $s\in\mathcal{S}$ and $\circledast^{-1}(\uparrow s)=s$.

(ii) We have $\ominus\circ\odot=\mathbf{1}_{\mathcal{D}}$ and $\odot\circ\ominus=\mathbf{1}_{\mathcal{F}^{\mathcal{K}(\mathcal{D})}}$:

$$\begin{array}{rcll}
\ominus(\odot(x)) & = & \bigsqcup\{c\in\mathcal{K}(\mathcal{D})\mid c\sqsubseteq x\} & \\
& = & x, & \text{since } \mathcal{D}\in\mathbf{ALG}_a.\\
\odot(\ominus(X)) & = & \{c\mid c\sqsubseteq\bigsqcup X\} & \\
& = & \{c\mid c\in X\}, & \text{since one has}\\
c\sqsubseteq_{\mathcal{D}}\bigsqcup X & \Leftrightarrow & \exists x\in X\; c\sqsubseteq_{\mathcal{D}} x, & \text{as } c \text{ is compact and } X\subseteq\mathcal{K}(\mathcal{D})\subseteq\mathcal{D} \text{ is}\\
& & & \text{a filter w.r.t. } \le, \text{ so directed w.r.t. } \sqsubseteq\\
& \Leftrightarrow & \exists x\in X\; x\le_{\mathcal{K}(\mathcal{D})} c & \\
& \Leftrightarrow & c\in X, & \text{as } X \text{ is a filter on } \mathcal{K}(\mathcal{D}).
\end{array}$$

We still have to show that $\ominus$ and $\odot$ are morphisms. One easiliy sees that $\ominus$ satisfies (cmp-pres). The map $\ominus$ is also additive, i.e. $\bigsqcup\ominus(\mathcal{X})=\ominus(\bigsqcup\mathcal{X})$ for arbitrary $\mathcal{X}\subseteq\mathcal{F}^{\mathcal{K}(\mathcal{D})}$. Indeed,

$$\begin{array}{rcll}
\bigsqcup\ominus(\mathcal{X}) & = & \bigsqcup\limits_{X\in\mathcal{X}}(\bigsqcup X), & \text{by definition of } \ominus,\\
& = & \bigsqcup(\bigcup\mathcal{X}), & \text{by Proposition 17.2.12(i)},\\
& = & \bigsqcup(\uparrow(\bigcup\mathcal{X})), & \text{by Proposition 17.2.12(ii)},\\
& = & \ominus(\uparrow(\bigcup\mathcal{X})), & \text{by definition of } \ominus,\\
& = & \ominus(\bigsqcup\mathcal{X}), & \text{by Proposition 15.4.4(i)}.
\end{array}$$

Now we have to prove that $\circledcirc$ satisfies (cmp-pres) and (add). As to (cmp-pres), assume that $b \in \mathcal{K}(\mathcal{D})$ and $\circledcirc(b) \sqsubseteq \bigsqcup X$, with X directed. Then $b \sqsubseteq \bigsqcup \ominus(X)$, since $\ominus$ satisfies (add). Since b is compact, there exists $x \in X$ such that $b \sqsubseteq \ominus(x)$, hence $\circledcirc(b) \sqsubseteq x$ and we are done. As to (add), let $X \subseteq \mathcal{D}$. Then

$$\begin{array}{rcll}
\circledcirc(\bigsqcup X) & = & \circledcirc(\bigsqcup\{\ominus(\circledcirc(x)) \mid x \in X\}), & \\
& = & \circledcirc(\ominus(\bigsqcup\{\circledcirc(x) \mid x \in X\})), & \text{since } \ominus \text{ satisfies (add)}, \\
& = & \bigsqcup\{\circledcirc(x) \mid x \in X\}. \blacksquare &
\end{array}$$

17.2.14. Corollary. (i) *Let* $\mathcal{S} \in \mathbf{MSL}^\top$. *Then*

$$\mathcal{S} \cong \mathcal{K}(\mathcal{F}^{\mathcal{S}}).$$

(ii) *Let* $\mathcal{D} \in \mathbf{ALG}_a$. *Then*

$$\mathcal{D} \cong \mathcal{F}^{\mathcal{K}(\mathcal{D})}.$$

17.2.15. Corollary. *The categories* $\mathbf{MSL}^\top$ *and* $\mathbf{ALG}_a$ *are equivalent; in fact the isomorphisms in 17.2.13 form natural isomorphisms showing*

$$\mathsf{Cmp} \circ \mathsf{Flt} \cong \mathbf{1}_{\mathbf{MSL}^\top} \ \& \ \mathsf{Flt} \circ \mathsf{Cmp} \cong \mathbf{1}_{\mathbf{ALG}_a}.$$

Proof. First one has to show that in $\mathbf{MSL}^\top$ the following diagram commutes.

$$\begin{tikzcd}
\mathcal{S} \arrow[r, "\uparrow"] \arrow[d, "f"'] & \mathcal{K}(\mathcal{F}^{\mathcal{S}}) \arrow[d, "\mathsf{Cmp}(\mathsf{Flt}(\mu))"] \\
\mathcal{S}' \arrow[r, "\uparrow"] & \mathcal{K}(\mathcal{F}^{\mathcal{S}'})
\end{tikzcd}$$

One must show $\mathsf{Cmp}(\mathsf{Flt}(f))(\uparrow s) = \uparrow(f(s))$. This follows from Lemma 17.2.10 and the definition of Cmp.

Secondly one has to show that in $\mathbf{ALG}_a$ the following diagram commutes.

$$\begin{tikzcd}
\mathcal{F}^{\mathcal{K}(\mathcal{D})} \arrow[r, "\bigsqcup"] \arrow[d, "\mathsf{Flt}(\mathsf{Cmp}(f))"'] & \mathcal{D} \arrow[d, "f"] \\
\mathcal{F}^{\mathcal{K}(\mathcal{D}')} \arrow[r, "\bigsqcup"] & \mathcal{D}'
\end{tikzcd}$$

Now for $X \in \mathcal{F}^{\mathcal{K}(\mathcal{D})}$ one has

$$\begin{array}{rcl}
\mathsf{Flt}(\mathsf{Cmp}(f))(X) & = & \{d' \in \mathcal{K}(\mathcal{D}') \mid \exists d \in X.\, f(d) \leq d'\} \\
& = & \{d' \in \mathcal{K}(\mathcal{D}') \mid \exists d \in X.\, d' \sqsubseteq f(d)\}.
\end{array}$$

Hence, using also the continuity of f,

$$\bigsqcup(\mathsf{Flt}(\mathsf{Cmp}(f))(X)) = \bigsqcup{}_{d \in X} f(d) = f(\bigsqcup X),$$

and the diagram commutes. As before we have $\mathsf{Flt} \circ \mathsf{Cmp} \cong \mathbf{1}_{\mathbf{ALG}_a}$. $\blacksquare$

This result is a special case of Stone duality, cf. Johnstone [1986] (II, 3.3).

Equivalence between MSL and $\mathbf{ALG}^s_a$

17.2.16. Definition. Let $\mathcal{S} \in \mathbf{MSL}$.

(i) The set of filters over $\mathcal{S}$, notation $\mathcal{F}^{\mathcal{S}}$, is defined as in Definition 15.4.1.

(ii) $\mathcal{F}^{\mathcal{S}}_s = \mathcal{F}^{\mathcal{S}} \cup \{\emptyset\}$ is the extension of $\mathcal{F}^{\mathcal{S}}$ with the empty subset of $\mathcal{S}$.

17.2.17. Proposition. *Let $S \in \mathbf{MSL}$.*

(i) *$\mathcal{F}^{\mathcal{S}}_s = \langle \mathcal{F}^{\mathcal{S}}_s, \subseteq \rangle$ is a complete lattice, with for $\mathcal{X} \subseteq \mathcal{F}^{\mathcal{S}}_s$ the sup is*

$$\bigsqcup \mathcal{X} = \begin{cases} \emptyset, & \text{if } \mathcal{X} = \emptyset \text{ or } \mathcal{X} = \{\emptyset\}, \\ \uparrow(\cup\mathcal{X}), & \text{else .} \end{cases}$$

(ii) *For $t \in S$ one has $\uparrow t = \uparrow\{t\}$ and $\uparrow t \in \mathcal{F}^{\mathcal{S}}_s$.*

(iii) *For $t, t' \in S$ one has $\uparrow t \sqcup \uparrow t' = \uparrow(t \cap t')$.*

(iv) *For $X \in \mathcal{F}^{\mathcal{S}}_s$ one has*

$$\begin{aligned} X &= \bigsqcup\{\uparrow t \mid t \in X\} &&= \bigsqcup\{\uparrow t \mid \uparrow t \subseteq X\} \\ &= \bigcup\{\uparrow t \mid t \in X\} &&= \bigcup\{\uparrow t \mid \uparrow t \subseteq X\}. \end{aligned}$$

(v) *$\{\uparrow t \mid t \in S\} \cup \{\emptyset\}$ is the set of finite elements of $\mathcal{F}^{\mathcal{S}}_s$.*

(vi) *$\mathcal{F}^{\mathcal{S}}_s \in \mathbf{ALG}^s_a$.*

Proof. Immediate. ■

17.2.18. Remark. If $\mathcal{T}$ is countable, then $\mathcal{F}^{\mathcal{S}}_s \in \mathbf{ALG}^s_a$.

17.2.19. Definition. The functor $\mathsf{Flt}^s : \mathbf{MSL} \to \mathbf{ALG}^s_a$ is defined as follows.

$$\mathsf{Flt}^s(S) = \langle \mathcal{F}^{\mathcal{S}}_s, \subseteq \rangle.$$

For $f \in \mathbf{MSL}(\mathcal{S}, \mathcal{S}')$ define $\mathsf{Flt}^s(f) \in \mathbf{ALG}^s_a(\mathcal{F}^{\mathcal{S}}_s, \mathcal{F}^{\mathcal{S}'}_s)$ by

$$\begin{aligned} \mathsf{Flt}^s(f)(X) &= \{s' \mid \exists s \in X. f(s) \leq s'\}, \qquad \text{if } X \in \mathcal{F}^{\mathcal{S}}, \\ \mathsf{Flt}^s(f)(\emptyset) &= \emptyset. \end{aligned}$$

17.2.20. Lemma. $\mathcal{K}^s(\mathcal{D}) \in \mathbf{MSL}$ *with*

$$\begin{aligned} d \cap_{\mathcal{K}^s(\mathcal{D})} e &= d \sqcup_{\mathcal{D}} e; \\ \bot_{\mathcal{K}^s(\mathcal{D})} &= \top_{\mathcal{D}}. \end{aligned}$$

Proof. Easy. ■

17.2.21. Definition. Let $\mathcal{D} \in \mathbf{ALG}^s_a$.

(i) The functor $\mathsf{Cmp}_s : \mathbf{ALG}^s_a \to \mathbf{MSL}$ is defined as follows

$$\mathsf{Cmp}_s(\mathcal{D}) = (\mathcal{K}^s(\mathcal{D}), \leq).$$

For a morphism f define $\mathsf{Cmp}_s(f)$ by

$$\mathsf{Cmp}_s(f)(d) = f(d).$$

17.2.22. Proposition. (i) *Let $\mathcal{S}\in\mathbf{MSL}$. Then $\circledast : \mathcal{S}\to\mathcal{K}^s(\mathcal{F}_s^{\mathcal{S}}) = \mathcal{K}(\mathcal{F}^{\mathcal{S}})$ defined by $\circledast(s) = \uparrow s$ is an* **MSL** *isomorphism.*

$$\begin{tikzcd}
\mathcal{S} \arrow[rr, dotted, "\mathsf{Flt}_s"] \arrow[dr, "\uparrow"'] & & \mathcal{F}_s^{\mathcal{S}} \arrow[dl, dotted, "\mathsf{Cmp}_s"] \\
& \mathcal{K}^s(\mathcal{F}_s^{\mathcal{S}}) &
\end{tikzcd}$$

(ii) *Let $\mathcal{D}\in\mathbf{ALG}_a^s$. Then $\ominus : \mathcal{F}_s^{\mathcal{K}^s(\mathcal{D})}\to\mathcal{D}$ defined by $\ominus(X) = \bigsqcup X$, is an $\mathbf{ALG}_a^s$ isomorphism with inverse $\circledcirc : \mathcal{D}\to\mathcal{F}_s^{\mathcal{K}^s(\mathcal{D})}$ defined by*

$$\circledcirc(d) = \{c\in\mathcal{K}^s(\mathcal{D}) \mid c \sqsubseteq d\}.$$

In diagram,

$$\begin{tikzcd}
\mathcal{K}^s(\mathcal{D}) \arrow[dr, dotted, "\mathsf{Flt}_s"'] & & \mathcal{D} \arrow[ll, dotted, "\mathsf{Cmp}_s"'] \\
& \mathcal{F}_s^{\mathcal{K}^s(\mathcal{D})} \arrow[ur, "\bigsqcup"'] &
\end{tikzcd}$$

Proof. Similar to the proof of Proposition 17.2.13. ■

17.2.23. Corollary. *The categories* **MSL** *and* $\mathbf{ALG}_a^s$ *are equivalent.*

17.2.24. Remark. (i) Notice that $\mathcal{K}^s(\mathcal{F}_s^{\mathcal{S}}) = \mathcal{K}(\mathcal{F}^{\mathcal{S}})$.

(ii) The map $\rho : \mathcal{F}_s^{\mathcal{K}^s(\mathcal{D})}\to\mathcal{F}^{\mathcal{K}(\mathcal{D})}$, given by $\rho(X) = X\cup\{\perp_{\mathcal{D}}\}$ is an isomorphism in the category $\mathbf{ALG}_a^s$, hence also in $\mathbf{ALG}_a$.

Summarizing, we have proved in this section the following statements.

$$\begin{array}{rcl}
\mathbf{MSL}^{\top} & \cong & \mathbf{ALG}_a; \\
\mathbf{MSL} & \cong & \mathbf{ALG}_a^s.
\end{array}$$

17.3. Type and zip structures

Now we will extend the equivalences of the previous Section to the various categories of type and zip structures.

Equivalence between $\mathbf{TS}^{\top}$ and $\mathbf{ZS}$

First the functors Flt and Cmp between $\mathbf{MSL}^{\top}$ and $\mathbf{ALG}_a$ will be lifted to act between the richer categories $\mathbf{TS}^{\top}$ and $\mathbf{ZS}$.

17.3.1. Definition. (i) For $\mathcal{S}\in\mathbf{TS}^{\top}$, define $\mathsf{Flt}(\mathcal{S})\in\mathbf{ZS}$ by

$$\mathsf{Flt}(\mathcal{S}) = \langle\mathcal{F}^{\mathcal{S}}, Z^{\mathcal{S}}\rangle,$$

with $Z^{\mathcal{S}} : \mathcal{K}(\mathcal{F}^{\mathcal{S}}) \times \mathcal{K}(\mathcal{F}^{\mathcal{S}}) \to \mathcal{K}(\mathcal{F}^{\mathcal{S}})$ defined by

$$Z^{\mathcal{S}}(\uparrow s, \uparrow t) \ = \uparrow (s \to t).$$

(ii) For $\mathcal{D}_Z = \langle \mathcal{D}, Z \rangle \in \mathbf{ZS}$, define $\mathsf{Cmp}(\mathcal{D}_Z) \in \mathbf{TS}^\top$ by

$$\mathsf{Cmp}(\mathcal{D}_Z) = \langle \mathcal{K}(\mathcal{D}), \leq, \cap, \rightarrow_Z, \top \rangle,$$

with $a \leq b \;\Leftrightarrow\; b \sqsubseteq_{\mathcal{D}} a$, $a \cap b \equiv a \sqcup_{\mathcal{D}} b$, and $\top \equiv \bot_{\mathcal{D}}$, as in Definition 17.2.7 and

$$a \rightarrow_Z b \equiv Z(a, b).$$

(iii) The actions of the maps $\mathsf{Cmp} : \mathbf{ZS} \rightarrow \mathbf{TS}^\top$ and $\mathsf{Flt} : \mathbf{TS}^\top \rightarrow \mathbf{ZS}$ on morphisms are defined as follows. Given $f \in \mathbf{TS}^\top(\mathcal{S}, \mathcal{S}')$, $X \in \mathcal{F}^{\mathcal{S}}$ and $g \in \mathbf{ZS}(\mathcal{D}_Z, \mathcal{D}'_Z)$ with $\mathcal{D}_Z = \langle \mathcal{D}, Z \rangle$, define

$$\begin{array}{rcl} \mathsf{Flt}(f)(X) & = & \{t \mid \exists s \in X. f(s) \leq t\}; \\ \mathsf{Cmp}(g) & = & g \restriction \mathcal{K}(\mathcal{D}). \end{array}$$

17.3.2. Lemma. *If $\mathcal{S} \in \mathbf{TS}^\top$ and $Z^{\mathcal{S}}$ is defined as in Definition 17.3.1(i), then $(\mathcal{K}(\mathcal{F}^{\mathcal{S}}), Z^{\mathcal{S}}) \in \mathbf{ZS}$. Moreover if $\rightarrow_{Z^{\mathcal{S}}}$ is defined for $(\mathcal{K}(\mathcal{F}^{\mathcal{S}}), Z^{\mathcal{S}}) \in \mathbf{ZS}$ as in Definition 17.3.1(ii), then*

$$\uparrow s \rightarrow_{Z^{\mathcal{S}}} \uparrow t = \uparrow(s \rightarrow t).$$

Proof. Immediate from Definition 17.3.1. ■

The following motivates why only one notation is used for the maps $\mathsf{Cmp}, \mathsf{Flt}$.

17.3.3. Proposition. (i) *Let $\mathsf{F}_{\mathrm{ts}} : \mathbf{TS}^\top \rightarrow \mathbf{MSL}^\top$ and $\mathsf{F}_{\mathrm{ls}} : \mathbf{ZS} \rightarrow \mathbf{ALG}$ be the 'forgetful functors' defined by*

$$\begin{array}{ll} \mathsf{F}_{\mathrm{ts}}(\langle \mathcal{S}, \leq, \cap, \rightarrow, \top \rangle) = \langle S, \leq, \cap, \top \rangle, & \mathsf{F}_{\mathrm{ts}}(f) = f; \\ \mathsf{F}_{\mathrm{ls}}(\langle \mathcal{D}, Z \rangle) = \mathcal{D}, & \mathsf{F}_{\mathrm{ls}}(f) = f. \end{array}$$

Then $\mathsf{F}_{\mathrm{ts}}, \mathsf{F}_{\mathrm{ls}}$ are functors indeed.

(ii) *The following figures are commutative diagrams.*

$$\begin{tikzcd} \mathbf{TS}^\top \arrow[r, "\mathsf{Flt}"'] \arrow[d, "\mathsf{F}_{\mathrm{ts}}"'] & \mathbf{ZS} \arrow[d, "\mathsf{F}_{\mathrm{ls}}"] \\ \mathbf{MSL}^\top \arrow[r, "\mathsf{Flt}"'] & \mathbf{ALG}_a \end{tikzcd} \qquad \begin{tikzcd} \mathbf{TS}^\top \arrow[d, "\mathsf{F}_{\mathrm{ts}}"'] & \mathbf{ZS} \arrow[l, "\mathsf{Cmp}"] \arrow[d, "\mathsf{F}_{\mathrm{ls}}"] \\ \mathbf{MSL}^\top & \mathbf{ALG}_a \arrow[l, "\mathsf{Cmp}"] \end{tikzcd}$$

Proof. As usual. ■

17.3.4. Proposition. *Cmp is a functor from $\mathbf{ZS}$ to $\mathbf{TS}^\top$.*

Proof. We just have to prove that Cmp transforms a morphism in $\mathbf{ZS}$ into a morphism into $\mathbf{TS}^\top$. Let $m : \langle \mathcal{D}, Z \rangle \rightarrow \langle \mathcal{D}', Z' \rangle$ be a morphism. By Lemma 17.2.8 we only need to show that $\mathsf{Cmp}(m)$ commutes with arrows. Now

$$\begin{array}{rcll} \mathsf{Cmp}(m)(a \rightarrow_Z b) & = & m(a \rightarrow_Z b), & \text{by definition of } \mathsf{Cmp}(m), \\ & = & m(Z(a, b)), & \text{by definition of } \rightarrow_Z, \\ & = & Z'(m(a), m(b)), & \text{since } m \text{ satisfies } (Z\text{-comm}), \\ & = & m(a) \rightarrow_{Z'} m(b)), & \\ & = & \mathsf{Cmp}(m(a)) \rightarrow_{Z'} \mathsf{Cmp}(m(b)). \;\blacksquare & \end{array}$$

17.3.5. PROPOSITION. *Flt is a functor from $\mathbf{TS}^\top$ to $\mathbf{ZS}$.*

PROOF. We have to prove that Flt transforms a morphism in $\mathbf{TS}^\top$ into a morphism in $\mathbf{ZS}$. Let $f \in \mathbf{TS}^\top(\mathcal{S}, \mathcal{S}')$ with arrows $\to_{Z^\mathcal{S}}$ and $\to_{Z^{\mathcal{S}'}}$ corresponding to $Z^\mathcal{S}$ and $Z^{\mathcal{S}'}$, respectively. By Proposition 17.2.11 we only need to show that $\mathsf{Flt}(f)$ satisfies (Z-comm). Indeed,

$$\begin{array}{rcll} \mathsf{Flt}(f)(Z^\mathcal{S}(\uparrow s, \uparrow t) & = & \mathsf{Flt}(f)(\uparrow (s \to_{Z^\mathcal{S}} t), & \text{by definition of } Z^\mathcal{S}, \\ & = & \uparrow f(s \to_{Z^\mathcal{S}} t), & \text{by Lemma 17.2.10}, \\ & = & \uparrow (f(s) \to_{Z^{\mathcal{S}'}} f(t)), & \text{since } f \in \mathbf{TS}^\top(\mathcal{S}, \mathcal{S}'), \\ & = & Z^{\mathcal{S}'}(\uparrow f(s), \uparrow f(t)), & \text{by definition of } Z^{\mathcal{S}'}, \\ & = & Z^{\mathcal{S}'}(\mathsf{Flt}(f)(\uparrow s), \mathsf{Flt}(f)(\uparrow t)), & \text{by Lemma 17.2.10.} \blacksquare \end{array}$$

Now we will prove that $\mathbf{ZS}$ and $\mathbf{TS}^\top$ are equivalent. To this aim we show that natural isomorphisms $\mathsf{Id}_{\mathbf{TS}^\top} \simeq \mathsf{Cmp} \circ \mathsf{Flt}$ and $\mathsf{Flt} \circ \mathsf{Cmp} \simeq \mathsf{Id}_{\mathbf{ZS}}$ are given by $\circledast$ and $\circleddash$ respectively, exactly as in the case of the equivalence between the categories $\mathbf{MSL}^\top$ and $\mathbf{ALG}_a$.

17.3.6. PROPOSITION. *Let $\mathcal{S}$ be a top type structure. Let $\circledast : \mathcal{S} \to \mathcal{K}(\mathcal{F}^\mathcal{S})$ be the map such that $\circledast(s) = \uparrow s$. Then $\circledast$ is an isomorphism in $\mathbf{TS}^\top$.*

PROOF. By Proposition 17.2.13(i) we only need to show that $\circledast$ and $\circledast^{-1}$ commute with arrows. Now $\circledast$ is a bijection, hence the following suffices.

$$\begin{array}{rcll} \circledast(s \to t) & = & \uparrow (s \to t), & \\ & = & \uparrow s {\to_{Z^\mathcal{S}}} \uparrow t, & \text{by Lemma 17.3.2}, \\ & = & \circledast(s) {\to_{Z^\mathcal{S}}} \circledast(t). \blacksquare & \end{array}$$

17.3.7. PROPOSITION. *Let $\mathcal{D}_Z = \langle \mathcal{D}, Z \rangle \in \mathbf{ZS}$. Define $\circleddash : \mathcal{F}^{\mathcal{K}(\mathcal{D})} \to \mathcal{D}$ by*

$$\circleddash(x) = \bigsqcup x, \quad \text{where the sup is taken in } \mathcal{D}.$$

Then $\circleddash$ is an isomorphism in $\mathbf{ZS}$.

PROOF. By Proposition 17.2.13(ii) we only need to show that $\circleddash$ and $\circledcirc$ satisfy (Z-comm). As to $\circleddash$, we have

$$\begin{array}{rcll} \circleddash(Z^{\mathcal{K}(\mathcal{D})}(\uparrow a, \uparrow b)) & = & \circleddash(\uparrow (a {\to_Z} b)), & \text{by definition of } Z^{\mathcal{K}(\mathcal{D})}, \\ & = & \circleddash(\uparrow Z(a, b)), & \text{by definition of } \to_Z, \\ & = & \bigsqcup(\uparrow Z(a, b)) & \\ & = & Z(a, b) & \\ & = & Z(\bigsqcup \uparrow a, \bigsqcup \uparrow b) & \\ & = & Z(\circleddash(\uparrow a), \circleddash(\uparrow b)). & \end{array}$$

As to $\circledcirc$, we must show

$$\circledcirc(Z(a, b)) = Z^{\mathcal{K}(\mathcal{D})}(\circledcirc(a), \circledcirc(b)).$$

This follows immediately from (Z-comm) for $\circleddash$ and $\circledcirc(a) = \uparrow a$, $\circledcirc(b) = \uparrow b$, $\circleddash \circ \circledcirc = \mathsf{Id}_\mathcal{D}$ and $\circledcirc \circ \circleddash = \mathsf{Id}_{\mathcal{F}^{\mathcal{K}(\mathcal{D})}}$. $\blacksquare$

17.3.8. THEOREM. *The categories* $\mathbf{TS}^\top$ *and* $\mathbf{ZS}$ *are equivalent.*

PROOF. As in Corollary 17.2.15, using Propositions 17.3.4-17.3.7. ∎

Equivalence between $\mathbf{LTS}^\top$ and LZS

17.3.9. PROPOSITION. Cmp *restricts to a functor from* $\mathbf{LZS}$ *to* $\mathbf{LTS}^\top$.

PROOF. Let $\mathcal{D}_Z = \langle \mathcal{D}, Z\rangle \in \mathbf{LLS}$. Then $\mathsf{Cmp}(\mathcal{D}_Z)$ is a LTS$^\top$. In fact, it satisfies $(\rightarrow)$, since

$$\begin{array}{rcll} a' \leq a \;\&\; b \leq b' & \Leftrightarrow & a \sqsubseteq a' \;\&\; b' \sqsubseteq b, & \\ & \Rightarrow & Z(a',b') \sqsubseteq Z(a,b), & \text{by } (Z\text{-contr}), \\ & \Leftrightarrow & a \rightarrow_Z b \leq a' \rightarrow_Z b', & \text{by definition of } \rightarrow_Z. \end{array}$$

$\mathsf{Cmp}(\mathcal{D}_Z)$ satisfies $(\rightarrow\cap)$, since

$$\begin{array}{rcll} (a\rightarrow_Z b_1) \cap (a\rightarrow_Z b_2) & = & (a\rightarrow_Z b_1) \sqcup (a\rightarrow_Z b_2), & \text{since } c \sqcup d = c \cap d, \\ & = & Z(a,b_1) \sqcup Z(a,b_2), & \text{by definition of } \rightarrow_Z, \\ & = & Z(a, b_1 \sqcup b_2), & \text{by } (Z\text{-add}), \\ & = & a\rightarrow_Z (b_1 \sqcup b_2), & \text{by definition of } \rightarrow_Z, \\ & = & a\rightarrow_Z (b_1 \cap b_2), & \text{since } c \sqcup d = c \cap d. \end{array}$$

Finally $\mathsf{Cmp}(\mathcal{D}_Z)$ satisfies $(\top_{\text{lazy}})$, since

$$\begin{array}{rcll} a\rightarrow_Z b & = & Z(a,b), & \\ & \leq & Z(\bot,\bot), & \text{by } (Z\text{-lazy}), \\ & = & \top\rightarrow_Z\top. & \end{array}$$

As to morphisms, we have to prove that $\mathsf{Cmp}(f) \in \mathbf{LTS}^\top(\mathcal{K}(\mathcal{D}), \mathcal{K}(\mathcal{D}'))$, for any $f \in \mathbf{LZS}(\mathcal{D}_Z, \mathcal{D}'_Z)$. Then $\mathsf{Cmp}(f) \in \mathbf{TS}^\top(\mathcal{K}(\mathcal{D}), \mathcal{K}(\mathcal{D}'))$, by Theorem 17.3.4. Finally $\mathbf{LTS}^\top(\mathcal{K}(\mathcal{D}), \mathcal{K}(\mathcal{D}')) = \mathbf{TS}^\top(\mathcal{K}(\mathcal{D}), \mathcal{K}(\mathcal{D}'))$, as $\mathbf{LTS}^\top$ is a full subcategory of $\mathbf{TS}^\top$. ∎

17.3.10. PROPOSITION. Flt *restricts to a functor from* $\mathbf{LTS}^\top$ *to* $\mathbf{LZS}$.

PROOF. Let $\mathcal{S}$ be a LTS$^\top$. Then $\mathsf{Flt}(\mathcal{S})$ is a LZS satisfying (Z-contr), as

$$\begin{array}{rcll} \uparrow a \subseteq \uparrow a' \;\&\; \uparrow b' \subseteq \uparrow b & \Leftrightarrow & a' \leq a \;\&\; b \leq b', & \\ & \Rightarrow & a \rightarrow b \leq a' \rightarrow b', & \text{by } (\rightarrow), \\ & \Leftrightarrow & \uparrow (a' \rightarrow b') \subseteq \uparrow (a \rightarrow b), & \\ & \Leftrightarrow & Z^{\mathcal{S}}(\uparrow a', \uparrow b') \subseteq Z^{\mathcal{S}}(\uparrow a, \uparrow b), & \text{by definition of } Z^{\mathcal{S}}. \end{array}$$

Now $\mathsf{Flt}(\mathcal{S})$ also satisfies (Z-add), since

$$\begin{array}{rcll} Z^{\mathcal{S}}(\uparrow a, \uparrow b_1 \sqcup \uparrow b_2) & = & Z^{\mathcal{S}}(\uparrow a, \uparrow (b_1 \cap b_2)), & \\ & = & \uparrow (a \rightarrow (b_1 \cap b_2)), & \text{by definition of } Z^{\mathcal{S}}, \\ & = & \uparrow ((a \rightarrow b_1) \cap (a \rightarrow b_2)), & \text{by } (\rightarrow \cap), \\ & = & Z^{\mathcal{S}}(\uparrow a, \uparrow b_1) \cap Z^{\mathcal{S}}(\uparrow a, \uparrow b_2))), & \\ & = & Z^{\mathcal{S}}(\uparrow a, \uparrow b_1) \sqcup Z^{\mathcal{S}}(\uparrow a, \uparrow b_2). & \end{array}$$

Finally $\mathsf{Flt}(\mathcal{S})$ satisfies (Z-lazy), since

$$\begin{array}{rcll} Z^{\mathcal{S}}(\bot,\bot) & = & Z^{\mathcal{S}}(\uparrow\top,\uparrow\top), & \\ & = & \uparrow(\top\to\top), & \text{by def of } Z^{\mathcal{S}}, \\ & \subseteq & \uparrow(a\to b), & \text{since } a\to b\le\top\to\top, \text{ by } (\top_{\text{lazy}}), \\ & = & Z^{\mathcal{S}}(\uparrow a,\uparrow b), & \text{by definition of } Z^{\mathcal{S}}. \end{array}$$

As to morphisms, from Theorem 17.2.11 it follows that $\mathsf{Flt}(f)\in\mathbf{ZS}(\mathcal{F}^{\mathcal{S}},\mathcal{F}^{\mathcal{S}'})$, for $f\in\mathbf{TS}^{\top}(\mathcal{S},\mathcal{S}')$. We are done, since $\mathbf{LZS}$ is a full subcategory of $\mathbf{ZS}$, hence $\mathbf{LZS}(\mathcal{F}^{\mathcal{S}},\mathcal{F}^{\mathcal{S}'})=\mathbf{ZS}(\mathcal{F}^{\mathcal{S}},\mathcal{F}^{\mathcal{S}'})$. ■

We will prove that $\mathbf{LTS}^{\top}$ and $\mathbf{LZS}$ are equivalent. To this aim we show that natural isomorphisms $\mathsf{Id}_{\mathbf{LTS}^{\top}}\simeq\mathsf{Cmp}\circ\mathsf{Flt}$ and $\mathsf{Flt}\circ\mathsf{Cmp}\simeq\mathsf{Id}_{\mathbf{LZS}}$ are given by $\circledast$ and $\ominus$.

17.3.11. Proposition. (i) *For $\mathcal{S}\in\mathbf{LTS}^{\top}$ the map $\circledast:\mathcal{S}\to\mathcal{K}(\mathcal{F}^{\mathcal{S}})$ defined by $\circledast(s)=\uparrow s$ is an isomorphism in $\mathbf{LTS}^{\top}$.*

(ii) *For $\langle\mathcal{D},Z\rangle\in\mathbf{LZS}$ the map $\ominus:\mathcal{F}^{\mathcal{K}(\mathcal{D})}\to D$ defined by $\ominus(X)=\sqcup X$ is an isomorphism in $\mathbf{LZS}$.*

Proof. Immediate, since $\circledast$ and $\ominus$ have been proved, in Propositions 17.3.6 and 17.3.7, to be isomorphisms in $\mathbf{TS}^{\top}$ and $\mathbf{ZS}$ respectively. We conclude by the fact that $\mathbf{LTS}^{\top}$ and $\mathbf{LZS}$ are full subcategories. ■

17.3.12. Theorem. *The categories $\mathbf{LTS}^{\top}$ and $\mathbf{LZS}$ are equivalent.*

Proof. As in Corollary 17.2.15, via Propositions 17.3.9, 17.3.10 and 17.3.11. ■

Equivalence between $\mathbf{NTS}^{\top}$ and $\mathbf{NZS}$

17.3.13. Proposition. Cmp *restricts to a functor from $\mathbf{NZS}$ to $\mathbf{NTS}^{\top}$.*

Proof. By Proposition 17.3.9 it follows that, given a $\mathcal{D}_Z\in\mathbf{NZS}$, the map $\mathsf{Cmp}(\mathcal{D}_Z)$ satisfies ($\to$), ($\to\cap$) (and ($\top_{\text{lazy}}$)). We just have to prove that $\mathsf{Cmp}(\mathcal{D}_Z)$ satisfies ($\top\!\to$):

$$\begin{array}{rcll} \top\!\to_Z\!\top & = & Z(\bot,\bot), & \\ & = & \bot, & \text{by } (Z\text{-bot}), \\ & = & \top. ■ & \end{array}$$

17.3.14. Proposition. Flt *restricts to a functor from $\mathbf{NTS}^{\top}$ to $\mathbf{NZS}$.*

Proof. By Proposition 17.3.10 it follows that, given an $\mathcal{S}\in\mathbf{NTS}^{\top}$, the map $\mathsf{Flt}(S)$ satisfies (Z-contr), (Z-add) (and (Z-lazy)). We just have to prove that $\mathsf{Flt}(S)$ satisfies (Z-bot):

$$\begin{array}{rcll} Z^{\mathcal{S}}(\bot,\bot) & = & Z^{\mathcal{S}}(\top,\top), & \\ & = & \uparrow(\top\to\top), & \text{by definition of } Z^{\mathcal{S}}, \\ & = & \uparrow\top, & \text{by } (\top\!\to), \\ & = & \bot. ■ & \end{array}$$

17.3.15. PROPOSITION. (i) *Let* $\mathcal{S} \in \mathbf{NTS}^\top$. *Then* $\circledast \in \mathbf{NTS}^\top(\mathcal{S}, \mathcal{K}(\mathcal{F}^\mathcal{S}))$ *is an isomorphism.*

(ii) *Let* $\langle \mathcal{D}, Z\rangle \in \mathbf{NZS}$. *Then* $\ominus \in \mathbf{NZS}(\mathcal{F}^{\mathcal{K}(\mathcal{D})}, \mathcal{D})$ *is an isomorphism.*

PROOF. As the proof of Proposition 17.3.11: $\circledast$ and $\ominus$ are isomorphisms in $\mathbf{TS}^\top$ and $\mathbf{ZS}$ respectively. We conclude by the fact that $\mathbf{NTS}^\top$ and $\mathbf{NZS}$ are full subcategories. ∎

17.3.16. THEOREM. *The categories* $\mathbf{NTS}^\top$ *and* $\mathbf{NZS}$ *are equivalent.*

PROOF. As in Corollary 17.2.15, via Propositions 17.3.13, 17.3.14 and 17.3.15. ∎

Equivalence between TS and $\mathbf{ZS}_\mathrm{s}$

NOTATION. In the present context, the rest of this section, we use I, J, K for *non-empty* finite sets of indexes. Note that for $\mathcal{S} \in \mathbf{TS}$ we have $\mathcal{K}^s(\mathcal{F}^\mathcal{S}_s) = \mathcal{K}(\mathcal{F}^\mathcal{S})$.

17.3.17. DEFINITION. (i) For $\mathcal{S} \in \mathbf{TS}$, define $\mathsf{Flt}_s(\mathcal{S}) \in \mathbf{ZS}_\mathrm{s}$ by

$$\mathsf{Flt}_s(\mathcal{S}) = (\mathcal{F}^\mathcal{S}_s, Z^\mathcal{S}_s),$$

with $Z^\mathcal{S}_s : \mathcal{K}^s(\mathcal{F}^\mathcal{S}_s) \times \mathcal{K}^s(\mathcal{F}^\mathcal{S}_s) \to \mathcal{K}^s(\mathcal{F}^\mathcal{S}_s)$ defined by

$$Z^\mathcal{S}_s(\uparrow s, \uparrow t) = \uparrow(s \to t).$$

(ii) For $\mathcal{D}_Z = (\mathcal{D}, Z) \in \mathbf{ZS}_\mathrm{s}$, define $\mathsf{Cmp}_s(\mathcal{D}_Z) \in \mathbf{TS}$ by

$$\mathsf{Cmp}_s(\mathcal{D}_Z) = \langle K^s(\mathcal{D}), \leq, \cap, \to_Z\rangle,$$

with $\leq, \cap, \to_Z$ as in Definition 17.3.1.

(iii) The action of the maps Cmp_s and Flt_s on morphisms are defined as follows. Given $f \in \mathbf{TS}(\mathcal{S}, \mathcal{S}')$, $X \in \mathcal{F}^\mathcal{S}$, and $g \in \mathbf{ZS}_\mathrm{s}(\mathcal{D}_Z, \mathcal{D}'_Z)$, define

$$\begin{array}{rcl} \mathsf{Flt}_s(f)(X) & = & \begin{cases} \{t \mid \exists s \in X. f(s) \leq t\}, & \text{if } X \neq \bot \ (= \emptyset), \\ \bot, & \text{else;} \end{cases} \\ \mathsf{Cmp}_s(g) & = & g \restriction \mathcal{K}(\mathcal{D}). \end{array}$$

Note that indeed $\mathsf{Cmp}_s(\mathcal{D}_Z) \in \mathbf{TS}$ by Lemma 17.2.20. The following five Propositions and the Theorem are proved as in 17.3.4-**??** for the $\mathbf{TS}^\top$-case.

17.3.18. PROPOSITION. Cmp_s *is a functor from* $\mathbf{ZS}_\mathrm{s}$ *to* $\mathbf{TS}$.

17.3.19. PROPOSITION. Flt_s *is a functor from* $\mathbf{TS}$ *to* $\mathbf{ZS}_\mathrm{s}$.

It follows that the categories $\mathbf{TS}$ and $\mathbf{ZS}_\mathrm{s}$ are equivalent. Natural isomorphisms are given by $\circledast$ and $\ominus$ respectively.

17.3.20. PROPOSITION. *Let* $\mathcal{S} \in \mathbf{TS}$. *Then* $\circledast : \mathcal{S} \to \mathcal{K}^s(\mathcal{F}_s^{\mathcal{S}})$ *defined by*

$$\circledast(s) \quad = \quad \uparrow s \ ,$$

is an isomorphism in $\mathbf{TS}$.

17.3.21. PROPOSITION. *Let* $(D, Z) \in \mathbf{ZS}_{\mathrm{s}}$. *Then* $\ominus : \mathcal{F}_s^{\mathcal{K}^s(\mathcal{D})} \to \mathcal{D}$ *defined by*

$$\ominus(x) = \bigsqcup x$$

is an isomorphism in $\mathbf{ZS}_{\mathrm{s}}$.

17.3.22. THEOREM. *The categories* $\mathbf{TS}$ *and* $\mathbf{ZS}_{\mathrm{s}}$ *are equivalent.* ■

Equivalence between PTS and PZS^s

17.3.23. DEFINITION. (i) Let $\mathcal{D} \in \mathbf{ALG}$ and $Z : (\mathcal{K}^s(\mathcal{D}) \times \mathcal{K}^s(\mathcal{D})) \to \mathcal{K}^s(\mathcal{D})$. Then $\mathcal{D}_Z = \langle \mathcal{D}, Z \rangle$ is called a *proper strict zip structure*, if one has

$$\begin{array}{ll} (Z\text{-contr}) & a \sqsubseteq a' \ \&\ b' \sqsubseteq b \ \Rightarrow\ Z(a', b') \sqsubseteq Z(a, b); \\ (Z\text{-add}) & Z(a, b_1 \sqcup b_2) = Z(a, b_1) \sqcup Z(a, b_2). \end{array}$$

(ii) The category $\mathbf{PZS}^s$ consists of proper strict zip structures as objects and as morphisms continuous maps $f : \mathcal{D} \to \mathcal{D}'$ satisfying

$$\begin{array}{ll} (\text{cmp-pres}^s) & \forall a \in \mathcal{K}^s(\mathcal{D}).f(a) \in \mathcal{K}^s(\mathcal{D}'); \\ (\text{add}) & \forall X \subseteq D.f(\bigsqcup X) = \bigsqcup f(X); \\ (Z\text{-comm}) & \forall a, b.f(Z(a, b)) = Z'(f(a), f(b)). \end{array}$$

17.3.24. PROPOSITION. Flt_s *restricts to a functor from* $\mathbf{PTS}$ *to* $\mathbf{PZS}^s$.

PROOF. For $S \in \mathbf{PTS}$ one has $\mathsf{Flt}_s(\mathcal{S}) = \langle \mathcal{F}_s^{\mathcal{S}}, Z_s^{\mathcal{S}} \rangle$. As in the proof of Proposition 17.3.10 one can show that $Z_s^{\mathcal{S}}$ satisfies (Z-contr) and (Z-add), hence $\mathsf{Flt}_s(\mathcal{S}) \in \mathbf{PZS}^s$. The proof that Flt_s is well-behaved on morphisms follows as in Proposition 17.3.5. ■

17.3.25. PROPOSITION. Cmp_s *restricts to a functor from* $\mathbf{PZS}^s$ *to* $\mathbf{PTS}$.

PROOF. Let $\mathcal{D} \in \mathbf{PTS}$. Then $\mathsf{Cmp}_s(\mathcal{D}) \in \mathbf{TS}$, by Proposition 17.3.18. As in the proof of Proposition 17.3.9 one shows that $\mathsf{Cmp}_s(\mathcal{D})$ is proper, hence in $\mathbf{PTS}$. The proof that Cmp_s is well-behaved on morphisms follows as in Proposition 17.3.4. ■

The proofs of the following two propositions are similar to the proofs of Propositions 17.3.6 and 17.3.7.

17.3.26. PROPOSITION. *Let* $\mathcal{S} \in \mathbf{PTS}$. *Then* $\circledast : \mathcal{S} \to \mathcal{K}^s(\mathcal{F}_s^{\mathcal{S}})$ *defined by*

$$\circledast(s) \quad = \quad \uparrow s \ ,$$

is an isomorphism in $\mathbf{PTS}$.

17.3.27. PROPOSITION. *Let* $\mathcal{D}_Z = \langle \mathcal{D}, Z\rangle \in \mathbf{PZS}^s$. *Then* $\ominus : \mathcal{F}_s^{\mathcal{K}^s(\mathcal{D})} \to \mathcal{D}$ *defined by*

$$\ominus(x) = \bigsqcup x.$$

is an isomorphism in $\mathbf{PZS}^s$.

It follows that we have an equivalence between categories.

17.3.28. THEOREM. *The categories* $\mathbf{PTS}$ *and* $\mathbf{PZS}^s$ *are equivalent, natural isomorphisms are given by* $\circledast$ *and* $\ominus$ *respectively.*

PROOF. As in Corollary 17.2.15. using Propositions 17.3.26 and 17.3.27. ■

Summarizing we have proved in this section the following equivalences.

$$\begin{array}{rcl}
\mathbf{TS}^{\top} & \cong & \mathbf{ZS}; \\
\mathbf{LTS}^{\top} & \cong & \mathbf{LZS}; \\
\mathbf{NTS}^{\top} & \cong & \mathbf{NZS}; \\
\mathbf{TS} & \cong & \mathbf{ZS}^s; \\
\mathbf{PTS} & \cong & \mathbf{PZS}^s.
\end{array}$$

17.4. From zip to lambda structures

In this section we see that the various categories of zip structures can be related to suitable categories of lambda structures.

Lambda structures

17.4.1. DEFINITION. (i) A *lambda structure*, notation LS, is a triple $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$, where $\mathcal{D} \in \mathbf{ALG}$ and $F : \mathcal{D} \to [\mathcal{D} \to \mathcal{D}]$ and $G : [\mathcal{D} \to \mathcal{D}] \to \mathcal{D}$ are continuous.

(ii) A *strict lambda structure*, notation LS^{s} is a triple $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$, where $\mathcal{D} \in \mathbf{ALG}$, and $F : \mathcal{D} \to_{\mathrm{s}} [\mathcal{D} \to_{\mathrm{s}} \mathcal{D}]$ and $G : [\mathcal{D} \to_{\mathrm{s}} \mathcal{D}] \to_{\mathrm{s}} \mathcal{D}$ are continuous.

A (strict) lambda structure $\mathcal{D}_{F,G}$ can be viewed as an *applicative structure* $\langle \mathcal{D}, \cdot\rangle$, with

$$d \cdot e = F(d)(e).$$

This makes the following definition rather natural.

17.4.2. DEFINITION. Let $\mathcal{D}_{F,G}$ be a lambda structure. A map $f : \mathcal{D} \to \mathcal{D}$ is called *representable* iff for some element $d \in \mathcal{D}$ one has

$$\forall e \in \mathcal{D}.\, f(e) = F(d)(e).$$

Comment: these definitions are given in Section 18.1, where they are used!
Comment: better to use m for the Galois pair?

17.4.3. Definition. A pair of continuous functions $\langle m,n\rangle$ with $m:\mathcal{D}\to\mathcal{D}'$, $n:\mathcal{D}'\to\mathcal{D}$ is said a *Galois connection* (shortly: $\langle m,n\rangle:\mathcal{D}\to\mathcal{D}'$) if

$$\begin{array}{ll}(\text{Galois-1}) & n\circ m \sqsupseteq \mathsf{Id}_{\mathcal{D}};\\ (\text{Galois-2}) & m\circ n \sqsubseteq \mathsf{Id}_{\mathcal{D}'}.\end{array}$$

An equivalent statement is

$$(\text{Galois})\quad \forall x\in\mathcal{D}, x'\in\mathcal{D}'.m(x)\sqsubseteq x' \;\Leftrightarrow x\sqsubseteq n(x').$$

Here m is said to be the *left adjoint* of the Galois connection, n the *right adjoint*.

Next lemma shows two useful properties of the left adjoint in Galois connections.

17.4.4. Lemma. *Let $\langle m,n\rangle:\mathcal{D}\to\mathcal{D}'$ be a Galois connection. Then*
(i) *m is additive.*
(ii) $\forall d\in\mathcal{K}(D).m(d)\in\mathcal{K}(\mathcal{D}')$.

Proof. (i) Of course, being m monotone, it is immediate that for arbitrary sets $\mathcal{I}$ one has

$$m(\bigsqcup_{i\in\mathcal{I}} d_i)\sqsupseteq \bigsqcup_{i\in\mathcal{I}} m(d_i).$$

As to the other inequality

$$\begin{array}{rcll} m(\bigsqcup_{i\in\mathcal{I}} d_i) & \sqsubseteq & m(\bigsqcup_{i\in\mathcal{I}}(n(m(d_i)))), & \text{by (Galois-1)},\\ & \sqsubseteq & m(n(\bigsqcup_{i\in\mathcal{I}} m(d_i))), & \text{since } n \text{ is monotone},\\ & \sqsubseteq & \bigsqcup_{i\in\mathcal{I}} m(d_i), & \text{by (Galois-2)}.\end{array}$$

We now prove (ii). Let $Z'\subseteq D'$ any directed set. We have

$$\begin{array}{rcll} m(d)\sqsubseteq\bigsqcup Z' & \Rightarrow & n(m(d))\sqsubseteq n(\bigsqcup Z'), & \\ & \Rightarrow & n(m(d))\sqsubseteq \bigsqcup n(Z'), & \text{since } n \text{ is continuous},\\ & \Rightarrow & d\sqsubseteq\bigsqcup n(Z'), & \text{by (Galois-1)},\\ & \Rightarrow & \exists z\in Z.d\sqsubseteq n(z), & \text{since } d\in\mathcal{K}(D),\\ & \Rightarrow & \exists z\in Z.m(d)\sqsubseteq m(n(z)) & \\ & \Rightarrow & \exists z\in Z.m(d)\sqsubseteq z & \text{by Galois-2.} \blacksquare\end{array}$$

17.4.5. Definition. A Galois connection $\langle m,n\rangle:\mathcal{D}{\to}\mathcal{D}'$ is called *special* if for any $x,y\in\mathcal{D}$, $x',y'\in\mathcal{D}'$, $f\in[\mathcal{D}\to\mathcal{D}]$ and $g\in[\mathcal{D}\to\mathcal{D}']$ one has

1. $m(G(f))\sqsubseteq G'(m\circ f\circ n)$;
2. $m(F(x)(y))\sqsubseteq F'(m(x))(m(y))$;
3. $n(G'(g))\sqsupseteq G(n\circ g\circ m)$;
4. $n(F'(x')(y'))\sqsupseteq F(n(x))(n(y))$.

17.4.6. Definition. (i) The category **LS** consists of lambda structures as objects and special Galois connections as morphisms. The composition between morphisms $\langle m, n\rangle : \mathcal{D} \to \mathcal{D}'$, $\langle m', n'\rangle : \mathcal{D}' \to \mathcal{D}'$ is given by $\langle m' \circ m, n \circ n'\rangle$.

(ii) The category of *strict* lambda structures, notation $\mathbf{LS}^s$, has as objects strict lambda structures and as morphisms special Galois connections $\langle m, n\rangle$ such that m and n are strict.

(iii) A *proper* lambda structure, notation PLS^s is a strict lambda structure $\mathcal{D}_{F,G}$ such that $\langle G, F\rangle$ is a Galois connection.

(iv) $\mathbf{PLS}^s$ is the full subcategory of $\mathbf{ZS}^s$ having as objects proper lambda structures.

In order to show the relation between **ZS** and **LS**, we will introduce two operators $\mathcal{A}$ and $\mathcal{L}$. Before that, we first show how to build a Galois connection out of a morphism in **ZS**.

17.4.7. Definition. Given a morphism $m : \langle \mathcal{D}, Z\rangle \to \langle \mathcal{D}', Z'\rangle$ in **ZS** we define $n = n_m : \mathcal{D}' \to \mathcal{D}$ by

$$n_m(x') = \bigsqcup\{x \mid m(x) \sqsubseteq x'\}.$$

The next lemma shows properties of n_m (we write just n for short).

17.4.8. Lemma. *Let $m : \langle \mathcal{D}, Z\rangle \to \langle \mathcal{D}', Z'\rangle$ be a morphism in* **ZS**. *Then*

(i) $\forall x \in \mathcal{D}, x' \in \mathcal{D}'.x \sqsubseteq n(x') \;\Leftrightarrow\; m(x) \sqsubseteq x'$.

(ii) *$\langle m, n\rangle$ is a Galois connection between $\mathcal{D}$ and $\mathcal{D}'$:*

$$\begin{aligned} n \circ m &\sqsupseteq \mathsf{Id}_{\mathcal{D}},\\ m \circ n &\sqsubseteq \mathsf{Id}_{\mathcal{D}'}. \end{aligned}$$

(iii) *n is continuous.*

Proof. (i) We have

$$\begin{array}{rcll} x \sqsubseteq n(x') & \Rightarrow & x \sqsubseteq \bigsqcup\{z \mid m(z) \sqsubseteq x'\}, & \text{by definition of } n,\\ & \Rightarrow & m(x) \sqsubseteq \bigsqcup\{m(z) \mid m(z) \sqsubseteq x'\}, & \text{by (add)},\\ & \Rightarrow & m(x) \sqsubseteq x', & \text{by definition of } \bigsqcup,\\ & \Rightarrow & x \sqsubseteq n(x'), & \text{by definition of } n. \end{array}$$

(ii) By (i).

(iii) To show that n is continuous it is enough to prove that for any $X' \subseteq \mathcal{D}'$ directed and $a \in \mathcal{K}(\mathcal{D})$, we have

$$a \sqsubseteq n(\bigsqcup X') \;\Leftrightarrow\; a \sqsubseteq \bigsqcup(n(X')).$$

Now

$$\begin{array}{rcll} a \sqsubseteq n(\bigsqcup X')) & \Leftrightarrow & m(a) \sqsubseteq \bigsqcup X', & \text{by (i)},\\ & \Leftrightarrow & \exists x' \in X'.m(a) \sqsubseteq x', & \text{by (cmp-pres)},\\ & \Leftrightarrow & \exists x' \in X'.a \sqsubseteq n(x'), & \text{by (i)},\\ & \Leftrightarrow & a \sqsubseteq \bigsqcup n(X'), & \text{as } n(X') \text{ is directed. } \blacksquare \end{array}$$

The following is another useful property of n_m.

17.4.9. LEMMA. *Let* $m \in \mathbf{ZS}(\mathcal{D}_Z, \mathcal{D}'_{Z'})$ *and* $a, b \in \mathcal{K}(\mathcal{D})$, *so* $(a \mapsto b) \in [\mathcal{D} \to \mathcal{D}]$. *Then*

(i) $m \circ (a \mapsto b) \circ n = (m(a) \mapsto m(b))$.

(ii) $\forall f \in [\mathcal{D}' \to \mathcal{D}'],\ (m(a) \mapsto m(b)) \sqsubseteq f' \ \Leftrightarrow\ (a \mapsto b) \sqsubseteq n \circ f' \circ m$.

PROOF. (i) For any $y' \in \mathcal{D}'$,

$$\begin{array}{rcll} m \circ (a \mapsto b) \circ n(y') & = & \left\{ \begin{array}{ll} m(b) & \text{if } a \sqsubseteq n(y') \\ m(\bot) & \text{otherwise} \end{array} \right\} & \begin{array}{l} \text{by definition of} \\ \text{step function,} \end{array} \\ & = & \left\{ \begin{array}{ll} m(b) & \text{if } m(a) \sqsubseteq y' \\ \bot & \text{otherwise} \end{array} \right\} & \begin{array}{l} \text{by Lemma 17.4.8(i)} \\ \text{and (add).} \end{array} \end{array}$$

(ii)

$$\begin{array}{rcll} m(a) \mapsto m(b) \sqsubseteq f' & \Leftrightarrow & m \circ (a \mapsto b) \circ n \sqsubseteq f' & \text{by (i)} \\ & \Rightarrow & n \circ m \circ (a \mapsto b) \circ n \circ m \sqsubseteq n \circ f' \circ m & \\ & \Rightarrow & a \mapsto b \sqsubseteq n \circ f' \circ m & \text{since } n \circ m \sqsupseteq \mathsf{Id}_{\mathcal{D}} \\ & \Rightarrow & m \circ (a \mapsto b) \circ n \sqsubseteq m \circ n \circ f' \circ m \circ n & \\ & \Rightarrow & m \circ (a \mapsto b) \circ n \sqsubseteq f' & \text{since } m \circ n \sqsubseteq \mathsf{Id}_{\mathcal{D}'} \\ & \Rightarrow & m(a) \mapsto m(b) \sqsubseteq f' & \text{by (i).} \ \blacksquare \end{array}$$

17.4.10. DEFINITION. Let $\mathcal{D}_Z = \langle \mathcal{D}, Z \rangle \in \mathbf{ZS}$. Then $\mathcal{D}$ induces the continuous functions $\cdot_Z : \mathcal{D}^2 \to \mathcal{D}$, $F_Z : \mathcal{D} \to [\mathcal{D} \to \mathcal{D}]$ and $G_Z : [\mathcal{D} \to \mathcal{D}] \to \mathcal{D}$ defined by

$$\begin{array}{rcl} x \cdot_Z y & = & \bigsqcup \Theta_{\mathcal{D}_Z}(x, y), \\ \text{where } \Theta_{\mathcal{D}_Z}(x, y) & = & \{ b \mid \exists a \sqsubseteq y \mid Z(a, b) \sqsubseteq x \}; \\ F_Z(x) & = & \lambda y. x \cdot_Z y; \\ G_Z(f) & = & \bigsqcup \{ Z(a, b) \mid b \sqsubseteq f(a) \}. \end{array}$$

Often we will omit the subscript $\mathcal{D}_Z$.

In this way we get a lambda structure from Z. Conversely, from a lambda structure $\langle \mathcal{D}, F, G \rangle$ with G compact preserving, one also can define a zip structure $\langle \mathcal{D}, Z \rangle$.

γ was not good for lambda structure, I used $\mathcal{D}_{F,G}$ (macro DDA)**Comment:** I always dislike γ, it does not agree with the symbols for the other structures

17.4.11. DEFINITION. Let $\mathcal{D}_{F,G}$ be a lambda structure with G compact preserving. Then we can define

$$Z_{F,G}(a, b) = G(a \mapsto b).$$

In this way we obtain a zip structure $\langle \mathcal{D}, Z_{F,G} \rangle$.

Of course, in the definition above, we cannot drop the requirement that G preserves compact elements, otherwise $Z_{F,G}$ is not well-defined.

This suggests to introduce the category of *cp-lambda structures*, notation $\mathbf{LS}^s$, which is the full subcategory of $\mathbf{LS}$ which has as objects lambda structures where G preserves compact elements. We will use $\mathbf{LS}^s$ later on.

17.4.12. LEMMA. *Define* $\mathcal{A} : \mathbf{ZS} \to \mathbf{LS}$ *as follows:*

- *let* $\mathcal{D}_Z \in \mathbf{ZS}$*. Define* $\mathcal{A}(\mathcal{D}_Z) = \langle \mathcal{D}, F_Z, G_Z \rangle$;
- *let* $m \in \mathbf{ZS}(\mathcal{D}_Z, \mathcal{D}'_Z)$*. Define* $\mathcal{A}(m) = \langle m, n_m \rangle$.

Then $\mathcal{A}$ *is a functor.*

PROOF. The non-trivial proof concerns morphisms. $\mathcal{A}(m)$ is a Galois connection because of Lemma 17.4.8. So we are left to prove the four points of Definition 17.4.5. Let $f : \mathcal{D}_Z \to \mathcal{D}'_Z$ and $G = G_Z$, $G' = G_{\mathcal{D}'_Z}$.

(1) We have

$$\begin{array}{lll} m(G(f)) = & & \\ = & m(\bigsqcup\{Z(a,b) \mid b \sqsubseteq f(a)\}), & \\ = & \bigsqcup\{m(Z(a,b)) \mid b \sqsubseteq f(a)\}, & \text{by (add)}, \\ = & \bigsqcup\{m(Z(a,b)) \mid a \mapsto b \sqsubseteq f\}, & \\ = & \bigsqcup\{Z'(m(a),m(b)) \mid a \mapsto b \sqsubseteq f\}, & \text{by } (Z\text{-comm}), \\ \sqsubseteq & \bigsqcup\{Z'(m(a),m(b)) \mid m \circ (a \mapsto b) \circ n \sqsubseteq m \circ f \circ n\}, & \\ = & \bigsqcup\{Z'(m(a),m(b)) \mid m(a) \mapsto m(b) \sqsubseteq m \circ f \circ n\}, & \text{by Lemma 17.4.9(i)}, \\ \sqsubseteq & \bigsqcup\{Z'(a',b') \mid a' \mapsto b' \sqsubseteq m \circ f \circ n\}, & \\ = & G'(m \circ f \circ n). & \end{array}$$

(2) We have

$$\begin{array}{llll} m(x \cdot y) & = & m(\bigsqcup\{b \mid \exists a \sqsubseteq y.Z(a,b) \sqsubseteq x\}), & \\ & = & \bigsqcup\{m(b) \mid \exists a \sqsubseteq y.Z(a,b) \sqsubseteq x\}, & \text{by (add)}, \\ & \sqsubseteq & \bigsqcup\{m(b) \mid \exists a \sqsubseteq y.m(Z(a,b)) \sqsubseteq m(x)\}, & \\ & \sqsubseteq & \bigsqcup\{m(b) \mid \exists a \sqsubseteq y.Z'(m(a),m(b)) \sqsubseteq m(x)\} & \text{by } (Z\text{-comm}), \\ & \sqsubseteq & \bigsqcup\{m(b) \mid \exists a.m(a) \sqsubseteq m(y) \;\&\; Z'(m(a),m(b)) \sqsubseteq m(x)\}, & \\ & \sqsubseteq & \bigsqcup\{b' \mid \exists a' \sqsubseteq m(y).Z'(a',b') \sqsubseteq m(x)\}, & \\ & = & m(x) \cdot m(y). & \end{array}$$

(3) We have

$$\begin{array}{llll} G(n \circ g \circ m) & \sqsubseteq & n(m(G(n \circ g \circ m))), & \text{since } n \circ m \sqsupseteq \mathsf{Id}, \\ & \sqsubseteq & n(G'(m \circ n \circ g \circ m \circ n)), & \text{by (1) above} \\ & \sqsubseteq & n(G'(g)), & \text{since } m \circ n \sqsubseteq \mathsf{Id}. \end{array}$$

(4) We have

$$\begin{array}{llll} n(x' \cdot y') & \sqsupseteq & n(m(n(x')) \cdot m(n(y'))), & \text{since } m \cdot n \sqsubseteq \mathsf{Id}, \\ & \sqsupseteq & n(m(n(x') \cdot n(y'))) & \text{by (2) above}, \\ & \sqsupseteq & n(x') \cdot n(y'), & \text{since } n \cdot m \sqsupseteq \mathsf{Id}. \blacksquare \end{array}$$

What about a functor from **LS** to **ZS**? As mentioned, there are problems in going from **LS** to **ZS** due to the fact that G does not preserve compact elements, so it is not clear how to define the induced Z.

We could try to avoid the problem and work with $\mathbf{LS}^s$. Similarly to the definition of $\mathcal{A}$, we could consider the following definition.

17.4.13. Definition. Define $\mathcal{L} : \mathbf{LS}^s \to \mathbf{ZS}$ as follows:

- Let $\langle \mathcal{D}, F, G\rangle \in \mathbf{ZS}^s$. Define $\mathcal{L}(\langle \mathcal{D}, F, G\rangle \in \mathbf{ZS}^s) = \langle \mathcal{D}, Z_{F,G}\rangle$;
- let $\langle m, n\rangle \in \mathbf{ZS}^s(\mathcal{D}_Z, \mathcal{D}'_{Z'})$. Define $\mathcal{L}(\langle m, n\rangle) = m$.

Unfortunately $\mathcal{L}$, even though well-defined on objects, is not a functor. In fact, given a Galois connection $\langle m, n\rangle$, then m satisfies (cmp-pres) and (add), by Lemma 17.4.4, but there is no guarantee that (comm-Z) is satisfied.

Consider for instance

- $\mathcal{D} = \{\perp, \top\}$ ordered by $\perp \sqsubset \top$. Such a structure will be denoted by $\mathcal{D} = \{\perp \sqsubset \top\}$.
- $F, F' : \mathcal{D} \to [\mathcal{D} \to \mathcal{D}]$ defined by:
  $\forall x \in \mathcal{D}. F(x) = \perp \mapsto \perp$;
  $\forall x \in \mathcal{D}. F'(x) = \perp \Rightarrow \top$;
- $G, G' : [\mathcal{D} \to \mathcal{D}] \to \mathcal{D}$ defined by:
  $\forall f \in [\mathcal{D} \to \mathcal{D}]. G(f) = \perp$;
  $\forall f \in [\mathcal{D} \to \mathcal{D}]. G'(f) = \top$.

Then $(\mathsf{Id}_{\mathcal{D}}, \mathsf{Id}_{\mathcal{D}'})$ is a Galois pair between $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ and $\mathcal{D}'_{F',G'} = \langle \mathcal{D}, F', G', \rangle$, but $m = \mathcal{L}(\langle \mathsf{Id}_{\mathcal{D}}, \mathsf{Id}_{\mathcal{D}'}\rangle)$ is not a morphism in $\mathbf{ZS}(\mathcal{L}(\mathcal{D}_{F,G}), \mathcal{L}(\mathcal{D}'_{F',G'}))$.

So leave out for a while morphisms. If we consider just the action of $\mathcal{A}$ and $\mathcal{L}$ on objects, one could ask if the correspondence induced by them is bijective (up to isomorphisms). The answer is negative: the actions of $\mathcal{A}$ and $\mathcal{L}$ on objects are not inverse of each other. By contrast, note that the correspondence will be perfect in the more specialized cases of lazy Galois zip structures and Galois zip structures as we will see below. In those settings $\mathcal{L}$ will be a functor and along with $\mathcal{A}$ will set up an isomorphism of categories.

On one side, Z cannot be recovered by G_Z and F_Z. For instance we can consider the zip structures over $\mathcal{D} = \{\perp \sqsubset a, b \sqsubset \top\}$. Define Z and Z' by

$$
\begin{array}{lll}
Z(a,a) = a; & Z(a,\top) = b; & Z(c,d) = \perp \text{ in the other cases} \\
Z'(a,a) = a; & Z'(a,\top) = \top; & Z'(c,d) = \perp \text{ in the other cases}
\end{array}
$$

Let $\mathcal{D}_Z = \langle \mathcal{D}, Z\rangle$, $\mathcal{D}'_Z = \langle \mathcal{D}, Z'\rangle$. Then $\mathcal{D}_Z$ and $\mathcal{D}'_Z$ induce the same G and F: $G_Z = G_{Z'}$ and $F_Z = F_{Z'}$. So we have proved that $\mathcal{A}(\mathcal{D}_Z) = \mathcal{A}(\mathcal{D}'_Z)$. Since $\mathcal{D}_Z$ and $\mathcal{D}'_Z$ are clearly not isomorphic, $\mathcal{A}$ is not injective (up to isomorphisms).

On the other side, not every cp-lambda structure $\mathcal{D}_{F,G}$ can be obtained by some zip structure $\mathcal{D}_Z$.

Consider for instance $\mathcal{D}$ as any non-trivial ω-algebraic lattice, and let $F : \mathcal{D} \to [\mathcal{D} \to \mathcal{D}]$, $G : [\mathcal{D} \to \mathcal{D}] \to \mathcal{D}$ defined by:

$$\begin{array}{rcl} F(x) & = & \lambda y.\bot \\ G(f) & = & \bot \end{array}$$

Then for no $\mathcal{D}_Z$, we have $F_Z = F$ and $G_Z = G$. In fact,

$$\begin{array}{rcl} F_Z(x)(y) & = & \bigsqcup\{b \mid \exists a \sqsubseteq y.Z(a,b) \sqsubseteq x\} \\ & = & \bigsqcup\{b \mid \exists a \sqsubseteq y.G(a \mapsto b) \sqsubseteq x\} \\ & = & \top. \end{array}$$

So $\mathcal{A}$ is not surjective.

From lazy zip structures to lambda structures

In this subsection we see how the correspondence between zip structures and lambda structures becomes very smooth (an isomorphism of categories) in the case of lazy zip structures.

We start with a remark and a technical lemma. Recall that $\Theta_{\mathcal{D}_Z}$ is defined in Definition 17.4.10.

17.4.14. REMARK. Note that by (Z-contr) and (Z-add) one has in **LZS**

(i) $\forall a \in \mathcal{K}(\mathcal{D}).Z(a,\bot) = Z(\bot,\bot)$.

(ii) $\Theta_{\mathcal{D}_Z}(x,y) \neq \emptyset \;\Leftrightarrow\; \Theta_{\mathcal{D}_Z}(x,y)$ is directed.

17.4.15. LEMMA. *Let $\langle \mathcal{D}, Z\rangle \in$* **LZS** *and let $a, b \in \mathcal{K}(\mathcal{D})$, $x \in \mathcal{D}$, with $b \neq \bot$. Then*

$$b \sqsubseteq x \cdot a \;\Leftrightarrow Z(a,b) \sqsubseteq x.$$

PROOF. ($\Leftarrow$) follows immediately from the definition of application.
We prove ($\Rightarrow$). We have

$$\begin{array}{rcl} b \sqsubseteq x \cdot a & \Leftrightarrow & b \sqsubseteq \bigsqcup \Theta_{\mathcal{D}_Z}(x,a), \\ & \Rightarrow & \exists a_1, b_1.b \sqsubseteq b_1 \;\&\; a_1 \sqsubseteq a \;\&\; Z(a_1,b_1) \sqsubseteq x, \text{ as } \Theta_{\mathcal{D}_Z}(x,a) \text{ is directed,} \\ & \Rightarrow & Z(a,b) \sqsubseteq x. \blacksquare \end{array}$$

This completes the proof. ■

Lazy zip structures admit a characterization by means of *lazy lambda structures*.

Comment: ξ is already used as type environment in chapter 14, the macro for γ is gb.

17.4.16. DEFINITION. Let $\mathcal{D} = \langle \mathcal{D}, F, G\rangle$ be a lambda structure. Write

$$\gamma := G(\bot \Rightarrow \bot).$$

We say that $\mathcal{D}$ is a *lazy* lambda structure if the following holds.

(i)	(γ-comp)	$\gamma \in \mathcal{K}(\mathcal{D})$;
(ii)	(adj1)	$\forall f \in [\mathcal{D} \to \mathcal{D}].F(G(f)) \sqsupseteq f$;
(iii)	(adj2)	$\forall x \in \mathcal{D}.\gamma \sqsubseteq x \;\Rightarrow\; G(F(x)) \sqsubseteq x$;
(iv)	($\gamma\bot$)	$\forall x \in \mathcal{D}.\gamma \not\sqsubseteq x \;\Rightarrow\; F(x) = \bot \mapsto \bot$.

Note that we are using the notation $\langle \mathcal{D}, F, G\rangle$ coherent with lambda structure notation, but this is in contrast with Galois connection notation, since the left adjoint G is put on the right. The following adjunction properties hold.

(ADJ1)	$G(f) \sqsubseteq x \;\Rightarrow\; f \sqsubseteq F(x)$
(ADJ2)	if $\gamma \sqsubseteq x$ then $f \sqsubseteq F(x) \;\Rightarrow\; G(f) \sqsubseteq x$.

In a lambda structure $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$, F and G set up a Galois connection if and only if $\mathcal{D}_{F,G}$ is a lazy lambda structure with $\gamma = \bot$. This is stated in the next lemma.

17.4.17. Lemma. *Let $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ be a lambda structure. Then $\langle G, F\rangle$ is a Galois connection if and only if $\mathcal{D}_{F,G}$ is a lazy lambda structure with $\gamma = \bot$.*

Proof. ($\Rightarrow$) Obviously (Galois-1) is equivalent to (adj1), and (Galois-2) implies (adj2). Moreover

$$\begin{array}{rcll} G(\bot \Rightarrow \bot) & \sqsubseteq & G(F(\bot)), & \\ & \sqsubseteq & \bot, & \text{by (Galois-2).} \end{array}$$

Therefore $\gamma = \bot$. Finally ($\gamma\bot$) is trivial, since the premise $\gamma \not\sqsubseteq x$ is always false.

($\Leftarrow$) If $\gamma = \bot$, then (adj2) becomes (Galois-2). ■

17.4.18. Definition. (i) The category **LLS** is the full subcategory of **LS** which has as objects lazy lambda structures.

(ii) A *natural lambda structure* is an object $\langle D, F, G\rangle$ in **LLS** such that F and G is a Galois connection.

(iii) The category of *natural lambda structures*, notation **NLS**, is the full subcategory of **LS** which has as objects natural lambda structures.

We will establish the following equivalences of catogories.

$$\begin{array}{rcl} \mathbf{LZS} & = & \mathbf{LLS}; \\ \mathbf{NZS} & = & \mathbf{NLS}; \\ \mathbf{PZS}^s & = & \mathbf{PLS}^s. \end{array}$$

17.4.19. Proposition. *Let $\mathcal{D}_Z = \langle \mathcal{D}, Z\rangle$ be a lazy zip structure. Then*

(i) $G_Z(a \mapsto b) = Z(a, b)$.

(ii) $\mathcal{A}(\mathcal{D}_Z) = \langle \mathcal{D}, F_Z, G_Z\rangle$ *is a lazy lambda structure.*

PROOF. (i) We have the following.

$$
\begin{array}{rcll}
G_Z(a{\mapsto}b) & = & \bigsqcup\{Z(a',b') \mid b' \sqsubseteq (a{\mapsto}b)a'\} & \\
& = & \bigsqcup\{Z(a',b') \mid a \sqsubseteq a' \ \&\ b' \sqsubseteq b\}, & \text{because } Z(a',\perp) = Z(\perp,\perp), \\
& & & \text{by Remark 17.4.14(i)}, \\
& = & Z(a,b), & \text{by } (Z\text{-contr}).
\end{array}
$$

(ii) Omitting the subscript $\mathcal{D}_Z$ we prove that $\mathcal{A}(\mathcal{D}_Z)$ satisfies the four points of Definition 17.4.16. We have $G(\perp{\mapsto}\perp) = Z(\perp,\perp)$, by (i). Therefore ($\gamma$-comp) holds, since $Z(\perp,\perp)$ is compact.

As to (adj1), it is sufficient to reason about compact elements, and prove that for any a, b,

$$b \sqsubseteq f(a) \ \Rightarrow b \sqsubseteq F(G(f))(a).$$

Notice that if $b \sqsubseteq f(a)$, then $b \in \Theta_{\mathcal{D}_Z}(G(f), a)$, so

$$
\begin{array}{rcl}
b & \sqsubseteq & \bigsqcup \Theta_{\mathcal{D}_Z}(G(f), a), \\
& = & G(f) \cdot a, \\
& = & F(G(f))(a).
\end{array}
$$

Now we prove (adj2). Suppose $\gamma \sqsubseteq x$, that is $Z(a,\perp) \sqsubseteq x$ for any a. Since $G(F(x)) = \bigsqcup\{Z(a,b) \mid b \sqsubseteq x \cdot a\}$, it is enough to prove that $Z(a,b) \sqsubseteq x$ whenever $b \sqsubseteq x \cdot a$. There are two cases. If $b = \perp$, then the thesis follows from the hypothesis. If $b \neq \perp$, then the thesis follows from Lemma 17.4.15.

Finally we prove ($\gamma\perp$). By (Z-lazy) it follows that $Z(a,b) \not\sqsubseteq x$, for all a, b. So $F(x)(y) = \perp$, for any y. ∎

As a corollary of Proposition 17.4.19 and Lemma 17.4.12 we get the following result.

17.4.20. THEOREM. *$\mathcal{A}$ restricts to a functor from* **LZS** *to* **LLS**.

Going the other direction, from any lazy lambda structure one can define a lazy zip structure. Before showing that, we need to extend Lemma 17.4.4 to lazy lambda structures. The proof is very similar to that other one and is left to the reader.

17.4.21. LEMMA. *Let $\langle \mathcal{D}, F, G\rangle$ be a lazy lambda structure. Then*

(i) *G is additive.*

(ii) $\forall f \in \mathcal{K}([\mathcal{D} \to \mathcal{D}]).G(f) \in \mathcal{K}(\mathcal{D})$.

17.4.22. PROPOSITION. *Let $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ be a lazy lambda structure. Then $\mathcal{L}(\mathcal{D}_{F,G}) = \langle \mathcal{D}, Z_{F,G}\rangle$ is a lazy zip structure.*

PROOF. We omit the subscript $\mathcal{D}$. First of all notice that Z is well-defined since by Lemma 17.4.21(ii), $Z(a,b)$ is a compact element.

We prove (Z-contr). Let $a \sqsubseteq a'$, $b' \sqsubseteq b$. Then, in $[\mathcal{D} \to \mathcal{D}]$, $a'{\mapsto}b' \sqsubseteq a{\mapsto}b$, hence $G(a'{\mapsto}b') \sqsubseteq G(a{\mapsto}b)$. By definition of Z, this implies $Z(a',b') \sqsubseteq Z(a,b)$ as desired.

We prove (Z-add). We have

$$\begin{array}{rcll} Z(a, b_1 \sqcup b_2) & = & G(a \mapsto (b_1 \sqcup b_2)), & \text{by definition,} \\ & = & G(a \mapsto b_1) \sqcup G(a \mapsto b_2), & \text{by Lemma 17.4.21(i).} \end{array}$$

Finally, (Z-lazy) is immediate by monotonicity of G and the fact that $(\perp \Rightarrow \perp) \sqsubseteq (a \mapsto b)$, for all $a, b \in \mathcal{K}(\mathcal{D})$. ■

By contrast to what happened with $\mathbf{ZS}^s$, restricting to **LLS** the functor $\mathcal{L}$ is well-behaved on morphisms, so that we get a functor.

17.4.23. Lemma. *Let* $\pi = \langle m, n \rangle \in \mathbf{LLS}(\mathcal{D}_{F,G}, \mathcal{D}'_{F',G'})$, *where* $\mathcal{D}_{F,G} = \langle D, F, G \rangle$, $\mathcal{D}'_{F',G'} = \langle D', F', G' \rangle$. *Define* $\mathcal{L}(\pi) = m$. *Then* $\mathcal{L}(\pi) \in \mathbf{LZS}(\mathcal{L}(\mathcal{D}_{F,G}), \mathcal{L}(\mathcal{D}'_{F',G'}))$.

Proof. $\mathcal{L}(\pi) = m$ satisfies (cmp-pres) and (add) by Lemma 17.4.21. So we are left to prove that m satisfies (Z-comm), that is, for any $a, b \in K(D)$,

$$m(Z(a, b)) = Z'(m(a), m(b))$$

where $Z = Z_{F,G}$, $Z' = Z_{F,G}$. We have

$$\begin{array}{rcll} m(Z(a, b)) & = & m(G(a \mapsto b)) & \text{by definition of } Z \\ & \sqsubseteq & G'(m \circ (a \mapsto b) \circ n) & \text{by Definition 17.4.5(1)} \\ & = & G'(m(a) \mapsto m(b)) & \text{by Lemma 17.4.9} \\ & = & Z'(m(a), m(b)) & \end{array}$$

On the other hand, being by definition $(a \mapsto b)(a) = b$

$$\begin{array}{rcll} b \sqsubseteq (a \mapsto b)(a) & \Rightarrow & b \sqsubseteq (F(G(a \mapsto b))(a) & \text{by (adj1)} \\ & \Rightarrow & m(b) \sqsubseteq m((F(G(a \mapsto b))(a)) & \\ & \Rightarrow & m(b) \sqsubseteq F'(m(G(a \mapsto b)))(m(a)) & \text{by Definition 17.4.5(2)} \\ & \Rightarrow & m(a) \mapsto m(b) \sqsubseteq F'(m(G(a \mapsto b))) & \\ & \Rightarrow & G'(m(a) \mapsto m(b)) \sqsubseteq m(G(a \mapsto b)) & \text{by (adj2)} \\ & \Rightarrow & Z'(m(a), m(b)) \sqsubseteq m(Z(a, b)) & \end{array}$$

As a consequence of Proposition 17.4.22 and Lemma 17.4.23 we obtain

17.4.24. Theorem. $\mathcal{L} : \mathbf{LLS} \to \mathbf{LZS}$ *is a functor.*

Actually, $\mathcal{L}$ and $\mathcal{A}$ set up an isomorphism between **LLS** and **LZS**. So the correspondence between **LLS** and **LZS** is perfect.

17.4.25. Theorem. (i) $\mathcal{A} \circ \mathcal{L} = \mathsf{Id}_{\mathbf{LLS}}$.
(ii) $\mathcal{L} \circ \mathcal{A} = \mathsf{Id}_{\mathbf{LZS}}$.

Proof. (i) We will prove that for any lazy lambda structure $\mathcal{D}_{F,G} = \langle D, F, G \rangle$, $\mathcal{A}(\mathcal{L}(\mathcal{D}_{F,G})) = \mathcal{D}_{F,G}$. This is equivalent to prove that

$$F_{Z_{F,G}} = F, \quad G_{Z_{F,G}} = G.$$

First we prove that $G_{Z_{F,G}} = G$. We have

$$\begin{array}{rcll} G_{Z_{F,G}}(f) & = & \bigsqcup\{Z_{F,G}(a,b) \mid b \sqsubseteq f(a)\}, & \\ & = & \bigsqcup\{G(a \mapsto b) \mid b \sqsubseteq f(a)\}, & \\ & = & G(\bigsqcup\{a \mapsto b \mid b \sqsubseteq f(a)\} & \text{by Lemma 17.4.21(i)}, \\ & = & G(\bigsqcup\{a \mapsto b \mid a \mapsto b \sqsubseteq f\}), & \\ & = & G(f). & \end{array}$$

Now we prove that for any $x \in \mathcal{D}$, $F_{Z_{F,G}}(x) = F(x)$. First consider the case $\gamma \not\sqsubseteq x$. In such a case $F(x) = \bot \mapsto \bot$ by $(\gamma\bot)$. On the other hand, since $Z_{F,G}$ satisfies (Z-lazy), it follows that $Z_{F,G}(a,b) \not\sqsubseteq x$, for any $a, b \in \mathcal{K}(\mathcal{D})$, so for any $a \in \mathcal{K}(\mathcal{D})$ we have $\Theta_{\mathcal{D}_Z}(x,a) = \emptyset$. This implies $F_{Z_{F,G}}(x)(y) = \bot$ for any $y \in D$, that is $F_{Z_{F,G}}(x) = \bot \mapsto \bot$. So we have proved that, if $\gamma \not\sqsubseteq x$, then $F(x) = F_{Z_{F,G}}(x))$.

Now let $\gamma \sqsubseteq x$. It is enough to prove that for all compact elements a, b one has

$$b \sqsubseteq F(x)(a) \;\Leftrightarrow\; b \sqsubseteq F_{Z_{F,G}}(x)(a).$$

$$\begin{array}{lcll} \text{Indeed,}\;\; b \sqsubseteq F_{Z_{F,G}}(x)(a) & \Leftrightarrow & b \sqsubseteq x \cdot a, & \\ & \Leftrightarrow & Z_{F,G}(a,b) \sqsubseteq x, & \text{by Lemma 17.4.15}, \\ & \Leftrightarrow & G(a \mapsto b) \sqsubseteq x, & \text{by definition of } Z_{F,G}, \\ & \Leftrightarrow & a \mapsto b \sqsubseteq F(x), & \text{by (Galois)}, \\ & \Leftrightarrow & b \sqsubseteq F(x)(a). & \end{array}$$

(ii) We prove that for any lazy zip structure $\mathcal{D}_Z = \langle \mathcal{D}, Z \rangle$, $\mathcal{L}(\mathcal{A}(\mathcal{D}_Z)) = \mathcal{D}_Z$. For this aim, it is enough to prove that $Z_{G_Z} = Z$. We have

$$\begin{array}{rcll} Z_{G_Z}(a,b) & = & G_Z(a \mapsto b) & \\ & = & Z(a,b), & \text{by Proposition 17.4.19(i).} \;\blacksquare \end{array}$$

17.4.26. COROLLARY. *The categories* **LZS** *and* **LLS** *are equivalent.*

From natural zip structures to natural lambda structures

In this short subsection we specialize the results of the previous subsection to natural zip structures.

17.4.27. PROPOSITION. *Let $\mathcal{D}_Z = \langle \mathcal{D}, Z \rangle$ be a natural zip structure. Then $\mathcal{A}(\mathcal{D}_Z) = \langle \mathcal{D}, F_Z, G_Z \rangle$ is a natural lambda structure.*

PROOF. We conclude immediately by Lemma 17.4.17, since $Z(\bot, \bot) = \bot$, hence $G_Z(\bot \mapsto \bot) = \bot$. $\blacksquare$

So we get the following.

17.4.28. THEOREM. *$\mathcal{A}$ restricts to a functor from* **NZS** *to* **NLS**.

We now go the other direction.

17.4.29. Proposition. *Let* $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ *be a natural lambda structure. Then* $\mathcal{L}(\mathcal{D}_{F,G}) = \langle D, Z_{F,G}\rangle$ *is a natural zip structure.*

Proof. By Proposition 17.4.22, we just have to prove that $Z_{F,G}(\perp,\perp) = \perp$. By Lemma 17.4.17 we know that $G(\perp\mapsto\perp) = \perp$, so we are done by the definition of $Z_{F,G}$.

Finally we get the following.

17.4.30. Theorem. $\mathcal{L}$ *restricts to a functor from* **NLS** *to* **NZS**.

17.4.31. Corollary. *The categories* **NLS** *and* **NZS** *are equivalent.*

Proof. From Theorem 17.4.26 and the fact that **NLS** and **NZS** are full subcategories of **LLS** and **LZS** respectively. ■

Comment: this was at page 107

17.4.32. Lemma. *Let* $\mathcal{D}_{F,G} \in$ **NLS**.

(i) *For* $f \in [\mathcal{D}\to\mathcal{D}]$

$$f \text{ is compact } \Rightarrow G(f) \text{ is compact.}$$

(ii) *For* $f_1, f_2 \in [\mathcal{D}\to\mathcal{D}]$

$$G(f_1 \sqcup f_2) = G(f_1) \sqcup G(f_2).$$

(iii) F *determines* G*: for all* $f \in [\mathcal{D}\to\mathcal{D}]$

$$G(f) = \sqcap\{d \mid f \sqsubseteq F(d)\}.$$

Proof. (i) Suppose f is compact. Then

$$\begin{array}{rcll}
G(f) \sqsubseteq \bigsqcup_{z\in Z} z & \Rightarrow & F(G(f)) \sqsubseteq \bigsqcup_{z\in Z} F(z), & \text{since } F \text{ is continuous,}\\
& \Rightarrow & f \sqsubseteq \bigsqcup_{z\in Z} F(z), & \text{since } \mathcal{D} \text{ is Galois,}\\
& \Rightarrow & \exists z \in Z. f \sqsubseteq F(z), & \text{since } f \text{ is compact}\\
& & & \text{and } F(Z) \text{ is directed,}\\
& \Rightarrow & \exists z \in Z. G(f) \sqsubseteq G(F(z)), & \text{since } G \text{ is monotonic,}\\
& \Rightarrow & \exists z \in Z. G(f) \sqsubseteq z, & \text{since } \mathcal{D} \text{ is Galois.}
\end{array}$$

(ii) Since $G(f_i) \sqsubseteq G(f_1 \sqcup f_2)$, by monotonicity of G, one has $G(f_1)\sqcup G(f_2) \sqsubseteq G(f_1 \sqcup f_2)$. Conversely,

$$\begin{array}{rcll}
G(f_1 \sqcup f_2) & \sqsubseteq & G[F(G(f_1)) \sqcup F(G(f_2))], & \text{since } F\circ G \sqsupseteq \mathbf{1}_{[\mathcal{D}\to\mathcal{D}]},\\
& \sqsubseteq & G(F[G(f_1) \sqcup G(f_2)]), & \text{by monotonicity of } F,\\
& \sqsubseteq & G(f_1) \sqcup G(f_2), & \text{since } G\circ F \sqsubseteq \mathbf{1}_{\mathcal{D}}.
\end{array}$$

(iii) It suffices to show that $G(f) \sqsubseteq d \Leftrightarrow f \sqsubseteq F(d)$. Indeed,

$$\begin{array}{rcll}
G(f) \sqsubseteq d & \Rightarrow & F(G(f)) \sqsubseteq F(d), & \text{since } F \text{ is monotonic}\\
& \Rightarrow & f \sqsubseteq F(d), & \text{as } \mathbf{1}_{[\mathcal{D}\to\mathcal{D}]} \sqsubseteq F\circ G,\\
& \Rightarrow & G(f) \sqsubseteq G(F(d)), & \text{since } G \text{ is monotonic,}\\
& \Rightarrow & G(f) \sqsubseteq d & \text{as } G\circ F \sqsubseteq \mathbf{1}_{\mathcal{D}}. \ ■
\end{array}$$

From proper strict zip structures to proper strict lambda structures

In this final subsection we specialize the previous results to proper strict zip structures. Most proofs of this section are left to the reader.

17.4.33. Definition. Let $\mathcal{D}_Z = \langle \mathcal{D}, Z\rangle \in \mathbf{ZS}^s$. Then $\mathcal{D}_Z$ induces the continuous maps $\cdot_Z : \mathcal{D}^2 \to \mathcal{D}$, $F_Z : \mathcal{D} \to_{\mathrm{s}} [\mathcal{D} \to_{\mathrm{s}} \mathcal{D}]$ and $G_Z : [\mathcal{D} \to_{\mathrm{s}} \mathcal{D}] \to_{\mathrm{s}} \mathcal{D}$ defined by

$$\begin{array}{rcl} x \cdot_Z y & = & \bigsqcup \Theta_{\mathcal{D}_Z}(x,y), \\ \text{where } \Theta_{\mathcal{D}_Z}(x,y) & = & \{b \mid \exists a \sqsubseteq y. Z(a,b) \sqsubseteq x\}; \\ F_Z(x) & = & \lambda y. x \cdot_Z y; \\ G_Z(f) & = & \bigsqcup \{Z(a,b) \mid b \sqsubseteq f(a)\}. \end{array}$$

Usually we will omit the Z.

The following Lemma shows that indeed $\langle \mathcal{D}, F_Z, G_Z\rangle \in \mathbf{ZS}^s$.

17.4.34. Lemma. (i) *For all $x \in \mathcal{D}$ the map $F_Z(x)$ is strict.*
(ii) *F_Z is strict.*
(iii) *G_Z is strict.*

Proof. (i) Since $\langle \bot, b\rangle$ is not in the domain of Z.
(ii) Since $Z(a,b) \neq \bot$.
(iii) If $b \sqsubseteq (\bot \Rightarrow \bot)(a)$, then $b = \bot$. Moreover, $\langle a, \bot\rangle$ is not in the domain of Z. ■

17.4.35. Remark. (i) In the proof of $\mathcal{A}^s \circ \mathcal{L}^s = \mathsf{Id}_{\mathbf{PLS}^s}$ use that Remark 17.4.14(ii) and Lemma 17.4.15 also hold for $\mathbf{PLS}^s$ and $a, b \in \mathcal{K}^s(\mathcal{D})$.
(ii) If m is a morphism in $\mathbf{ZS}^s$, then m is strict by (add). Moreover, if $m(d) = \bot$, then $d = \bot$, hence $n(\bot) = \bot$ holds for n of Definition 17.4.7. So m and n are both strict, hence $\langle m, n\rangle$ is a morphism in $\mathbf{LS}^{\mathrm{s}}$.

From a proper strict lambda structure $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ one also can define a proper strict zip structure $\langle \mathcal{D}, Z_{F,G}\rangle$, by setting for any $a, b \in \mathcal{K}^s(\mathcal{D})$

$$Z_{F,G}(a,b) = G(a \mapsto b).$$

Note that from the strictness of F and the fact that $\langle G, F\rangle$ is a Galois connection it follows that $G(a \mapsto b) \neq \bot$ if $a, b \neq \bot$. Hence $Z_{F,G}(a,b)$ is well defined.

These actions allow to define in the obvious way two operators:

$$\begin{array}{c} \mathcal{A}^s : \mathbf{PZS}^s \to \mathbf{PLS}^s \\ \mathcal{L}^s : \mathbf{PLS}^s \to \mathbf{PZS}^s \end{array}$$

Actually both $\mathcal{A}^s$ and $\mathcal{L}^s$ are functors:

17.4.36. Theorem. *$\mathcal{A}^s$ is a functor from $\mathbf{PZS}^s$ to $\mathbf{PLS}^s$.*

17.4.37. Theorem. *$\mathcal{L}^s$ is a functor from $\mathbf{PLS}^s$ to $\mathbf{PZS}^s$.*

17.4.38. Theorem. *The categories $\mathbf{PZS}^s$ and $\mathbf{PLS}^s$ are equivalent.*

Comment: I propose to escape the following section, since the call-by-value is treated in Simona book.

17.5. Type structures with top for arrow types

17.5.1. Definition. (i) Let $\mathcal{S} \in \mathbf{GTS}^{!}$, see Definition ??. Let $\mathbb{T}^{\mathcal{S}}$ contain a special atom ν. Then we say that $\mathcal{S}$ is a ν-strict TT ($\mathrm{GTS}^{!\nu}$) if it satisfies

$$\boxed{(\nu) \quad A{\to}B \leq \nu}$$

(ii) The category $\mathbf{GTS}^{!\nu}$ is the full subcategory of $\mathbf{GTS}^{!}$ with $\mathrm{GTS}^{!\nu}$ as objects.

(iii) Let $\mathcal{S}$ be in $\mathbf{GTS}^{!\nu}$. Then $\lambda{\cap}^{\mathcal{S}}_{\nu}$ denotes the type assignment system with the axioms and rules of $\lambda^{\mathcal{T}}_{\cap}$, see Definition 15.2.2, extended with

$$\boxed{(\nu \text{ universal}) \quad \Gamma \vdash \lambda x.M : \nu}$$

We write $\vdash^{\mathcal{S}}_{\cap\nu}$, or simply $\vdash^{\mathcal{S}}$, $\vdash^{\nu}$ or $\vdash$, for derivability in this system.

In this section all TSs are ν-strict and all type assignment systems have rule (ν-universal). Note that ν-strict type structures are also strict natural type structures.

The prime example of a ν-strict TS is the syntactic intersection type structure EHR coming from the intersection type theory EHR with $\mathbb{A} = \{\nu\}$ axiomatized by (η), $(\to\cap)$, (ν).

Most definitions, propositions and theorems for **NTS**s and $\lambda_{\cap(\Omega)}$ hold in a slightly modified form for the present setting. We give some of them, with short indications of proofs. We will also indicate some differences.

Clearly Lemma 15.2.6(ii) and Proposition 15.2.8 hold for $\lambda^{\mathcal{S}}_{\cap\nu}$.

Also the Generation Theorem 16.1.1 holds for $\lambda^{\mathcal{S}}_{\cap\nu}$, without the condition $A \neq \Omega$ in (i) and (ii) and with the extra condition $\nu \not\leq A$ in (iii).

17.5.2. Definition. $\mathcal{S} \in \mathbf{GTS}^{!\nu}$ is called *ν-sound* if for all $A, B \in \mathcal{S}$

$$\nu \not\leq (A{\to}B).$$

17.5.3. Theorem (Inversion Theorem for $\lambda^{\mathcal{S}}_{\cap\nu}$). *Let $\mathcal{S} \in \mathbf{GTS}^{!\nu}$. Then in $\lambda^{\mathcal{S}}_{\cap\nu}$ the following hold provided that in (iii) $\mathcal{S}$ is β-sound and ν-sound and $C \neq \nu$.*

$$\begin{array}{lrcl}
\text{(i)} & \Gamma, x{:}A \vdash x : B & \Leftrightarrow & A \leq B. \\
\text{(ii)} & \Gamma \vdash (MN) : A & \Leftrightarrow & \exists B \in \mathcal{S}\,[\Gamma \vdash M : (B{\to}A)\ \&\ \Gamma \vdash N : B]. \\
\text{(iii)} & \Gamma \vdash (\lambda x.M) : (B{\to}C) & \Leftrightarrow & \Gamma, x{:}B \vdash M : C.
\end{array}$$

Proof. Similar to the proof of Theorem 16.1.10. ■

17.5.4. Proposition. EHR *is β-sound and ν-sound.*

Proof. (β-soundness) Similar to the proof of β-soundness of CDV in Theorem 16.1.8.

(ν-soundness) By induction on the definition of $\leq_{\mathrm{EHR}}$ one shows

$$\nu \leq A \;\Rightarrow\; A \equiv \nu \cap \ldots \cap \nu.$$

The result follows. ■

17.5.5. Corollary. *For* EHR *Theorem 17.5.3(iii) holds unconditionally.*

17.5.6. Remark. One can show that $\lambda_{\cap\nu}^{\mathrm{EHR}}$ satisfies β**I**-red, but not β**I**-exp and η-exp but not η-red. Indeed, $\vdash_{\nu}^{\mathrm{EHR}} \lambda x.z : \nu$, but $\not\vdash_{\nu}^{\mathrm{EHR}} (\lambda yx.y)z : \nu$; moreover $\vdash_{\nu}^{\mathrm{EHR}} \lambda y.xy : \nu$, but $\not\vdash_{\nu}^{\mathrm{EHR}} x : \nu$.

$\mathbf{NLS}^{\mathrm{s}\nu}$ and $\mathbf{GTS}^{\mathrm{l}\nu}$

17.5.7. Definition. Let $\mathcal{D}$ be an ω-algebraic lattice.

(i) Let $Z : (\mathcal{K}^s(\mathcal{D}) \times \mathcal{K}^s(\mathcal{D})) \cup \{\langle \bot, \bot \rangle\} \to \mathcal{K}^s(\mathcal{D})$. Then $\mathcal{D}_Z = (D, H)$ is called a *ν-strict lambda structure* (SLS^ν), if it satisfies the following conditions.

$$
\begin{array}{ll}
(Z\text{-contr}) & a \sqsubseteq a' \ \&\ b' \sqsubseteq b \ \Rightarrow\ Z(a', b') \sqsubseteq Z(a, b); \\
(Z\text{-add}) & Z(a, b_1 \sqcup b_2) = Z(a, b_1) \sqcup Z(a, b_2); \\
(Z\text{-lazy}) & Z(\bot, \bot) \sqsubseteq Z(a, b).
\end{array}
$$

(ii) The category $\mathbf{NLS}^{\mathrm{s}\nu}$ consists of SLS^νs as objects and as morphisms continuous functions which satisfy

$$
\begin{array}{ll}
(\text{cmp-pres}) & \forall a. m(a) \in \mathcal{K}^s(\mathcal{D}'); \\
(\text{add}) & \forall X \subseteq D. m(\bigsqcup X) = \bigsqcup m(X); \\
(\text{comm-H}) & \forall a, b. m(Z(a, b)) = Z'(m(a), m(b)).
\end{array}
$$

Application is defined similarly to the non-strict case.

17.5.8. Definition. Let $\mathcal{D}_Z = (D, Z) \in \mathbf{NLS}^{\mathrm{s}\nu}$. Then $\mathcal{D}_Z$ induces the continuous functions $\cdot_H : D^2 \to D$, $F_H : D \to^s [D \to^s D]$ and $G_H : [D \to^s D] \to D$ defined by

$$
\begin{array}{rcl}
x \cdot_H y & = & \bigsqcup \Theta_{\mathcal{D}_Z}(x, y), \\
\text{where } \Theta_{\mathcal{D}_Z}(x, y) & = & \{b \mid \exists a \sqsubseteq y \mid Z(a, b) \sqsubseteq x\}; \\
F_H(x) & = & \lambda y. x \cdot_H y; \\
G_H(f) & = & \bigsqcup \{Z(a, b) \mid b \sqsubseteq f(a)\}.
\end{array}
$$

Usually we will omit the H. $F_Z(x)$ is strict, since $(\bot, b)$ is in the domain of H only if $b = \bot$ and F_Z is strict, since $Z(a, b) \neq \bot$.

17.5.9. Remark. Note that by (H-contr) and (H-add) we have for $\Theta_{\mathcal{D}_Z}$

$$\Theta_{\mathcal{D}_Z}(x, y) \neq \emptyset \ \Leftrightarrow\ \Theta_{\mathcal{D}_Z}(x, y) \text{ is directed.}$$

17.5.10. Lemma. *Let $\mathcal{D}_Z = (D, Z)$ be an* SLS^ν *and let $a, b \in \mathcal{K}^s(\mathcal{D}_Z)$, $x \in D$. Then*

$$b \sqsubseteq x \cdot a \ \Leftrightarrow Z(a, b) \sqsubseteq x$$

Proof. As for Lemma 17.4.15. ∎

The ν-strict lambda structures admit a characterization by means of *strict lazy Galois connections.*

17.5.11. Definition. Let $\mathcal{D} \in \mathbf{ALG}$. A *semi strict applicative structure* is a triple $(\mathcal{D}, F, G)$ with $F : [[\mathcal{D} \to^s \mathcal{D}] \to^s \mathcal{D}]$ and $G : \mathcal{D} \to^s [\mathcal{D} \to^s \mathcal{D}]$ continuous.

17.5.12. Definition. Let $\langle D, F, G\rangle$ be a semi strict applicative structure. Write $\xi := G(\perp \Rightarrow \perp)$. We say that $\langle D, F, G\rangle$ is a *strict lazy* Galois connection if the following conditions are fullfilled.

(ξ-comp)	$\xi \in \mathcal{K}(\mathcal{D})$;
(adj1)	$\forall f \in [D \to^s D].F(G(f)) \sqsupseteq f$;
(adj2)	$\forall x \in D.\xi \sqsubseteq x \ \Rightarrow\ G(F(x)) \sqsubseteq x$;
($\xi\perp$)	$\forall x \in D.\xi \not\sqsubseteq x \ \Rightarrow\ F(x) = \perp \Rightarrow \perp$.

17.5.13. Proposition. *Let* (D, Z) *be an* SLS$^\nu$.

(i) $\forall a, b \in \mathrm{dom}(Z).G_H(a \Rightarrow b) = H(a, b)$.

(ii) $\langle D, F_H, G_H\rangle$ *is a strict lazy Galois connection.*

Proof. Similar to the proof of Proposition 17.4.19. In the proof of (i) we use the fact that $Z(\perp,\perp) \sqsubseteq Z(a, b)$, by ($Z$-lazy). In (ii) we get ($\xi$-comp) from $Z(\perp,\perp) \in \mathcal{K}^s(\mathcal{D})$. ■

Going the other direction, from any strict lazy Galois connection one can define an LLS. Before showing that, we need a preliminary lemma.

17.5.14. Lemma. *Let* $\langle D, F, G\rangle$ *be a strict lazy Galois connection. Then*

(i) *G is additive.*

(ii) $\forall f \in \mathcal{K}([D \to D]).G(f) \in \mathcal{K}^s(\mathcal{D})$.

Proof. Similar to the proof of Lemma 17.4.4. In (ii) we must show additionally $G(f) \neq \perp$. This follows by monotonicity of G from $G(\perp \Rightarrow \perp) = \xi \neq \perp$, by ξ-comp. ■

17.5.15. Proposition. *Let* $\gamma = \langle D, F, G\rangle$ *be a strict lazy Galois connection. Define* $Z_\gamma : \mathcal{K}^s(\mathcal{D}) \times \mathcal{K}^s(\mathcal{D}) \cup \{\perp,\perp\} \to \mathcal{K}^s(\mathcal{D})$ *by*

$$Z_\gamma(a, b) = G(a \Rightarrow b)$$

Then (D, Z_γ) *is a ν-strict lambda structure.*

Proof. As the proof of Proposition 17.4.22. By Lemma 17.5.14(ii) we get $Z_\gamma(a, b) \in \mathcal{K}^s(\mathcal{D})$. ■

The next theorem shows that SLS$^\nu$'s can be characterized completely as strict lazy Galois connections, and vice-versa.

17.5.16. Theorem. (i) *Let* (D, Z) *be a* SLS$^\nu$. *Let* $\gamma = \langle D, F_Z, G_Z\rangle$. *Then* $Z_\gamma = H$.

(ii) *Let* $\gamma = \langle D, F, G\rangle$ *be a strict lazy Galois connection. Then* $F_{Z_\gamma} = F$, $G_{Z_\gamma} = G$.

Proof. Similar to the proof of Theorem ??. ■

17.5.17. DEFINITION. Given an object $\mathcal{S}$ in $\mathbf{GTS}^{\mathsf{T}\nu}$, we define the strict lambda structure $\mathsf{Flt}_\nu(S) = (\mathcal{F}_s^{\mathcal{S}}, Z_\nu^{\mathcal{S}})$, where $Z_\nu^{\mathcal{S}} : \mathcal{K}^s(\mathcal{F}_s^{\mathcal{S}}) \times \mathcal{K}^s(\mathcal{F}_s^{\mathcal{S}}) \cup \{(\emptyset, \emptyset)\} \to \mathcal{K}^s(\mathcal{F}_s^{\mathcal{S}})$ is defined as follows.

$$\begin{array}{rcl} Z_\nu^{\mathcal{S}}(\uparrow^s s, \uparrow^s t) & = & \uparrow^s (s \to t) \\ Z_\nu^{\mathcal{S}}(\emptyset, \emptyset) & = & \uparrow^s \nu. \end{array}$$

17.5.18. PROPOSITION. *Let $S \in \mathbf{GTS}^{\mathsf{T}\nu}$. Then $\langle \mathcal{F}_s^{\mathcal{S}}, Z_\nu^{\mathcal{S}} \rangle$ is an $\mathbf{NLS}^{s\nu}$.*

PROOF. By Proposition 15.4.6 $\mathcal{F}_s^{\mathcal{S}}$ is in $\mathbf{MSL}^s$. The map $Z_\nu^{\mathcal{S}}$ satisfies (H-contr) and (H-add) as in the proof of Proposition 17.3.10. The extra cases needed to show that $(\perp, \perp) \in \mathrm{dom}(Z)$ are easy. As to (Z-lazy).

$$\begin{array}{rcll} Z_\nu^{\mathcal{S}}(\perp, \perp) & = & Z_\nu^{\mathcal{S}}(\emptyset, \emptyset) & \\ & = & \uparrow^{\mathcal{S}}(s \to t), & \text{by } (\nu), \\ & = & Z_\nu^{\mathcal{S}}(\uparrow^{\mathcal{S}} s, \uparrow^{\mathcal{S}} t). \blacksquare & \end{array}$$

17.5.19. DEFINITION. Let $\mathcal{S} \in \mathbf{GTS}^{\mathsf{T}\nu}$. Define

(i) $F_\nu^{\mathcal{S}} : \mathcal{F}_s^{\mathcal{S}} \to^s [\mathcal{F}_s^{\mathcal{S}} \to^s \mathcal{F}_s^{\mathcal{S}}]$ and $G_\nu^{\mathcal{S}} : [\mathcal{F}_s^{\mathcal{S}} \to^s \mathcal{F}_s^{\mathcal{S}}] \to^s \mathcal{F}_s^{\mathcal{S}}$ by $F_\nu^{\mathcal{S}} = F_{Z_\nu^{\mathcal{S}}}$, $G_\nu^{\mathcal{S}} = G_{Z_\nu^{\mathcal{S}}}$, that is:

$$\begin{array}{rcl} F_\nu^{\mathcal{S}}(X)(Y) & = & \uparrow^s \{t \in \mathcal{S} \mid \exists s \in Y. s \to t \in X\}; \\ G_\nu^{\mathcal{S}}(f) & = & \uparrow^s \{s \to t \mid t \in f(\uparrow^s s)\} \cup \uparrow^s \nu. \end{array}$$

(ii) The triple $\langle \mathcal{F}_\nu^{\mathcal{S}}, F_\nu^{\mathcal{S}}, G_\nu^{\mathcal{S}} \rangle$ is called the *ν-strict filter structure* over $\mathcal{S}$.

17.5.20. DEFINITION. (i) Given an object $(\mathcal{D}, Z) \in \mathbf{NLS}^{s\nu}$, we define the type structure $\mathsf{Cmp}_\nu(\mathcal{D}) = (K^s(\mathcal{D}), \leq, \cap, \nu, \to_Z)$, with

- $a \leq b \;\Leftrightarrow\; b \sqsubseteq a$;
- $a \cap b \equiv a \sqcup b$;
- $a \to_Z b \equiv Z(a, b)$;
- $\nu = Z(\perp, \perp)$.

(ii) The maps Cmp_ν and Flt_ν can be extended to morphisms as follows. Given $m \in \mathbf{LS}(\mathcal{D}, \mathcal{D}')$ and $f \in \mathbf{TS}(S, S')$, define

$$\begin{array}{rcl} \mathsf{Cmp}_\nu(m) & = & m \restriction \mathcal{K}(\mathcal{D}); \\ \mathsf{Flt}_\nu(f)(X) & = & \{t \mid \exists s \in X. f(s) \leq t\}. \end{array}$$

We now will prove the equivalence between the categories $\mathbf{NLS}^{s\nu}$ and $\mathbf{GTS}^{\mathsf{T}\nu}$.

17.5.21. PROPOSITION. *Cmp_s is a functor from $\mathbf{NLS}^s$ to $\mathbf{GTS}^{\mathsf{T}}$.*

PROOF. For any $\mathcal{D} \in \mathbf{NLS}^s$, $\mathsf{Cmp}_s(\mathcal{D})$ is a $\mathrm{GTS}^{\mathsf{T}}$ by Proposition ??. The proof that Cmp_s is well-behaved on morphisms follows as in the non-strict case (see Proposition 17.2.11). $\blacksquare$

17.5.22. PROPOSITION. *Flt_ν is a functor from $\mathbf{GTS}^{\mathsf{T}\nu}$ to $\mathbf{NLS}^{s\nu}$.*

PROOF. For any $\mathcal{S}\in\mathbf{GTS}^{\mathsf{l}}$, $\mathsf{Flt}_\nu(\mathcal{S})$ is an SLS^ν by Proposition 17.5.18. The proof that Flt_ν is well-behaved on morphisms follows as in the non-strict case (see Proposition 17.3.6). ■

17.5.23. PROPOSITION. *Let $\mathsf{F}_{\mathrm{ts}} : \mathbf{GTS}^{\mathsf{l}^\nu}\to\mathbf{MSL}^s$ and $\mathsf{F}_{\mathrm{ls}} : \mathbf{NLS}^{\mathrm{s}\nu}\to\mathbf{ALG}^s$ be the forgetful functors. Then the following figures are commutative diagrams.*

$$\begin{tikzcd}
\mathbf{GTS}^{\mathsf{l}} \arrow[r, "\mathsf{Flt}"'] \arrow[d, "\mathsf{F}_{\mathrm{ts}}"'] & \mathbf{NLS}^{\mathrm{s}} \arrow[d, "\mathsf{F}_{\mathrm{ls}}"] \\
\mathbf{MSL}^s \arrow[r, "\mathsf{Flt}"'] & \mathbf{ALG}^s
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{GTS}^{\mathsf{l}} \arrow[d, "\mathsf{F}_{\mathrm{ts}}"'] & \mathbf{NLS}^{\mathrm{s}} \arrow[l, "\mathsf{Cmp}"] \arrow[d, "\mathsf{F}_{\mathrm{ls}}"] \\
\mathbf{MSL}^s & \mathbf{ALG}^s \arrow[l, "\mathsf{Cmp}"]
\end{tikzcd}$$

It follows that the categories $\mathbf{NLS}^{\mathrm{s}\nu}$ and $\mathbf{GTS}^{\mathsf{l}^\nu}$ are equivalent. Natural isomorphisms are given by ξ and τ respectively.

The proof of the following two propositions are similar to the proofs of Propositions 17.3.6 and 17.3.7 in the non-strict case.

17.5.24. PROPOSITION. *Let $\mathcal{S}\in\mathbf{GTS}^{\mathsf{l}^\nu}$. Then $\xi_{\mathcal{S}} : \mathcal{S}\to\mathcal{K}^s(\mathcal{F}_\nu^{\mathcal{S}})$ defined as*

$$\xi_{\mathcal{S}}(s) \;=\; \uparrow s\ ,$$

is an isomorphism in $\mathbf{GTS}^{\mathsf{l}^\nu}$.

17.5.25. PROPOSITION. *Let $(D,Z)\in\mathbf{NLS}^{\mathrm{s}\nu}$. Then $\tau_{\mathcal{D}} : \mathcal{F}_\nu^{\mathcal{K}^s(\mathcal{D})}\to\mathcal{D}$ defined by*

$$\tau_{\mathcal{D}}(x) = \bigsqcup x$$

is an isomorphism in $\mathbf{NLS}^{\mathrm{s}\nu}$.

17.5.26. THEOREM. *The categories $\mathbf{NLS}^{\mathrm{s}\nu}$ and $\mathbf{GTS}^{\mathsf{l}^\nu}$ are equivalent.*

PROOF. As in Corollary 17.2.15. using Propositions 17.5.24 and 17.5.25. ■

17.6. Exercises

17.6.1. Let $\mathcal{S}$ be an arbitrary free type structure. Show that

$$\begin{array}{lcll} \emptyset\cdot X & = & \emptyset & \text{for all } X\in\mathcal{F}^{\mathcal{T}} \\ X\cdot\emptyset & = & \emptyset & \text{for all } X\in\mathcal{F}^{\mathcal{T}}. \end{array}$$

17.6.2. Let $\mathcal{S}$ be a natural and β-sound type structure. Show that

$$\uparrow\textstyle\bigcap_{i\,\in\,I}(A_i\to B_i)\cdot\uparrow C \;=\; \textstyle\bigcap_{j\,\in\,I} B_j,$$

where $J = \{i\in I \mid C\le_{\mathcal{T}} A_i\}$.

17.6.3. Let $\mathcal{S}$ be a top type structure. Show that

$$F^{\mathcal{S}}(G^{\mathcal{S}}(\bot\mapsto\bot)) = (\bot\mapsto\bot) \;\Rightarrow$$
$$B =_{\mathcal{T}} \top \text{ whenever } A \to B =_{\mathcal{T}} \top \to \top.$$

17.6.4. Let $\mathcal{S}$ be a top type structure. Show that

$$F^{\mathcal{S}}(G^{\mathcal{S}}(\bot\mapsto\bot)) = (\bot\mapsto\bot) \;\Leftrightarrow$$
$$[\bigcap_{i\,\in\,I}(C_i{\to}D_i) \leq_{\mathcal{T}} A{\to}B \;\&\; \forall i\in I.D_i =_{\mathcal{T}} \top] \;\Rightarrow\; B =_{\mathcal{T}} \top.$$

17.6.5. Let $\mathcal{S}$ be an arbitrary type structure. Show that
$G^{\mathcal{S}}(\bigsqcup_{i\,\in\,I}(\uparrow A_i\mapsto\uparrow B_i)) \;\supseteq\; \uparrow\bigcap_{i\,\in\,I}(A_i{\to}B_i)$.

17.6.6. Let $\mathcal{S}$ be a proper type structure. Show that
$G^{\mathcal{S}}(\bigsqcup_{i\,\in\,I}(\uparrow A_i\mapsto\uparrow B_i)) \;=\; \uparrow\bigcap_{i\,\in\,I}(A_i{\to}B_i)$.

17.6.7. Let $\mathcal{S}$ be a natural type structure. Show that
$G^{\mathcal{S}}(\bigsqcup_{i\,\in\,I}(\uparrow A_i\mapsto\uparrow B_i)) \;=\; \uparrow\bigcap_{i\,\in\,I}(A_i{\to}B_i)$. **Comment:** immediate from previous exercise

17.6.8. Let $\mathcal{S}$ be a natural type structure. Show that $G^{\mathcal{S}}(\uparrow\top\mapsto\uparrow\top) \;=\; \uparrow\top$.

Chapter 18

Models 16.10.2006:1032

In this Chapter filter models, the main tool of Part III on the intersection types, will be introduced. A *filter* is a collection of types closed under intersection ($\cap$) and expansion ($\leq$). It turns out that there is a natural way to define application on such filters. This depends on the order $\leq$ on types and it will be shown for which of the type theories introduced in Chapter 15 the filters will turn out to be models of the untyped lambda calculus.

In Section 18.2 the filter models will be introduced as an applicative structures. Also it will be shown that the value of an untyped lambda term M in this structure is the collection of types that can be assigned to M. In Section 18.3 the approximation theorem will be shown, i.e. the interpretation of a lambda term is the supremum of those of its approximations.

18.1. Lambda models

Given a lambda structure $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$, i.e. a $\mathcal{D} \in \mathbf{ALG}$ with continuous $F : \mathcal{D} {\to} \mathcal{D} {\to} \mathcal{D}$ and $G : [\mathcal{D} {\to} \mathcal{D}] {\to} \mathcal{D}$, it is well known how one can interprete (untyped) lambda-terms in it. For lambda structures of the form $\mathcal{D} = \mathcal{F}^{\mathcal{T}}$ this interpretation turns out to have a simple form: the interpretation of a lambda term equals the set (actually a filter) of its possible types (in $\mathbb{T}^{\mathcal{T}}$). This will help us to determine for what $\mathcal{T}$ the corresponding filter structure is a lambda-model. This characterization can also be given for restricted lambda calculi, such as the λI-calculus.**Comment:** we do not consider other restricted calculi

18.1.1. Definition. (i) Let $\mathcal{D}$ be a set and Var the set of variables of the untyped lambda calculus. An *environment in* $\mathcal{D}$ is a total map

$$\rho : \mathrm{Var} {\to} \mathcal{D}.$$

The set of environments in $\mathcal{D}$ is denoted by $\mathsf{Env}_{\mathcal{D}}$.

(ii) If $\rho \in \mathsf{Env}_{\mathcal{D}}$ and $d \in \mathcal{D}$, then $\rho[x := d]$ is the $\rho' \in \mathsf{Env}_{\mathcal{D}}$ defined by

$$\begin{array}{rcll} \rho'(x) & = & d; & \\ \rho'(y) & = & \rho(y), & \text{if } y \neq x. \end{array}$$

The definition of a syntactic lambda-models was given in Barendregt [1984] (Definition 5.3.1 **Comment:** where you do not use coercion!) or Hindley and

Longo [1980]. We simply call these λ-models. We introduce also *applicative structures* (Definition 5.1.1 of Barendregt [1984]) and *quasi λ-models.*

Comment: better to avoid coercion? use $\mathcal{D}^{\mathsf{I}}$?

18.1.2. Definition. (i) An *applicative structure* is a pair $\langle\mathcal{D},\cdot\rangle$, where $\mathcal{D}$ is a set and $\cdot : \mathcal{D}\times\mathcal{D}\rightarrow\mathcal{D}$ is a binary operation on $\mathcal{D}$.

(ii) A *quasi λ-model* is of the form

$$\langle\mathcal{D},\cdot,[\![\ ]\!]\rangle,$$

where $\langle\mathcal{D},\cdot\rangle$ is an applicative structure and $[\![\ ]\!] : \Lambda\times\mathsf{Env}_{\mathcal{D}}\rightarrow\mathcal{D}$ satisfies the following.

$$\begin{array}{llcll}
(1) & [\![x]\!]^{\mathcal{D}}_{\rho} & = & \rho(x) & \\
(2) & [\![MN]\!]^{\mathcal{D}}_{\rho} & = & [\![M]\!]^{\mathcal{D}}_{\rho}\cdot[\![N]\!]^{\mathcal{D}}_{\rho} & \\
(3) & [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} & = & [\![\lambda y.M[x:=y]]\!]^{\mathcal{D}}_{\rho} & (\alpha), \\
 & & & \text{provided } y\notin \mathrm{FV}(M), & \\
(4) & \forall d\in\mathcal{D}.[\![M]\!]_{\rho[x:=d]}=[\![N]\!]_{\rho[x:=d]} & \Rightarrow & [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho}=[\![\lambda x.N]\!]^{\mathcal{D}}_{\rho} & (\xi) \\
(5) & \rho\restriction\mathrm{FV}(M)=\rho'\restriction\mathrm{FV}(N) & \Rightarrow & [\![M]\!]^{\mathcal{D}}_{\rho}=[\![M]\!]_{\rho'}. &
\end{array}$$

(iii) A *λ-model* is a quasi λ-model which safisfies:

$$(6)\quad [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho}\cdot d \;=\; [\![M]\!]_{\rho[x:=d]}\quad (\boldsymbol{\beta})$$

(iv) A *(quasi) λI-model* is defined similarly but replacing Λ by Λ^{I}, the set of λI-terms that require for each abstraction term $\lambda x.M$ that $x\in\mathrm{FV}(M)$. The corresponding clauses are denoted by (αI), ($\boldsymbol{\beta}$I) and (ξI).

We will write simply $[\![\ ]\!]_{\rho}$ instead of $[\![\ ]\!]^{\mathcal{D}}_{\rho}$ when there is no danger of confusion.

We have the following implications:
λ-model $\Longrightarrow$ λI-model $\Longrightarrow$ quasi λI-model;
λ-model $\Longrightarrow$ quasi λ-model $\Longrightarrow$ quasi λI-model.**Comment:** make a diagram?

18.1.3. Definition. Let $\mathcal{D}=\langle\mathcal{D},\cdot,[\![\ ]\!]\rangle$ be a (quasi) λ(I)-model.

(i) The statement $M=N$, for M,N untyped lambda terms, is *true in* $\mathcal{D}$, notation $\mathcal{D}\models M=N$ iff

$$\forall\rho\in\mathsf{Env}_D.[\![M]\!]_{\rho}=[\![N]\!]_{\rho}.$$

(ii) As usual one defines $\mathcal{D}\models\chi$, where χ is any statement built up using first order predicate logic from equations between untyped lambda terms.

(iii) A λ(I)-model $\mathcal{D}$ is called *extensional* iff

$$\mathcal{D}\models(\forall x.Mx=Nx)\;\Rightarrow\; M=N.$$

(iv) A λ(I)-model $\mathcal{D}$ is a λ(I)$\boldsymbol{\eta}$-model iff

$$\mathcal{D}\models\lambda x.Mx=M \text{ for } x\notin\mathrm{FV}(M)\qquad(\boldsymbol{\eta})$$

18.1.4. Definition. (i) Let $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ be a lambda structure, see Definition 17.4.1(i). Then $\mathcal{D}_{F,G}$ induces an interpretation $[\![\ ]\!] : \Lambda \times \mathsf{Env}_{\mathcal{D}} \to \mathcal{D}$ as follows.

$$\begin{array}{rcl} [\![x]\!]_\rho & = & \rho(x); \\ [\![MN]\!]_\rho & = & F([\![M]\!]_\rho)([\![N]\!]_\rho); \\ [\![\lambda x.M]\!]_\rho & = & G(\boldsymbol{\lambda} d \in \mathcal{D}.[\![M]\!]_{\rho[x:=d]}). \end{array}$$

Notice that the function $\boldsymbol{\lambda} d \in \mathcal{D}.[\![M]\!]_{\rho[x:=d]}$ used for $[\![\lambda x.M]\!]_\rho$ is continuous.

(ii) Now let $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ be a strict lambda structure, see Definition 17.4.1(ii). Then also $\mathcal{D}_{F,G}$ induces an interpretation $[\![\ ]\!] : \Lambda \times \mathsf{Env}_{\mathcal{D}} \to \mathcal{D}$ as above, changing the clause for $[\![\lambda x.M]\!]_\rho$ into

$$[\![\lambda x.M]\!]_\rho = G(\boldsymbol{\lambda} d \in \mathcal{D}.\ \text{if } d = \bot_{\mathcal{D}} \text{ then } \bot_{\mathcal{D}} \text{ else } [\![M]\!]_{\rho[x:=d]}).$$

18.1.5. Proposition. *For all (strict) lambda structures $\mathcal{D}_{F,G}$ induce quasi $\lambda(\mathsf{I})$-models.*

Proof. Let $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ be a lambda structure. Defining the interpretation $[\![\ ]\!] : \Lambda \times \mathsf{Env}_{\mathcal{D}} \to \mathcal{D}$ as in Definition 18.1.4 and the application $\cdot : \mathcal{D}\times\mathcal{D}\to\mathcal{D}$ as $d \cdot e = F(d)e$ it is easy to prove that one gets a quasi (strict) lambda model. ■

18.1.6. Proposition. *For all (strict) lambda structures $\mathcal{D}_{F,G}$ we have*

$$\begin{array}{lll} \text{(i)} & [\![\lambda x.M]\!]_\rho = [\![\lambda y.M[x := y]]\!]_\rho, \ \text{if } y \notin \mathrm{FV}(M). & (\alpha) \\ \text{(ii)} & (\forall d \in \mathcal{D}. [\![M]\!]_{\rho[x:=d]} = [\![N]\!]_{\rho[x:=d]}) \Rightarrow [\![\lambda x.M]\!]_\rho = [\![\lambda x.N]\!]_\rho. & (\xi) \\ \text{(iii)} & \rho \restriction \mathrm{FV}(M) = \rho' \restriction \mathrm{FV}(M) \ \Rightarrow\ [\![M]\!]_\rho = [\![M]\!]_{\rho'}. & \end{array}$$

Proof. Easy. ■

Therefore the only requirement that a (strict) lambda structure misses to be a $\lambda(\mathsf{I})$-model is the axiom $(\boldsymbol{\beta}(\mathsf{I}))$.

18.1.7. Proposition. (i) *Let $\langle \mathcal{D}, F, G\rangle$ with $\mathcal{D} \in \mathbf{ALG}$ be a lambda structure. Then the following statements are equivalent.*

(1) *$\mathcal{D} \models (\lambda x.M)N = M[x: = N]$, for all $M, N \in \Lambda$;*

(2) *$[\![\lambda x.M]\!]_\rho.d = [\![M]\!]_{\rho(x:=d)}$, for all $M \in \Lambda$ and $d \in \mathcal{D}$;*

(3) *$\mathcal{D}$ is a λ-model;*

(4) *$\mathcal{D} \models \lambda\boldsymbol{\beta}$, where $\lambda\boldsymbol{\beta} = \{M = N \mid \lambda\boldsymbol{\beta} \vdash M = N\}$.*

(ii) *As* (i)*, but $\mathcal{D}$ is a strict lambda structure, in* (1) *and* (2) *the extra condition '$x \in \mathrm{FV}(M)$', in* (1) *$\mathcal{D} \models (\lambda x.M)N = M[x: = N]$, in* (3) *'$\mathcal{D}$ is a $\lambda\mathsf{I}$-model' and in* (4) *'$\mathcal{D} \models \lambda\boldsymbol{\beta}\mathsf{I}$, where $\lambda\boldsymbol{\beta}\mathsf{I} = \{M = N \mid \lambda\boldsymbol{\beta}\mathsf{I} \vdash M = N\}$.*

Proof. (i) (1)$\Rightarrow$(2). By (1) one has $[\![(\lambda x.M)N]\!]_\rho = [\![M[x := N]]\!]_\rho$. Taking $N \equiv x$ and $\rho' = \rho(x := d)$ one obtains

$$[\![(\lambda x.M)x]\!]_{\rho'} = [\![M]\!]_{\rho'},$$

hence

$$[\![\lambda x.M]\!]_\rho \cdot d = [\![M]\!]_{\rho'},$$

as $\rho \restriction \mathrm{FV}(\lambda x.M) = \rho' \restriction \mathrm{FV}(\lambda x.M)$.

(2)$\Rightarrow$(3). By (ii), Definition 18.1.4 and Proposition 18.1.5 all conditions to be a λ-model, see Definition 18.1.2, are fulfilled.

(3)$\Rightarrow$(4). By Theorem 5.3.4 in Barendregt [1984].

(4)$\Rightarrow$(1). Trivial.

(ii) Similarly. ∎

18.1.8. Corollary. *Let $\mathcal{D}_{F,G} = \langle \mathcal{D}, F, G\rangle$ be a (strict) lambda structure and a $\lambda(\mathsf{I})$-model. Then*

$$\mathcal{D} \textit{ is a } \lambda(\mathsf{I})\boldsymbol{\eta}\textit{-model} \Leftrightarrow \mathcal{D} \textit{ is an extensional } \lambda(\mathsf{I})\textit{-model.}$$

Proof. ($\Rightarrow$) Suppose that for some ρ one has for all $d \in \mathcal{D}$

$$[\![Mx]\!]_{\rho[x:=d]} = [\![Nx]\!]_{\rho[x:=d]}.$$

Then by ($\boldsymbol{\eta}$) and Proposition 18.1.5(ii) one has

$$[\![M]\!]_\rho = [\![\lambda x.Mx]\!]_\rho = [\![\lambda x.Nx]\!]_\rho = [\![N]\!]_\rho.$$

($\Leftarrow$) Note that by ($\boldsymbol{\beta}(\mathsf{I})$) one has $\mathcal{D} \models (\lambda x.Mx)y = My$, where x is fresh. Hence by extensionality one has $\mathcal{D} \models \lambda x.Mx = M$.

Similarly. ∎

18.1.1. Isomorphisms of λ-models

18.1.9. Definition. (i) A *reflexive structure* $\langle D, F, G\rangle$ is a lambda structure such that $F \circ G = Id$.

(ii) An *extensional* reflexive structure is a reflexive lambda structure such that $G \circ F = Id$.

18.1.10. Proposition. (i) *Each reflexive structure is a λ-model.*

(ii) *Each extensional reflexive structure is an extensional λ-model.*

Proof. This is Theorem 5.4.4 of Barendregt [1984].

18.1.11. Definition. (i) An isomorphism between two reflexive structures $\langle \mathcal{D}, F, G\rangle$ and $\langle \mathcal{D}', F', G'\rangle$ is a bijective mapping m such that

(1) $\mathsf{m}(G(f)) = G'(\mathsf{m} \circ f \circ \mathsf{m}^{-1})$

(2) $\mathsf{m}(F(d)(e)) = F'(\mathsf{m}(d))(\mathsf{m}(e))$

18.1.12. PROPOSITION. (i) *If $\mathcal{D}$ and $\mathcal{D}'$ are isomorphic lambda models via* m *then for all λ-terms M and environments ρ:*

$$\mathsf{m}([\![M]\!]^{\mathcal{D}}_{\rho}) = [\![M]\!]^{\mathcal{D}'}_{\mathsf{m}\circ\rho}$$

(ii) *If two lambda models are isomorphic then they equal the same terms, i.e. $\mathcal{D} \models M = N$ iff $\mathcal{D}' \models M = N$.*

PROOF. (i) is proved by induction on M.
(ii) follows from (i).

18.2. Filter models

Now we introduce the fundamental notion of filter structure, which will be used extensively in this Section. It is of paramount importance, and one can say that all the preceding sections in this Chapter are a build-up to it. Since the seminal paper Barendregt et al. [1983], this notion has played a major role in the study of the mathematical semantics of lambda-calculus. By Definition ?? we write for $\mathcal{T} \in \mathbf{TS}$ and X a non-empty subset of $\mathcal{T}$

$$\uparrow X = \{x \in \mathcal{T} \mid \exists n \geq 1 \exists x_1, \ldots, x_n . x_1 \cap \ldots \cap x_n \leq x\}.$$

Now we extend this notion as follows.

Comment: we need filter models also for TT, not only for TS

By Definition 15.4.2(ii) we have for $\mathcal{T}$ a TT and X a non-empty subset of $\mathcal{T}$

$$\uparrow X = \{x \in \mathcal{T} \mid \exists n \geq 1\, \exists x_1 \ldots x_n \in X . x_1 \cap \ldots \cap x_n \leq x\}.$$

Now we extend this notion as follows.

18.2.1. DEFINITION. (i) We define $\uparrow\emptyset = \{\top\}$.
(ii) Let $\mathcal{T}$ be a TT. Then we define $\uparrow^s X \in \mathcal{F}^{\mathcal{S}}_s$ by

$$\begin{aligned} \uparrow^s X &= \uparrow X, \quad \text{if } X \neq \emptyset; \\ \uparrow^s \emptyset &= \emptyset. \end{aligned}$$

Comment: moved at page 36

18.2.2. DEFINITION. (i) Let $\mathcal{T}$ be a $\mathrm{TT}^{\top}$. We define

$$\begin{aligned} F^{\mathcal{T}} &: [\mathcal{F}^{\mathcal{T}} \rightarrow [\mathcal{F}^{\mathcal{T}} \rightarrow \mathcal{F}^{\mathcal{T}}]] \text{ and} \\ G^{\mathcal{T}} &: [[\mathcal{F}^{\mathcal{T}} \rightarrow \mathcal{F}^{\mathcal{T}}] \rightarrow \mathcal{F}^{\mathcal{T}}] \end{aligned}$$

by

$$\begin{aligned} F^{\mathcal{T}}(X)(Y) &= \uparrow\{B \in \mathbb{T}^{\mathcal{T}} \mid \exists A \in Y . (A \rightarrow B) \in X\}; \\ G^{\mathcal{T}}(f) &= \uparrow\{A \rightarrow B \mid B \in f(\uparrow A)\}. \end{aligned}$$

(ii) Let $\mathcal{T}$ be a TT. We define

$$\begin{array}{lcl} F^{\mathcal{T}}_s & : & [\mathcal{F}^{\mathcal{T}}_s \to_{\mathrm{s}} [\mathcal{F}^{\mathcal{T}}_s \to_{\mathrm{s}} \mathcal{F}^{\mathcal{T}}_s]] \text{ and} \\ G^{\mathcal{T}}_s & : & [[\mathcal{F}^{\mathcal{T}}_s \to_{\mathrm{s}} \mathcal{F}^{\mathcal{T}}_s] \to_{\mathrm{s}} \mathcal{F}^{\mathcal{T}}_s] \end{array}$$

by

$$\begin{array}{rcl} F^{\mathcal{T}}_s(X)(Y) & = & \uparrow^s \{B \in \mathbb{T}^{\mathcal{T}} \mid \exists A \in Y.(A \to B) \in X\}; \\ G^{\mathcal{T}}_s(f) & = & \uparrow^s \{A \to B \mid B \in f(\uparrow A)\}. \end{array}$$

18.2.3. Lemma. (i) *The structure* $\mathcal{F}^{\mathcal{T}} = \langle \mathcal{F}^{\mathcal{T}}, F^{\mathcal{T}}, G^{\mathcal{T}} \rangle$ *is a lambda structure: it is called the* filter structure *over* $\mathcal{T}$.

(ii) *The structure* $\mathcal{F}^{\mathcal{T}}_s = \langle \mathcal{F}^{\mathcal{T}}_s, F^{\mathcal{T}}_s, G^{\mathcal{T}}_s \rangle$ *is a strict lambda structure: it is called the* strict filter structure *over* $\mathcal{T}$.

Proof. It is easy to verify that $F^{\mathcal{T}}, G^{\mathcal{T}}, F^{\mathcal{T}}_s, G^{\mathcal{T}}_s$ are continuous. ■

Recall that by Proposition 15.3.3 a compatible $\mathrm{TT}^{(\top)}$ induces a $\mathrm{TS}^{(\top)}$. We can take advantage in this case of the equivalencies between type and zip structures (Theorems 17.3.8 and 17.3.28).

18.2.4. Lemma. (i) *If* $\mathcal{S} \in \mathbf{TS}^{\top}$, *then* $F^{\mathcal{S}} = F_{Z^{\mathcal{S}}}$ *and* $G^{\mathcal{S}} = G_{Z^{\mathcal{S}}}$, *where* $F_{Z^{\mathcal{S}}}$ *and* $G^{\mathcal{S}} = G_{Z^{\mathcal{S}}}$ *are defined in Definitions 17.3.1 and 17.4.10.*

(ii) *If* $\mathcal{S} \in \mathbf{TS}$, *then* $F^{\mathcal{S}}_s = F_{Z^{\mathcal{S}}_s}$ *and* $G^{\mathcal{S}}_s = G_{Z^{\mathcal{S}}_s}$, *where* $F_{Z^{\mathcal{S}}_s}$ *and* $G_{Z^{\mathcal{S}}_s}$ *are defined in Definitions 17.3.17 and 17.4.33*

Proof. (i) Taking the suprema in $\mathcal{F}^{\mathcal{S}}$ one has

$$\begin{array}{rcl} F^{\mathcal{S}}(X)(Y) & = & \uparrow\{\uparrow A \mid \exists B \in Y.(B \to A) \in X\} \\ & = & \bigsqcup\{\uparrow A \mid \exists B \in Y.\uparrow(B \to A) \subseteq X\} \\ & = & \bigsqcup\{\uparrow A \mid \exists \uparrow B \subseteq Y.Z^{\mathcal{S}}(\uparrow B, \uparrow A) \subseteq X\} \\ & = & X \cdot_{Z^{\mathcal{S}}} Y. \end{array}$$

Moreover,

$$\begin{array}{rcl} G^{\mathcal{S}}(f) & = & \uparrow\{B \to A \mid A \in f(\uparrow B)\} \\ & = & \bigsqcup\{\uparrow(B \to A) \mid A \in f(\uparrow B)\} \\ & = & \bigsqcup\{Z(\uparrow B, \uparrow A) \mid \uparrow A \subseteq f(\uparrow B)\}. \end{array}$$

(ii) Now the suprema are taken in $\mathcal{F}^{\mathcal{S}}_s$ and $\bigsqcup\emptyset = \emptyset$, the bottom of $\mathcal{F}^{\mathcal{S}}_s$. ■

The rest of this section is devoted to the characterization of those type theories $\mathcal{T}$ such that $\mathcal{F}^{\mathcal{T}}$ is a $\lambda(\mathsf{I})$-model, a so-called filter λ-model. The following *type-semantics theorem* is important. It has as consequence that for a closed untyped lambda term M and a $\mathrm{TT}^{\top}$ $\mathcal{T}$ one has

$$[\![M]\!]^{\mathcal{F}^{\mathcal{T}}} = \{A \mid \vdash^{\mathcal{T}}_{\cap^{\top}} M : A\},$$

i.e. the semantical meaning of M in the filter λ-model corresponding to a a $\mathrm{TT}^{\top}$ $\mathcal{T}$ is the collection of its types. For a TT $\mathcal{T}$ one has

$$[\![M]\!]^{\mathcal{F}_s^{\mathcal{T}}} = \{A \mid \vdash_{\cap}^{\mathcal{T}} M : A\}.$$

In order to deal with arbitrary (hence open) terms, we need to extend the notion of context.

18.2.5. Definition. (i) A *flexible context* Γ is a set, now possibly infinite, of statements of the form $x : A$ where the requirement

$$(x{:}A), (x{:}B) \in \Gamma \;\Rightarrow\; A = B$$

is dropped.

(ii) If ρ is a valuation in a filter structure $\mathcal{F}^{\mathcal{T}}_{(s)}$, then we define the flexible context $\Gamma_\rho = \{(x{:}B) \mid B \in \rho(x)\}$.

(iii) For a flexible context Γ we define

$$\Gamma \vdash_{\cap^{\top}(\top)} M : A \;\Leftrightarrow\; \exists \Gamma_0 \subseteq \Gamma.[\Gamma_0 \text{ finite } \;\&\; \cap\Gamma_0 \vdash_{\cap^{\top}(\top)} M : A],$$

where $\cap\Gamma_0 = \{x{:}(B_1 \cap \ldots \cap B_k) \mid (x{:}B_i) \in \Gamma_0\}$.

18.2.6. Definition. A context Γ *agrees with an environment* $\rho \in \mathsf{Env}^{(s)}_{\mathcal{F}^{\mathcal{T}}_{(s)}}$ (notation $\Gamma \models \rho$) if $x : A \in \Gamma$ implies $A \in \rho(x)$.

We immediately get:

18.2.7. Proposition. (i) *If $\Gamma \models \rho$ and $\Gamma' \models \rho$, then $\Gamma \uplus \Gamma' \models \rho$.*

(ii) *If $\Gamma \models \rho[x := \uparrow^{(s)} A]$, then $\Gamma \backslash x \models \rho$.*

18.2.8. Theorem (Type-semantics Theorem). (i) *Let $\mathcal{T}$ be a $\mathrm{TT}^{\top}$ and $\mathcal{F}^{\mathcal{T}}$ its corresponding filter structure. Then, for any lambda-term M and $\rho \in \mathsf{Env}_{\mathcal{F}^{\mathcal{T}}}$,*

$$[\![M]\!]_\rho = \{A \mid \Gamma \vdash^{\mathcal{T}}_{\cap^{\top}} M : A \textit{ for some } \Gamma \models \rho\}.$$

(ii) *Let $\mathcal{T}$ be a TT and $\mathcal{F}^{\mathcal{T}}_s$ its corresponding strict filter structure. Then, for any lambda-term M and $\rho \in \mathsf{Env}^{s}_{\mathcal{F}^{\mathcal{T}}_s}$,*

$$[\![M]\!]_\rho = \{A \mid \Gamma \vdash^{\mathcal{T}}_{\cap^{\top}} M : A \textit{ for some } \Gamma \models \rho\}.$$

Proof. (i) By induction on the structure of M.
If $M \equiv x$, then

$$\begin{array}{rcll} [\![x]\!]_\rho & = & \rho(x) & \\ & = & \{A \mid A \in \rho(x)\} & \\ & = & \{A \mid A \in \rho(x) \;\&\; x : A \vdash^{\mathcal{T}}_{\cap^{\top}} x : A\} & \\ & = & \{A \mid \Gamma \vdash^{\mathcal{T}}_{\cap^{\top}} x : A \text{ for some } \Gamma \models \rho\}, & \text{by Definition 18.2.6.} \end{array}$$

If $M \equiv NL$, then

$$\begin{aligned}
[\![NL]\!]_\rho &= [\![N]\!]_\rho \cdot [\![L]\!]_\rho \\
&= \uparrow\{A \mid \exists B \in [\![L]\!]_\rho.(B \to A) \in [\![N]\!]_\rho\} \\
&= \{A \mid \exists k{>}0 \exists B_1,\ldots,B_k, C_1,\ldots,C_k. \\
&\quad [\,(B_i{\to}C_i) \in [\![N]\!]_\rho \;\&\; B_i \in [\![L]\!]_\rho \;\&\; (\textstyle\bigcap_{1\leq i\leq k} C_i) \leq A]\} \cup \uparrow\{\top\}, \\
&\quad \text{by definition of } \uparrow, \\
&= \{A \mid \exists k{>}0 \exists B_1,\ldots,B_k, C_1,\ldots,C_k. \text{ for some } \Gamma_{1i}, \Gamma_{2i} \models \rho \\
&\quad [\Gamma_{1i} \vdash^{\mathcal{T}}_{\cap\top} N : (B_i{\to}C_i) \;\&\; \Gamma_{2i} \vdash^{\mathcal{T}}_{\cap\top} L : B_i \;\&\; C_1 \cap \ldots \cap C_k \leq A]\} \cup \uparrow\{\top\}, \\
&\quad \text{by induction hypothesis,} \\
&= \{A \mid \Gamma \vdash^{\mathcal{T}}_{\cap\top} NL : A \text{ for some } \Gamma \models \rho\},\ \text{taking } \Gamma = \Gamma_{11} \uplus \ldots \uplus \Gamma 1k \uplus \ldots \uplus \Gamma_{21} \uplus \ldots \uplus \Gamma_{2k}, \\
&\quad \text{by Theorem 16.1.1(ii) and Proposition 18.2.7(i).}
\end{aligned}$$

If $M \equiv \lambda x.N$, then

$$\begin{aligned}
[\![\lambda x.N]\!]_\rho &= G^{\mathcal{T}}(\boldsymbol{\lambda} X \in \mathcal{F}^{\mathcal{T}}.[\![N]\!]_{\rho[x:=X]}) \\
&= \uparrow\{(B{\to}C) \mid C \in [\![N]\!]_{\rho[x:=\uparrow B]}\} \\
&= \{A \mid \exists k{>}0 \exists B_1,\ldots,B_k, C_1,\ldots,C_k. \text{ for some } \Gamma_i \models \rho[x:=\uparrow B_i] \\
&\quad [\Gamma_i, x{:}B_i \vdash^{\mathcal{T}}_{\cap\top} N : C_i \;\&\; (B_1{\to}C_1) \cap \ldots \cap (B_k{\to}C_k) \leq A]\},\ \text{by the induction hypothesis} \\
&= \{A \mid \Gamma \vdash^{\mathcal{T}}_{\cap\top} \lambda x.N : A \text{ for some } \Gamma \models \rho\},\ \text{taking } \Gamma = (\Gamma_1 \uplus \ldots \uplus \Gamma_k)\backslash x \\
&\quad \text{by Theorem 16.1.1(iii), rule } (\leq) \text{ and Proposition 18.2.7(ii).}
\end{aligned}$$

(ii) Similarly, with $\uparrow$ replaced by $\uparrow^s$. Note that in the case $M = NL$ we drop '$\cup\uparrow\{\top\}$' both times and in the case $M = \lambda x.N$, using Definition 18.1.4, $[\![\lambda x.N]\!]^{\mathcal{T}}_\rho = \uparrow^s \{(B{\to}C) \mid C \in [\![N]\!]^{\mathcal{T}}\}_{\rho[x:=\uparrow B]}$ holds because $\uparrow B \neq \emptyset$. ■

18.2.9. Corollary. (i) *Let $\mathcal{T}$ be a* TT$^\top$. *Then*

$$\mathcal{F}^{\mathcal{T}} \text{ is a } \lambda\text{-model} \Leftrightarrow [\Gamma \vdash^{\mathcal{T}}_{\cap\top} (\lambda x.M) : (B{\to}A) \;\Rightarrow\; \Gamma, x{:}B \vdash^{\mathcal{T}}_{\cap\top} M : A].$$

(ii) *Let $\mathcal{T}$ be a* TT. *Then*

$$\mathcal{F}^{\mathcal{T}}_s \text{ is a } \lambda\mathsf{I}\text{-model} \Leftrightarrow$$
$$[\Gamma \vdash^{\mathcal{T}}_{\cap\top} (\lambda x.M) : (B{\to}A) \;\&\; x \in \mathrm{FV}(M) \;\Rightarrow\; \Gamma, x{:}B \vdash^{\mathcal{T}}_{\cap\top} M : A].$$

Proof. (i) By Propositions 18.1.7(i), 16.2.1(ii) and Corollary 16.2.5(i).
(ii) By Propositions 18.1.7(ii), 16.2.1(i) and Corollary 16.2.5(ii). ■

18.2.10. Corollary. *Let $\mathcal{T}$ be an* TS$^\top$.

(i) *Suppose*

$$\Gamma \vdash_{\Omega}^{\mathcal{T}} \lambda x.M : A \to B \;\Rightarrow\; \Gamma, x{:}A \vdash_{\Omega}^{\mathcal{T}} M : B.$$

Then $\mathcal{F}^{\mathcal{T}}$ is a λ-model.

(ii) *Suppose*

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

Then $\mathcal{F}_s^{\mathcal{T}}$ is a λI-model.

PROOF. By the Corollary above and the Theorem 16.1.10(iii). ■

18.2.11. COROLLARY. (i) *Let $\mathcal{T}$ be a* $\mathrm{TT}^{\top}$. *Then*

$$\mathcal{T} \textit{ is } \boldsymbol{\beta}\textit{-sound} \;\Rightarrow\; \mathcal{F}^{\mathcal{T}} \textit{ is a } \lambda\textit{-model.}$$

(ii) *Let $\mathcal{T}$ be a* TT. *Then*

$$\mathcal{T} \textit{ is } \boldsymbol{\beta}\textit{-sound} \;\Rightarrow\; \mathcal{F}_s^{\mathcal{T}} \textit{ is a } \lambda\mathsf{I}\textit{-model.}$$

PROOF. By the Corollary above and Theorem 16.1.10(iii). ■

18.2.12. COROLLARY. (i) *Let* $\mathcal{T} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{HR}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Plotkin}, \mathrm{Engeler}, \mathrm{CDS}\}$. *Then*

$$\mathcal{F}^{\mathcal{T}} \textit{ is a } \lambda\textit{-model.}$$

(ii) *Let* $\mathcal{T} \in \{\mathrm{HL}, \mathrm{CDV}, \mathrm{CD}\}$. *Then*

$$\mathcal{F}_s^{\mathcal{T}} \textit{ is a } \lambda\mathsf{I}\textit{-model.}$$

PROOF. (i) By (i) of the previous Corollary and Theorem 16.1.8.
(ii) By (ii) of the Corollary, using Theorem 16.1.8. ■

18.2.13. PROPOSITION. (i) *Let $\mathcal{T}$ be a* $\mathrm{TT}^{\top}$. *Then*

$$\mathcal{T} \textit{ is natural and } \boldsymbol{\beta}\textit{- and } \boldsymbol{\eta}^{\top}\textit{-sound} \;\Rightarrow\; \mathcal{F}^{\mathcal{T}} \textit{ is an extensional } \lambda\textit{-model.}$$

(ii) *Let $\mathcal{T}$ be a TT. Then*

$$\mathcal{T} \textit{ is proper and } \boldsymbol{\beta}\textit{- and } \boldsymbol{\eta}\textit{-sound} \;\Rightarrow\; \mathcal{F}_s^{\mathcal{T}} \textit{ is an extensional } \lambda\mathsf{I}\textit{-model.}$$

PROOF. (i) and (ii) $\mathcal{F}^{\mathcal{T}}$ ($\mathcal{F}_s^{\mathcal{T}}$) is a λ(I)-model by Corollary 18.2.11(i)((ii)). For extensionality by Corollary 18.1.10 one needs to verify for $x \notin \mathrm{FV}(M)$

$$[\![\lambda x.Mx]\!]_\rho = [\![M]\!]_\rho. \qquad (\boldsymbol{\eta})$$

This follows from Theorems 18.2.8(i), and 16.2.15(i).
(ii) We know that extensionality is equivalent to (eta). Barendregt [1984], Theorem ??. This (eta) follows from Theorems 18.2.8(ii) and 16.2.15(ii). ■

18.2.14. Corollary. (i) *Let* $\mathcal{T} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{HR}, \mathrm{DHM}\}$. *Then*

$$\mathcal{F}^{\mathcal{T}} \text{ is an extensional } \lambda\text{-model.}$$

(ii) *Let* $\mathcal{T} = \mathrm{HL}$. *Then*

$$\mathcal{F}^{\mathcal{T}}_s \text{ is an extensional } \lambda\mathsf{I}\text{-model.}$$

Proof. (i) and (ii) By Corollary 16.2.13.
(ii) By Corollary 16.2.13. ■

As shown in Meyer [1982], see also Barendregt [1984] Ch.4, an lambda structure is a λ-model provided that it contains the combinators K, S and ε, satisfying certain properties. Thus, a condition for being a filter λ-model can be obtained by simply forcing the existence of such combinators. This yields a characterization of the natural type theories which induce filter λ-models. See Alessi [1991] for the rather technical proof.

18.2.15. Theorem. *Let* $\mathcal{T}$ *be a* NTT.

(i) *The filter structure* $\mathcal{F}^{\mathcal{T}}$ *is a filter* λ*-model if and only if the following three conditions are fulfilled in* $\mathcal{T}$.

(K) *For all* C, E *one has*

$$C \leq E \;\Leftrightarrow\; \forall D\, \exists\{A_i, B_i\}_{i \in I}. \bigcap_{i \in I}(A_i \to B_i \to A_i) \leq C \to D \to E.$$

(S) *For all* D, E, F, G *one has*

$$\exists H.[E \leq F \to H \;\&\; D \leq F \to H \to G] \;\Leftrightarrow$$

$$\left[\begin{array}{l} \exists\{A_i, B_i, C_i\}_{i \in I}. \\ \bigcap_{i \in I}(A_i \to B_i \to C_i) \to (A_i \to B_i) \to (A_i \to C_i) \leq D \to E \to F \to G \end{array}\right].$$

(ε) *For all* C, D *one has*

$$\exists\{A_i, B_i\}_{i \in I}.(\bigcap_{i \in I}(A_i \to B_i) \to (A_i \to B_i)) \leq (C \to D) \;\Leftrightarrow$$

$$\exists\{E_j, F_j\}_{j \in J}.C \leq (\bigcap_{j \in J} E_j \to F_j) \leq D.$$

(ii) *The structure* $\mathcal{F}^{\mathcal{T}}$ *is an extensional filter* λ*-model iff the third condition above is replaced by the following two.*

(ϵ_1) $\forall A\, \exists\{A_i, B_i\}_{i \in I}.A =_{\mathcal{T}} \bigcap_{i \in I}(A_i \to B_i)$;

(ϵ_2) $\forall A, B\, \exists\{A_i\}_{i \in I}.\bigcap_{i \in I}(A_i \to A_i) \leq (A \to B) \Leftrightarrow A \leq B$. ■

The call-by-value λ-calculus.

18.2.16. THEOREM. *Let $\mathcal{T} \in \mathbf{GTS}^{\mathsf{I}^\nu}$ and $\mathcal{F}^{\mathcal{T}}_{\nu}$ be its corresponding ν-filter structure. Then, for any lambda term M, $\rho \in \mathsf{Env}_{\mathcal{F}^{\mathcal{T}}_{\nu}}$ and Γ_ρ as above,*

$$[\![M]\!]^{\mathcal{F}^{\mathcal{T}}_{\nu}}_{\rho} = \{A \in \mathcal{T} \mid \Gamma_\rho \vdash^{\mathcal{T}} M : A\}.$$

PROOF. Similar to the proof of Theorem 18.2.8. Instead of Theorem 16.1.1 we now use Theorem 17.5.3. ∎

The model induced by $\lambda_{\cap\nu}$ gives a *fully abstract* model for the *call-by-value* operational semantics (for a proof see Egidi et al. [1992]). Moreover one has the following.

18.2.17. THEOREM (Egidi et al. [1992]). *Let D^{strict}_{∞} be an inverse limit solution of the domain equation $D \cong [D \to_{\perp} D]_{\perp}$. Then $D^{strict}_{\infty} \cong \mathcal{F}^{\mathrm{EHR}}$.* ∎

Models of the call-by-value λ-calculus are the topic of the book Ronchi Della Rocca and Paolini [2004], which uses many intersection type systems. EXPAND! (After 12.2.2006) **Comment:** I will make a chapter on further reading as soon as the other parts of the book are finished

Representability of continuous functions

In this subsection following Alessi, Barbanera and Dezani-Ciancaglini [2004] we will isolate a number of conditions on a NTT $\mathcal{T}$ implying that the set of *representable functions* $\mathcal{F}^{\mathcal{T}} \to \mathcal{F}^{\mathcal{T}}$, i.e. the set of functions in the image of $F^{\mathcal{T}}$, contains several interesting classes of functions.

18.2.18. DEFINITION. A function $f : \mathcal{D} \to \mathcal{D}$ is representable in the lambda structure $\langle \mathcal{D}, F, G\rangle$ if $f = F(d)$ for some $d \in \mathcal{D}$.

18.2.19. LEMMA. *Let $\mathcal{T}$ be a NTT and let $f \in [\mathcal{F}^{\mathcal{T}} \to \mathcal{F}^{\mathcal{T}}]$. Then*

$$f \text{ is representable} \;\Leftrightarrow\; F^{\mathcal{T}} \circ G^{\mathcal{T}}(f) = f.$$

PROOF. ($\Leftarrow$) Trivial. ($\Rightarrow$) Suppose $F^{\mathcal{T}}(X) = f$. Claim $F^{\mathcal{T}}(G^{\mathcal{T}}(F^{\mathcal{T}}(X))) = F^{\mathcal{T}}(X)$. One has $G^{\mathcal{T}}(F^{\mathcal{T}}(X)) = \uparrow\{a \to b \mid a \to b \in X\}$. Hence $a \to b \in G^{\mathcal{T}}(F^{\mathcal{T}}(X)) \Leftrightarrow a \to b \in X$. So for each Y we have indeed $F^{\mathcal{T}}(G^{\mathcal{T}}(F^{\mathcal{T}}(X)))(Y) = F^{\mathcal{T}}(X)(Y)$. ∎

Comment: moved at page 88

18.2.20. LEMMA. *Let $\mathcal{T}$ be a NTT and define the function $h : \mathcal{F}^{\mathcal{T}} \to \mathcal{F}^{\mathcal{T}}$ by*

$$h = (\uparrow A_1 \Rightarrow \uparrow B_1) \sqcup \ldots \sqcup (\uparrow A_n \Rightarrow \uparrow B_n).$$

Then for $C \in \mathbb{T}^{\mathcal{T}}$

(i) $h(\uparrow C) = \{D \mid B_{i1} \cap \ldots \cap B_{ik} \leq D \;\&\; C \leq A_{i1} \cap \ldots \cap A_{ik}\}$.

(ii) $(F^{\mathcal{T}} \circ G^{\mathcal{T}})(h)(\uparrow c) = \{d \mid (a_1 \to b_1) \cap \ldots \cap (a_n \to b_n) \leq (c \to d)\}$.

PROOF. (i)
$$\begin{aligned} h(\uparrow C) &= \bigsqcup\{\uparrow B_i \mid \uparrow A_i \leq \uparrow C\} \\ &= \uparrow \bigcap\{B_i \mid C \leq A_i\}, && \text{by Proposition 15.4.4(ii} \\ &= \{D \mid B_{i1} \cap \ldots \cap B_{ik} \leq D \mathbin{\&} C \leq A_{i1} \cap \ldots \cap A_{ik}\}. \end{aligned}$$

(ii) $(F^{\mathcal{T}} \circ G^{\mathcal{T}})(h)(\uparrow C) =$ [Correct!]

$$\begin{aligned} &= F^{\mathcal{T}}(G^{\mathcal{T}}((\uparrow A_1 \Rightarrow \uparrow B_1)) \sqcup \ldots \sqcup G^{\mathcal{T}}((\uparrow A_n \Rightarrow \uparrow B_n))))(\uparrow C), && \text{by Lemma 17.4.32(ii),} \\ &= F^{\mathcal{T}}((\uparrow A_1 \to \uparrow B_1) \sqcup \ldots \sqcup (\uparrow A_n \to \uparrow B_n)))(\uparrow c), && \text{by Definition 18.2.2(i),} \\ &= F^{\mathcal{T}}(((\uparrow A_1 \to \uparrow B_1) \cap \ldots \cap (\uparrow A_n \to \uparrow B_n)))(\uparrow C), && \text{by Proposition 15.4.4(iii),} \\ &= \{D \mid \exists E \in \uparrow C.(E \to D) \in \uparrow((A_1 \to B_1) \cap \ldots \cap (A_n \to B_n))\}, && \text{see Definition 18.2.2(ii),} \\ &= \{D \mid \exists E \geq C.(A_1 \to B_1) \cap \ldots \cap (A_n \to B_n) \leq (E \to D)\} \\ &= \{D \mid (A_1 \to B_1) \cap \ldots \cap (A_n \to B_n) \leq (C \to D)\}, && \text{by } (\to). \end{aligned}$$

18.2.21. THEOREM. *Let $\mathcal{T}$ be a NTT. Let $\mathcal{R}$ be the set of representable functions $\mathcal{F}^{\mathcal{T}} \to \mathcal{F}^{\mathcal{T}}$. Then we have the following.*

(i) *$\mathcal{R}$ contains the bottom function $\mathsf{K}\bot = \bot_{\mathcal{F}^{\mathcal{T}}} \mapsto \bot_{\mathcal{F}^{\mathcal{T}}}$ iff for all C, D*

$$\top \leq C \to D \;\Rightarrow\; \top \leq D.$$

(ii) *$\mathcal{R}$ contains the constant functions iff for all B, C, D*

$$\top \to B \leq C \to D \;\Rightarrow\; B \leq D.$$

(iii) *$\mathcal{R}$ contains the continuous step functions iff for all A, B, C, D*

$$A \to B \leq C \to D \mathbin{\&} D \neq \top \;\Rightarrow\; C \leq A \mathbin{\&} B \leq D.$$

(iv) *$\mathcal{R}$ contains the continuous functions iff*

$$\begin{aligned} &\forall n \geq 0, \forall A_1, \ldots, A_n, B_1, \ldots, B_n, C, D \in \mathcal{T} \\ &(A_1 \to B_1) \cap \ldots (A_n \to B_n) \leq (C \to D) \;\Rightarrow\; [C \leq A_{i_1} \cap \ldots \cap A_{i_k} \mathbin{\&} B_{i_1} \cap \ldots \cap B_{i_k} \leq D] \\ &\qquad\qquad\qquad\qquad\qquad\qquad\quad \textit{for some } k \geq 0,\ 1 \leq i_1 < \ldots < i_k \leq n. \end{aligned}$$

PROOF. (i) Assume that $\mathsf{K}\bot \in \mathcal{R}$. Then $F^{\mathcal{T}}(G^{\mathcal{T}}(\mathsf{K}\bot)) = \mathsf{K}\bot$, by Lemma 18.2.19. Observe that

$$\begin{aligned} G^{\mathcal{T}}(\mathsf{K}\bot) &= \uparrow\{A \to \top\}, && \text{by Definition 18.2.2(i),} \\ &= \uparrow\top && \text{since } \mathcal{T} \text{ is natural.} \end{aligned}$$

Hence $F^{\mathcal{T}}(\bot_{\mathcal{F}^{\mathcal{T}}}) = \mathsf{K}\bot$, so in particular $F^{\mathcal{T}}(\bot_{\mathcal{F}^{\mathcal{T}}})(\uparrow C) = \bot_{\mathcal{F}^{\mathcal{T}}}$, and therefore

$$\begin{aligned} \{D \mid \top \leq D\} &= \uparrow\top = \bot_{\mathcal{F}^{\mathcal{T}}} \\ &= F^{\mathcal{T}}(\bot_{\mathcal{F}^{\mathcal{T}}})(\uparrow C) \\ &= F^{\mathcal{T}}(\uparrow\top)(\uparrow C) \\ &= \{d \mid \top \leq (C \to D)\}, && \text{by Definition 18.2.2(i).} \end{aligned}$$

But then $\top \leq C \to D \;\Rightarrow\; \top \leq D$.

($\Leftarrow$) Suppose that $\top \le C{\to}D \;\Rightarrow\; \top \le D$. We show that $F^{\mathcal{T}}(\bot_{\mathcal{F}^{\mathcal{T}}}) = \mathsf{K}\bot$.

$$\begin{array}{rcll}
F^{\mathcal{T}}(\bot_{\mathcal{F}^{\mathcal{T}}})(X) & = & \{B \mid \exists A \in X.(A{\to}B) \in \bot_{\mathcal{F}^{\mathcal{T}}}\} & \\
& = & \{B \mid \exists A \in X.(A{\to}B) \in \uparrow\!\top\} & \\
& = & \{B \mid \exists A \in X.\top \le (A{\to}B)\} & \\
& = & \{B \mid \top \le B\} & \text{by the assumption} \\
& = & \uparrow\!\top = \bot_{\mathcal{F}^{\mathcal{T}}}. &
\end{array}$$

(ii) Suppose that $\top{\to}B \le C{\to}D \;\Rightarrow\; B \le D$. We first show that each compact constant function $\mathsf{K}\uparrow\!B (= \boldsymbol{\lambda} X \in \mathcal{F}^{\mathcal{T}}.\uparrow\!B)$ is represented by $\uparrow\!(\top{\to}B)$. Indeed,

$$\begin{array}{rcll}
D \in \uparrow\!(\top{\to}B)\cdot\uparrow\!C & \Leftrightarrow & C{\to}D \in \uparrow\!(\top{\to}B) & \text{by } (\to) \\
& \Leftrightarrow & \top{\to}B \le C{\to}D & \\
& \Leftrightarrow & B \le D & \text{by the assumption, } (\top)\text{, and } (\to) \\
& \Leftrightarrow & D \in \uparrow\!B = (\mathsf{K}\uparrow\!B)\,(\uparrow\!C). &
\end{array}$$

Now let $\mathsf{K}\,X$ be an arbitrary constant function, where X is directed. Then $\mathsf{K}\,X = \sqcup_{B \in X}\mathsf{K}\uparrow\!B$ and $\{\mathsf{K}\uparrow\!B \mid B \in X\}$ is directed. Hence by Lemma 18.2.19 and the continuity of $F^{\mathcal{T}} \circ G^{\mathcal{T}}$ we get

$$\begin{array}{rcl}
\mathsf{K}\,X & = & \sqcup_{B \in X}\mathsf{K}\uparrow\!B \\
& = & \sqcup_{B \in X}F^{\mathcal{T}} \circ G^{\mathcal{T}}(\mathsf{K}\uparrow\!B) \\
& = & F^{\mathcal{T}} \circ G^{\mathcal{T}}(\sqcup_{B \in X}\mathsf{K}\uparrow\!B).
\end{array}$$

Conversely, suppose that all constant functions are representable. Then $F^{\mathcal{T}} \circ G^{\mathcal{T}}(\mathsf{K}\uparrow\!b) = \mathsf{K}\uparrow\!b$, by Lemma 18.2.19. Therefore

$$\begin{array}{rcll}
\top{\to}B \le C{\to}D & \Rightarrow & (C{\to}D) \in \uparrow\!(\top{\to}B) & \\
& \Rightarrow & d \in \uparrow\!(\top{\to}B)\cdot\uparrow\!C & \\
& \Rightarrow & d \in ((F^{\mathcal{T}} \circ G^{\mathcal{T}})(\mathsf{K}\uparrow\!B))(\uparrow\!C), & \text{by Definition 18.2.2(i),} \\
& \Rightarrow & D \in (\mathsf{K}\uparrow\!B)(\uparrow\!C) = \uparrow\!B & \\
& \Rightarrow & B \le D. &
\end{array}$$

(iv) (We prove (iii) later.) Let $\mathcal{T} \in \mathrm{NTT}$. Using the axioms of the natural type structure $\mathcal{T}$, it is not difficult to see that the condition in the RHS of (iv) is equivalent with

$$(A_1{\to}B_1)\cap\ldots\cap(A_n{\to}B_n) \le (C{\to}D) \;\Leftrightarrow\; B_{i1}\cap\ldots\cap B_{ik} \le D \;\&\; C \le A_{i1}\cap\ldots\cap A_{ik}. \tag{18.1}$$

($\Leftarrow$ always does hold) Let $h \in \mathcal{K}([\mathcal{F}^{\mathcal{T}}{\to}\mathcal{F}^{\mathcal{T}}])$. By Lemma 18.2.20 Proposition 17.1.11(ii) it follows that for some $A_1,\ldots,A_n, B_1,\ldots,B_n \in \mathcal{T}$

$$h = (\uparrow\!A_1 \Rightarrow \uparrow\!B_1) \sqcup \ldots \sqcup (\uparrow\!A_n \Rightarrow \uparrow\!B_n),$$

since a finite element of $\mathcal{F}^{\mathcal{T}}$ is of the form $\uparrow\!A$.

($\Rightarrow$) Suppose all continuous functions are representable. Then since h above is such a function, one has

$$(F^{\mathcal{S}} \circ G^{\mathcal{S}})(h)(\uparrow c) = h(\uparrow c)$$

and therefore (18.1).

($\Leftarrow$) Suppose we have (18.1). Then for the compact continuous functions h one has that $(F^{\mathcal{S}} \circ G^{\mathcal{S}})(h)$ and h coincide on the compact elements $\uparrow c$ and therefore by Proposition 17.1.5 everywhere. But then it follows again that $f = (F^{\mathcal{S}} \circ G^{\mathcal{S}})(f)$ for every continuous $f : \mathcal{F}^{\mathcal{T}} \to \mathcal{F}^{\mathcal{T}}$. Hence Lemma 18.2.19 applies.

(iii) ($\Rightarrow$) Suppose all continuous step functions are representable. Suppose $a \to b \leq c \to d$, $d \neq \top$. Take $h = \uparrow a \Rightarrow \uparrow b$. By Lemma 18.2.20(ii) we have

$$\begin{array}{rcl} (F^{\mathcal{S}} \circ G^{\mathcal{S}})(h)(\uparrow c) & = & \{e \mid a \to b \leq c \to e\} \\ h(\uparrow c) & = & \{e \mid \bigcap\{b \mid c \leq a\} \leq e\}. \end{array}$$

By the first assumption these two sets are equal. By the second assumption it follows that $c \leq a$ & $b \leq d$.

($\Leftarrow$) Let $h = X \Rightarrow Y$ be continuous. We have to show that

$$(F^{\mathcal{S}} \circ G^{\mathcal{S}})(h) = h \tag{18.1}$$

By Lemma 17.1.5(ii) it suffices to show this for compact h. If $Y \neq \perp_{\mathcal{F}^{\mathcal{T}}}$, then by 17.1.9 both X, Y are compact, so $h = \uparrow a \Rightarrow \uparrow b$. Then (18.1) holds by Lemma 18.2.20 and the assumption. If $Y = \perp_{\mathcal{F}^{\mathcal{T}}}$, then h is the bottom function and hence representable (the assumption in (iii) implies the assumption in (i)). ■

A non-continuous filter lambda-model

The intersection type theories $\mathcal{T} \in \{\text{Scott}, \text{Park}, \text{BCD}, \text{CDZ}, \text{HR}, \text{AO}, \text{DHM}\}$ all induce filter lambda-models, by Corollary 18.2.12(i). These type theories are all natural and $\boldsymbol{\beta}$-sound. Therefore by Theorem 18.2.21(iv) all continuous functions are representable in these $\mathcal{F}^{\mathcal{T}}$. In Section 18 we will give many more filter lambda-models arising from domain models. It is therefore interesting to ask whether there exist filter lambda-models where *not all* continuous functions are representable. We answer affirmatively, and end this section by giving an example of a natural type structure which is not $\boldsymbol{\beta}$-sound. Therefore by the same theorem not all continuous functions are representable in the induced filter structures. The model builds on an idea in Coppo et al. [1984]. In exercise 18.4.7 another such model, due to Alessi [1993], is constructed. See also Alessi, Barbanera and Dezani-Ciancaglini [2004].

The theory $\diamondsuit$

18.2.22. Definition. Let the TT $\diamondsuit$ with constants $\mathbb{A}^{\diamondsuit} = \{\top, \diamondsuit, \heartsuit\}$ be axiomatized by rule $(\to)$ and axioms $(\to\cap), (\top), (\top \to)$ and $(\diamondsuit)$ where

$$(\diamondsuit) \; A \leq_{\diamondsuit} A[\diamondsuit := \heartsuit].$$

18.2.23. LEMMA. (i) $A \leq_{\diamondsuit} B \quad \Rightarrow \quad A[\diamondsuit := \heartsuit] \leq_{\diamondsuit} B[\diamondsuit := \heartsuit]$.

(ii) $\Gamma \vdash^{\diamondsuit} M : A \quad \Rightarrow \quad \Gamma[\diamondsuit := \heartsuit] \vdash^{\diamondsuit} M : A[\diamondsuit := \heartsuit]$.

(iii) $\Gamma, \Gamma' \vdash^{\diamondsuit} M : A \quad \Rightarrow \quad \Gamma, \Gamma'[\diamondsuit := \heartsuit] \vdash^{\diamondsuit} M : A[\diamondsuit := \heartsuit]$.

(iv) $\Gamma, x{:}A_i \vdash^{\diamondsuit} M : B_i$ *for* $1 \leq i \leq n$ & $(A_1 \to B_1) \cap \ldots \cap (A_n \to B_n) \leq_{\diamondsuit} C \to D$
$\Rightarrow \ \Gamma, x{:}C \vdash^{\diamondsuit} M : D$.

PROOF. (i) By induction on the definition of $\leq_{\diamondsuit}$.

(ii) By induction on derivations using (i) for rule $(\leq_{\diamondsuit})$.

(iii) From (ii) and rule $(\leq_{\diamondsuit}\text{-L})$, taking into account that if $(x{:}B) \in \Gamma$, then $(x{:}B[\diamondsuit := \heartsuit]) \in \Gamma[\diamondsuit := \heartsuit]$ and $B \leq_{\diamondsuit} B[\diamondsuit := \heartsuit]$.

(iv) We will denote by α, β (possibly with indices) elements of $\mathbb{A}^{\diamondsuit}$. We show by induction on the definition of $\leq_{\diamondsuit}$ that

$$\Big[\Big(\textstyle\bigcap_{i \in I}(A_i \to B_i) \cap^{\top} (\bigcap_{h \in H} \alpha_h)\Big) \leq_{\diamondsuit} \Big(\bigcap_{j \in J}(C_j \to D_j) \cap^{\top} (\bigcap_{k \in K} \beta_k)\Big)$$
$$\& \ \forall i \in I.\, \Gamma, x{:}A_i \vdash^{\diamondsuit} M : B_i\Big] \ \Rightarrow \ \forall j \in J.\, \Gamma, x{:}C_j \vdash^{\diamondsuit} M : D_j.$$

The only interesting case is when the applied rule is $(\diamondsuit)$, i.e. we have

$$\Big(\textstyle\bigcap_{i \in I}(A_i \to B_i) \cap^{\top} (\bigcap_{h \in H} \alpha_h)\Big) \leq_{\diamondsuit} \Big(\bigcap_{i \in I}(A_i \to B_i) \cap^{\top} (\bigcap_{h \in H} \alpha_h)\Big) [\diamondsuit := \heartsuit].$$

By hypothesis $\Gamma, x{:}A_i \vdash^{\diamondsuit} M : B_i$, so we are done by (iii). ■

18.2.24. THEOREM. (i) $\mathcal{T}^{\diamondsuit}$ *is a* TT *that is not* $\boldsymbol{\beta}$*-sound.*

(ii) *Nevertheless* $\mathcal{F}^{\diamondsuit}$ *is a filter lambda-model.*

PROOF. (i) By definition $\mathcal{T}^{\diamondsuit}$ is a TT. We have $\diamondsuit \to \diamondsuit \leq_{\diamondsuit} \heartsuit \to \heartsuit$, but $\heartsuit \not\leq_{\diamondsuit} \diamondsuit$, so it is not $\boldsymbol{\beta}$-sound.

(ii) To show that $\mathcal{F}^{\diamondsuit}$ is a lambda-model, it suffices, by Proposition 18.2.9, to verify that $\Gamma \vdash^{\diamondsuit} \lambda x.M : A \to B \ \Rightarrow \ \Gamma, x{:}A \vdash^{\diamondsuit} M : B$. By Lemma 16.1.1(iii) $\Gamma \vdash^{\diamondsuit} \lambda x.M : A \to B \ \Rightarrow \ \Gamma, x{:}C_i \vdash^{\diamondsuit} M : D_i$ for some I, C_i, D_i such that $\bigcap_{i \in I}(C_i \to D_i) \leq_{\diamondsuit} A \to B$. So, we are done by Lemma 18.2.23(iv). ■

For example the step function $\uparrow\diamondsuit \Rightarrow \uparrow\diamondsuit$ is not representable in $\mathcal{F}^{\diamondsuit}$.

18.3. Approximation theorems

A crucial result in the study of the equational theory of ω-algebraic λ-models are the *Approximation Theorems*, see e.g. Hyland [1975/76], Wadsworth [1976], Barendregt [1984], Longo [1987], Ronchi Della Rocca [1988], Honsell and Ronchi Della Rocca [1992]. An Approximation Theorem expresses 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 e.g. Barendregt [1984], Honsell and Ronchi Della Rocca [1992]. There are other possible methods of showing computational adequacy, both semantical and syntactical, e.g. Hyland [1975/76], Wadsworth [1976], Honsell and Ronchi

Della Rocca [1992], Abramsky and Ong [1993], but the method based on Approximation Theorems is usually the most straightforward. However, proving an Approximation Theorem for a given model theory is usually rather difficult. Most of the proofs in the literature are based on the technique of *indexed reduction*, see Wadsworth [1976], Abramsky and Ong [1993], Honsell and Ronchi Della Rocca [1992]. 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 using the syntactical machinery of proof theory, such as *logical predicates* and *computability* techniques.

The aim of the present section is to show in a uniform way that all the type assignment systems which induce filter models isomorphic to the models in Scott [1972], Park [1976], Coppo et al. [1987], Honsell and Ronchi Della Rocca [1992], Dezani-Ciancaglini et al. [2005], Barendregt et al. [1983], Abramsky and Ong [1993] satisfy the Approximation Theorem. To this end following Dezani-Ciancaglini et al. [2001] we use a technique which can be constructed as a version of stable sets over a Kripke applicative structure. **Comment:** Add relations with Ronchi Della Rocca and Paolini [2004].

For almost all the type theories of Figure 15.2 which induce λ-models 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. We end this section showing some applications of the Approximation Theorem.

Approximate normal forms

In this section we consider two extensions of λ-calculus both obtained by adding one constant. The first one is the well known language $\lambda\bot$, see Barendregt [1984]. The other extension is obtained by adding Φ and it is discussed in Honsell and Ronchi Della Rocca [1992].

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

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

We consider two mappings ($\boxtimes$ and $\boxplus$) from λ-terms to $\lambda\bot$-terms and one mapping ($\boxdot$) from λ-terms to $\lambda\Phi$-terms. These mappings differ in the translation of $\boldsymbol{\beta}$-redexes. Clearly the values of these mappings are $\boldsymbol{\beta}$-irreducible terms, i.e. normal forms for an extended language. As usual we call such a term an *approximate normal form* or abbreviated an *anf*.

18.3.2. Definition. The mappings $\boxtimes : \Lambda \to \Lambda\bot, \boxplus : \Lambda \to \Lambda\bot, \boxdot : \Lambda \to \Lambda\Phi$ are inductively defined as follows.

$$
\begin{array}{rcl}
\dagger(\lambda\vec{x}.y\vec{M}) & = & \lambda\vec{x}.y\dagger(M_1)\dots\dagger(M_m);\\
\boxtimes(\lambda\vec{x}.(\lambda y.R)N\vec{M}) & = & \bot;\\
\boxplus(\lambda\vec{x}.(\lambda y.R)N\vec{M}) & = & \lambda\vec{x}.\bot;\\
\boxdot(\lambda\vec{x}.(\lambda y.R)N\vec{M}) & = & \lambda\vec{x}.\Phi\boxdot(\lambda y.R)\boxdot(N)\boxdot(M_1)\dots\boxdot(M_m),
\end{array}
$$

where $\dagger \in \{\boxtimes, \boxplus, \boxdot\}$, $\vec{M} \equiv M_1 \dots M_m$ and $m \geq 0$.

The mapping $\boxtimes$ is related to the Böhm-tree of untyped lambda terms, whereas $\boldsymbol{\beta}$ to the Lévy-Longo trees, see van Bakel et al. [2002], where these trees are related to intersection types.

In order to give the appropriate Approximation Theorem we will use the mapping $\boxtimes$ for the type assignment systems $\lambda_{\cap}^{\mathrm{Scott}}$, $\lambda_{\cap}^{\mathrm{CDZ}}$, $\lambda_{\cap}^{\mathrm{DHM}}$, $\lambda_{\cap}^{\mathrm{BCD}}$ the mapping $\boxplus$ for the type assignment system $\lambda_{\cap}^{\mathrm{AO}}$, and the mapping $\boxdot$ for the type assignment systems $\lambda_{\cap}^{\mathrm{Park}}$, $\lambda_{\cap}^{\mathrm{HR}}$. Each one of the above mappings associates a set of approximants to each λ-term in the standard way.

Comment: the order here is that one of the figure but for AO, Park, HR which are at the end, since this corresponds to the 3 mappings, would it be better to change?

18.3.3. Definition. Let $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Park}, \mathrm{HR}\}$.
The set $\mathcal{A}_{\mathcal{S}}(M)$ of $\mathcal{S}$-*approximants* of M is defined by

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

where

$$
\begin{array}{rcll}
\dagger & = & \boxtimes, & \text{for } \mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}\},\\
\dagger & = & \boxplus, & \text{for } \mathcal{S} \in \{\mathrm{AO}\},\\
\dagger & = & \boxdot, & \text{for } \mathcal{S} \in \{\mathrm{Park}, \mathrm{HR}\}. \blacksquare
\end{array}
$$

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

18.3.4. Definition. (i) Let $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}\}$. We extend the definition of type assignment $\Gamma \vdash_{\cap}^{\mathcal{S}} M : A$ to $\lambda\bot$-terms by letting M, N in Definition 15.2.2 range over $\Lambda\bot$.

(ii) We extend the type assignment $\lambda_{\cap}^{\mathrm{Park}}$ to $\lambda\Phi$-terms by adding the axiom

(Ax-Φ-Park) $\quad \Gamma \vdash_{\cap}^{\mathrm{Park}} \Phi : \omega$.

(iii) We extend the type assignment $\lambda_{\cap}^{\mathrm{HR}}$ to $\lambda\Phi$-terms by adding the axiom

(Ax-Φ-HR) $\quad \Gamma \vdash_{\cap}^{\mathrm{HR}} \Phi : \varphi$. $\blacksquare$

It is easy to verify that the appropriate generalization of the Inversion Theorems (Theorems 16.1.1 and 16.1.10) holds also for these two extensions of the type assignment system. Therefore we do not introduce different notations for these extended type assignment systems and we will freely refer to the Inversion Theorems extended with the following proposition.

18.3.5. Proposition. (i) *Let* $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{BCD}, \text{AO}\}$. *Then*

$$\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} \bot : A \;\Leftrightarrow\; A =_{\mathcal{S}} \top.$$

(ii) $\Gamma \vdash^{\text{Park}}_{\cap^{\top}} \Phi : A \;\Leftrightarrow\; \omega \leq_{\text{Park}} A.$

(iii) $\Gamma \vdash^{\text{HR}}_{\cap^{\top}} \Phi : A \;\Leftrightarrow\; \varphi \leq_{\text{HR}} A.$ ∎

18.3.6. Lemma. *Let* $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{BCD}, \text{AO}, \text{Park}, \text{HR}\}$.

(i) $M_1 \twoheadrightarrow_{\boldsymbol{\beta}} M_2 \;\&\; \Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} \dagger(M_1) : A \;\Rightarrow\; \Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} \dagger(M_2) : A.$

(ii) *If* $P, P' \in \mathcal{A}_{\mathcal{S}}(M)$, $\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} P : A$ *and* $\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} P' : B$, *then*

$$\exists P'' \in \mathcal{A}_{\mathcal{S}}(M).\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} P'' : A \cap B.$$

Proof. (i). For $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{BCD}, \text{AO}\}$ the proof follows by induction on the structure of the term dividing by cases (head normal form or not head normal form) and using Inversion Theorems.

For $\mathcal{S} \in \{\text{Park}, \text{HR}\}$ it sufficies to consider the case $M_1 \equiv (\lambda x.M)N$ and $M_2 \equiv M[x := N]$. Notice that $\boxdot(M[x := N])$ is $\boxdot(M)$ where the occurrences of x have been replaced by $\Phi\boxdot(N)$ if they are functional and N is an abstraction, and by $\boxdot(N)$ otherwise. More formally, if we define the mapping $\overline{\boxdot} : \Lambda \to \Lambda\Phi$ by

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

and the mapping $\{\ \}^x_y : \Lambda \to \Lambda$ by

$$\begin{array}{lcl} \{z\}^x_y & = & z \\ \{M_1M_2\}^x_y & = & \begin{cases} y\{M_2\}^x_y & \text{if } M_1 \equiv x \\ \{M_1\}^x_y\{M_2\}^x_y & \text{otherwise} \end{cases} \\ \{\lambda z.M\}^x_y & = & \lambda z.\{M\}^x_y. \end{array}$$

then $\boxdot(M_1M_2) = \overline{\boxdot}(M_1)\boxdot(M_2)$ and one can check, by induction on M, that $\boxdot(M[x := N]) \equiv \boxdot(\{M\}^x_y)[x := \boxdot(N)][y := \overline{\boxdot}(N)]$ for y fresh.

We may assume $A \neq_{\mathcal{S}} \top$. Then from $\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} \Phi(\lambda x.\boxdot(M))\boxdot(N) : A$ we get $\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} \Phi(\lambda x.\boxdot(M)) : C \to A$, $\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} \boxdot(N) : C$ for some C, by Theorem 16.1.10(ii). By Lemma 15.1.16 we have $C \to A \neq_{\mathcal{S}} \top$, hence again by Theorem 16.1.10(ii) $\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} \Phi : B \to C \to A$, $\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} \lambda x.\boxdot(M) : B$, for some B.

For $\mathcal{S} = \text{Park}$ we get $\omega \leq_{\text{Park}} B \to C \to A$ from $\Gamma \vdash^{\text{Park}}_{\cap^{\top}} \Phi : B \to C \to A$ by Proposition 18.3.5(ii). This implies $B \leq_{\text{Park}} \omega$, $C \leq_{\text{Park}} \omega$, and $\omega \leq_{\text{Park}} A$, since $\omega =_{\mathcal{T}} \omega \to \omega$, the type structure Park is $\boldsymbol{\beta}$-sound by Theorem 16.1.8(ii), $(C {\to} A) \neq_{\mathcal{S}} \top$ and $A \neq_{\mathcal{S}} \top$. We obtain by rule $(\leq)$ $\Gamma \vdash^{\text{Park}}_{\cap^{\top}} \lambda x.\boxdot(M) : \omega$ and $\Gamma \vdash^{\text{Park}}_{\cap^{\top}} \boxdot(N) : \omega$. We get $\Gamma, x{:}\omega \vdash^{\text{Park}}_{\cap^{\top}} \boxdot(M) : \omega$ (by Theorem 16.1.10(ii)) and $\Gamma \vdash^{\text{Park}}_{\cap^{\top}} \Phi\boxdot(N) : \omega$ since $\omega =_{\mathcal{T}} \omega \to \omega$. Now $\boxdot(\{M\}^x_y)$ is equal to $\boxdot(M)$ with some occurrences of x replaced by the fresh variable y. Hence $\Gamma, y{:}\omega, x{:}\omega \vdash^{\text{Park}}_{\cap^{\top}} \boxdot(\{M\}^x_y){:}\omega$. So we conclude $\Gamma \vdash^{\text{Park}}_{\cap^{\top}} \boxdot(\{M\}^x_y)[x := \boxdot(N)][y := \overline{\boxdot}(N)] : A$ by rules (cut) and $(\leq)$.

For $\mathcal{S} = \mathrm{HR}$ we get $\varphi \leq_{\mathrm{HR}} B \to C \to A$ from $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \Phi : B \to C \to A$ by Theorem 18.3.5. This implies either ($B \leq_{\mathrm{HR}} \varphi$ and $\varphi \leq_{\mathrm{HR}} C \to A$) or ($B \leq_{\mathrm{HR}} \omega$ and $\omega \leq_{\mathrm{HR}} C \to A$) since $\varphi =_{\mathrm{HR}} (\varphi \to \varphi)\cap^\top(\omega \to \omega)$ and HR is $\boldsymbol{\beta}$-sound by Theorem 16.1.8(ii), $C{\to}A \neq_{\mathrm{HR}} \top$ and $A \neq_{\mathrm{HR}} \top$ (notice that $\varphi\cap^\top\omega = \omega$). Similarly in the first case from $\varphi \leq_{\mathrm{HR}} C \to A$ we get either $C \leq_{\mathrm{HR}} \varphi$ and $\varphi \leq_{\mathrm{HR}} A$ or $C \leq_{\mathrm{HR}} \omega$ and $\omega \leq_{\mathrm{HR}} A$. In the second case from $\omega \leq_{\mathrm{HR}} C \to A$ we get $C \leq_{\mathrm{HR}} \varphi$ and $\omega \leq_{\mathrm{HR}} A$ since $\omega =_{\mathrm{HR}} \varphi \to \omega$.

To sum up, using rule $(\leq)$ we have the following alternative cases.

- $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \lambda x.\boxdot(M) : \varphi$, $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \boxdot(N) : \varphi$, and $\varphi \leq_{\mathrm{HR}} A$;
- $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \lambda x.\boxdot(M) : \varphi$, $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \boxdot(N) : \omega$, and $\omega \leq_{\mathrm{HR}} A$;
- $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \lambda x.\boxdot(M) : \omega$, $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \boxdot(N) : \varphi$, and $\omega \leq_{\mathrm{HR}} A$.

Therefore we get alternatively:

- $\Gamma, x{:}\varphi \vdash^{\mathrm{HR}}_{\cap^\top} \boxdot(M) : \varphi$, and $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \Phi\boxdot(N) : \varphi$;
- $\Gamma, x{:}\omega \vdash^{\mathrm{HR}}_{\cap^\top} \boxdot(M) : \omega$, and $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \Phi\boxdot(N) : \omega$;
- $\Gamma, x{:}\varphi \vdash^{\mathrm{HR}}_{\cap^\top} \boxdot(M) : \omega$, and $\Gamma \vdash^{\mathrm{HR}}_{\cap^\top} \Phi\boxdot(N) : \varphi$,

so we can conclude as in the previous case.

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

Approximation Theorem - Part 1

It is useful to introduce the following definition.

18.3.7. Definition. Let $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Park}, \mathrm{HR}\}$. Write

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

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

In this subsection we prove that, if $M \in [A]^{\mathcal{S}}_{\Gamma}$, then there exists a derivation of $\Gamma \vdash^{\mathcal{S}}_{\cap^\top} M : A$.

18.3.8. Proposition. *Let* $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Park}, \mathrm{HR}\}$.

$$M \in [A]^{\mathcal{S}}_{\Gamma} \Rightarrow \Gamma \vdash^{\mathcal{S}}_{\cap^\top} M : A.$$

Proof. By Corollary 16.2.14(ii) it is sufficient to show

$$\Gamma \vdash^{\mathcal{S}}_{\cap^\top} P : A \Rightarrow \Gamma \vdash^{\mathcal{T}}_{\cap^\top} M : A. \tag{18.2}$$

where $P \equiv \dagger(M)$ and
$\dagger = \boxtimes$ for $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}\}$,
$\dagger = \boxplus$ for $\mathcal{S} = \mathrm{AO}$,
$\dagger = \boxdot$ for $\mathcal{S} \in \{\mathrm{Park}, \mathrm{HR}\}$.

For $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{BCD}, \text{AO}\}$ from $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P : A$ we get $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M : A$ by Proposition 18.3.5(i) and the definition of the mappings $\boxtimes$ and $\boxplus$.
For $\mathcal{S} \in \{\text{Park}, \text{HR}\}$ we prove (18.2) by induction on M, assuming $A \neq_{\mathcal{S}} \top$.

Case $M \equiv x$ is trivial.

Case $M \equiv \lambda x.M'$, then $P \equiv \lambda x.P'$ where $P' \equiv \boxdot(M')$. By Theorem 16.1.1(iii) from $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P : A$ we get $\Gamma, x{:}B_i \vdash^{\mathcal{T}}_{\cap^\top} P' : C_i$ and $\bigcap_{i \in I}(B_i \to C_i) \leq_{\mathcal{S}} A$ for some I, B_i, C_i. We get by induction $\Gamma, x{:}B_i \vdash^{\mathcal{T}}_{\cap^\top} M' : C_i$ and so we conclude $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M : A$ using rules $(\to\text{I})$, $(\cap\text{I})$ and $(\leq_{\mathcal{S}})$.

Case $M \equiv M_1 M_2$ where M_1 is not an abstraction. Then $P \equiv P_1 P_2$ where $P_1 \equiv \boxdot(M_1)$ and $P_2 \equiv \boxdot(M_2)$. By Theorem 16.1.10(ii) from $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P : A$ we get $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P_1 : B \to A$, $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P_2 : B$ for some B. By induction this implies $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M_1 : B \to A$ and $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M_2 : B$, hence $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M \equiv M_1 M_2 : A$.

Case $M \equiv M_1 M_2$ where M_1 is an abstraction. Then $P \equiv \Phi P_1 P_2$ where $P_1 \equiv \boxdot(M_1)$ and $P_2 \equiv \boxdot(M_2)$. As in the proof of Lemma 18.3.6(i) from $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P : A$, where $A \neq_{\mathcal{S}} \top$, we get $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} \Phi : B \to C \to A$, $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P_1 : B$, $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P_2 : C$ for some B, C. By induction this implies $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M_1 : B$ and $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M_2 : C$.

For $\mathcal{S} = \text{Park}$, as in the proof of Lemma 18.3.6(i), we get $\Gamma \vdash^{\text{Park}}_{\Omega} M_1 : \omega$ and $\Gamma \vdash^{\text{Park}}_{\Omega} M_2 : \omega$. We can conclude $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M : A$ using rules $(\leq_{\text{Park}})$ and $(\to\text{E})$ since $\omega =_{\text{Park}} \omega \to \omega$.

For $\mathcal{S} = \text{HR}$ as in the proof of Lemma 18.3.6(i) we have the following alternative cases.

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

It is easy to verify that in all cases we can derive $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M : A$ from (I) and $(\varphi \to \omega)$ using rules $(\leq_{\text{HR}})$ and $(\to\text{E})$. ∎

Approximation Theorem - Part 2

In order to prove the converse of Proposition 18.3.8 we will use a Kripke-like version of stable sets Mitchell [1996]. First we need a technical result.

18.3.9. Lemma. *Let* $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{AO}, \text{BCD}, \text{Park}, \text{HR}\}$. *If* $\Gamma' = \Gamma, z : B$, $z \notin \text{FV}(M)$, *and* $A \neq_{\mathcal{S}} \top$ *for* $\mathcal{S} = \text{AO}$, *then* $Mz \in [A]^{\mathcal{S}}_{\Gamma'}$ *implies* $M \in [B \to A]^{\mathcal{S}}_{\Gamma}$.

Proof. Let $P \in \mathcal{A}_{\mathcal{S}}(Mz)$ and $\Gamma' \vdash^{\mathcal{T}}_{\cap^\top} P : A$. We show by cases on P and M that there is $\hat{P} \in \mathcal{A}_{\mathcal{S}}(M)$ such that $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} \hat{P} : B \to A$.

There are two possibilities.

- $Mz \twoheadrightarrow_{\boldsymbol{\beta}} M'z$ and $P \equiv \dagger(M'z)$;
- $Mz \twoheadrightarrow_{\boldsymbol{\beta}} (\lambda x.M')z \to_{\boldsymbol{\beta}} M'[x := z]$ and $P \in \mathcal{A}_{\mathcal{S}}(M'[x := z])$.

In the first case again there are two possibilities.

- $M' \equiv yM_1 \dots M_m, \quad m \geq 0;$
- $M' \equiv (\lambda y.M_0)M_1 \dots M_m, \quad m \geq 0.$

In total there are 4 cases:

- $P \equiv P'z$ and $P' \in \mathcal{A}_{\mathcal{S}}(M)$;
- $P \equiv \bot$ and $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{BCD}, \text{AO}\}$;
- $P \equiv \Phi P'z$, $P' \in \mathcal{A}_{\mathcal{S}}(M)$ and $\mathcal{S} \in \{\text{Park}, \text{HR}\}$;
- $M \twoheadrightarrow_{\boldsymbol{\beta}} \lambda x.M'$ and $P \in \mathcal{A}_{\mathcal{S}}(M'[x := z])$.

Case $P \equiv P'z$ where $P' \in \mathcal{A}_{\mathcal{S}}(M)$, then we can choose $\hat{P} \equiv P'$. This is clear if $A =_{\mathcal{S}} \top$ because then $\mathcal{S} \neq \text{AO}$, hence we have $(\top{-}\eta)$. Now let $A \neq_{\mathcal{S}} \top$. Then by Theorem 16.1.10(ii) from $\Gamma' \vdash^{\mathcal{T}}_{\cap^\top} P : A$ we get $\Gamma' \vdash^{\mathcal{T}}_{\cap^\top} P' : C \to A$, $\Gamma' \vdash^{\mathcal{T}}_{\cap^\top} z : C$ for some C. $B \leq_{\mathcal{S}} C$ by Theorem 16.1.10(i) and we conclude using $(\leq)$ and (*strenghtening*) $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P' : B \to A$.

Case $P \equiv \bot$ is trivial for $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{BCD}\}$ because of $(\top{-}\eta)$ and impossible for $\mathcal{S} = \text{AO}$ by Proposition 18.3.5(i).

Case $P \equiv \Phi P'z$ where $P' \in \mathcal{A}_{\mathcal{S}}(M)$ and $\mathcal{S} \in \{\text{Park}, \text{HR}\}$, then we show that we can choose $\hat{P} \equiv P'$. Again let $A \neq_{\mathcal{S}} \top$. Then from $\Gamma' \vdash^{\mathcal{T}}_{\cap^\top} P : A$ we get by Theorem 16.1.10(ii) and (i) $\Gamma' \vdash^{\mathcal{T}}_{\cap^\top} \Phi : C \to D \to A$, $\Gamma' \vdash^{\mathcal{T}}_{\cap^\top} P' : C$, $\Gamma' \vdash^{\mathcal{T}}_{\cap^\top} z : D$ for some C, D with $B \leq_{\mathcal{S}} D$. For $\mathcal{S} = \text{Park}$ using Proposition 18.3.5(ii) as in the proof of Lemma 18.3.6(i), we get $C \leq_{\text{Park}} \omega$, $D \leq_{\text{Park}} \omega$, and $\omega \leq_{\text{Park}} A$ (remember that $\omega =_{\text{Park}} \omega{\to}\omega$). Similarly for $\mathcal{S} = \text{HR}$, using Proposition 18.3.5(iii), we get either $C \leq_{\text{HR}} \varphi$, $D \leq_{\text{HR}} \varphi$, and $\varphi \leq_{\text{Park}} A$ or $C \leq_{\text{HR}} \varphi$, $D \leq_{\text{HR}} \omega$, and $\omega \leq_{\text{HR}} A$ or $C \leq_{\text{Park}} \omega$, $D \leq_{\text{HR}} \varphi$, and $\omega \leq_{\text{Park}} A$ (remember that $\varphi =_{\text{HR}} (\varphi{\to}\varphi){\cap}^\top(\omega{\to}\omega)$ and $\omega =_{\text{HR}} \varphi{\to}\omega$). In all cases we can conclude $C \leq D \to A \leq B \to A$ and therefore by $(\leq)$ and (*strenghtening*) $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} P' : B \to A$.

Case $M \twoheadrightarrow_{\boldsymbol{\beta}} \lambda x.M'$ and $P \in \mathcal{A}_{\mathcal{S}}(M'[x := z])$.If $\dagger = \boxtimes$ and $P \equiv \bot$, then we choose $\hat{P} \equiv P$. Else we choose $\hat{P} \equiv \lambda z.P$. ∎

The following crucial definition is somewhat involved. It amounts essentially to the definition of the natural set-theoretic semantics of intersection types over a suitable Kripke applicative structure, where bases play the role of worlds.[1] 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.

[1]As already observed in Berline [2000]) we cannot use here stable sets as we will do in section 19.2 since we need to take into account also the $\mathcal{S}$-bases, not only the λ-terms and their types.

18.3.10. DEFINITION (Kripke type interpretation).
Let $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{BCD}, \text{AO}, \text{Park}, \text{HR}\}$. Define

$$
\begin{array}{rcll}
[\![\alpha]\!]^{\mathcal{S}}_{\Gamma} & = & [\alpha]^{\mathcal{S}}_{\Gamma} & \text{for } \alpha \in \mathbb{A}_\infty \cup \{\top, \varphi\} \\
[\![\omega]\!]^{\mathcal{S}}_{\Gamma} & = & \{M \mid \forall \vec{N}. M\vec{N} \in [\omega]^{\mathcal{S}}_{\Gamma}\} & \text{for } \mathcal{S} \in \{\text{Scott}, \text{DHM}\}; \\
[\![\omega]\!]^{\mathcal{S}}_{\Gamma} & = & \{M \mid \forall \Gamma' \Supset \Gamma\, \forall \vec{N} \in [\varphi]^{\mathcal{S}}_{\Gamma'}. M\vec{N} \in [\omega]^{\mathcal{S}}_{\Gamma'}\} & \text{for } \mathcal{S} \in \{\text{CDZ}, \text{HR}\} \\
[\![\omega]\!]^{\text{Park}}_{\Gamma} & = & [\omega]^{\text{Park}}_{\Gamma} & \\
[\![A \to B]\!]^{\mathcal{S}}_{\Gamma} & = & \{M \mid \forall \Gamma' \Supset \Gamma\, \forall N \in [\![A]\!]^{\mathcal{S}}_{\Gamma'}. MN \in [\![B]\!]^{\mathcal{S}}_{\Gamma'}\}, & \text{if } \mathcal{S} \neq \text{AO or } B \neq_{\text{AO}} \top; \\
[\![A \to B]\!]^{\text{AO}}_{\Gamma} & = & [A \to B]^{\text{AO}}_{\Gamma}, & \text{if } B =_{\text{AO}} \top; \\
[\![A \cap^{\top} B]\!]^{\mathcal{S}}_{\Gamma} & = & [\![A]\!]^{\mathcal{S}}_{\Gamma} \cap [\![B]\!]^{\mathcal{S}}_{\Gamma}. &
\end{array}
$$

If in clause four the condition $\mathcal{S} \neq \text{AO}$ or $B \neq_{\text{AO}} \top$ or alternatively the whole clause five is dropped, then Lemma 18.3.12(ii) would not hold for $\mathcal{S} = \text{AO}$.

It is easy to verify that:

18.3.11. PROPOSITION. (i) *$M \in [\![A]\!]^{\mathcal{S}}_{\Gamma}$ and $N \twoheadrightarrow_{\beta} M$ imply $N \in [\![A]\!]^{\mathcal{S}}_{\Gamma}$.*
(ii) *$\Gamma \Subset \Gamma'$ implies $[\![A]\!]^{\mathcal{S}}_{\Gamma} \subseteq [\![A]\!]^{\mathcal{S}}_{\Gamma'}$ for all types $A \in \mathbb{T}^{\mathcal{S}}$.*

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

The Lemmas 18.3.12, 18.3.15 and the final theorem are standard.

18.3.12. LEMMA. *Let* $\mathcal{S} \in \{\text{Scott}, \text{CDZ}, \text{DHM}, \text{BCD}, \text{AO}, \text{Park}, \text{HR}\}$. *Then*
(i) $x\vec{M} \in [A]^{\mathcal{S}}_{\Gamma} \;\Rightarrow\; x\vec{M} \in [\![A]\!]^{\mathcal{S}}_{\Gamma}$;
(ii) $[\![A]\!]^{\mathcal{S}}_{\Gamma} \subseteq [A]^{\mathcal{S}}_{\Gamma}$.

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

(i) Case $A \equiv \omega$ and $\mathcal{S} = \text{CDZ}$. Let $\Gamma' \Supset \Gamma$ and $\vec{N} \in [\varphi]^{\text{CDZ}}_{\Gamma'}$.

Clearly $P \in \mathcal{A}_{\text{CDZ}}(x\vec{M})$, $\vec{Q} \in \mathcal{A}_{\text{CDZ}}(\vec{N}) \Rightarrow P\vec{Q} \in \mathcal{A}_{\text{CDZ}}(x\vec{M}\vec{N})$. Hence

$$
\begin{array}{rcll}
x\vec{M} \in [\omega]^{\text{CDZ}}_{\Gamma} & \Rightarrow & x\vec{M}\vec{N} \in [\omega]^{\text{CDZ}}_{\Gamma'} & \text{by rules } (\leq_{\text{CDZ}}) \text{ and } (\to\text{E}) \\
 & & & \text{since } \omega =_{\text{CDZ}} \varphi \to \omega, \\
 & \Rightarrow & x\vec{M} \in [\![\omega]\!]^{\text{CDZ}}_{\Gamma} & \text{by Definition 18.3.10.}
\end{array}
$$

Case $A \equiv B \to C$. Let $\Gamma' \Supset \Gamma$ and $\mathcal{S} \neq \text{AO}$ or $C \neq_{\text{AO}} \top$ and let $N \in [\![B]\!]^{\mathcal{S}}_{\Gamma'}$. $[\![B]\!]^{\mathcal{S}}_{\Gamma'} \subseteq [B]^{\mathcal{S}}_{\Gamma'}$ by induction on (ii). Hence

$$
\begin{array}{rcll}
x\vec{M} \in [A]^{\mathcal{S}}_{\Gamma} & \Rightarrow & x\vec{M}N \in [C]^{\mathcal{S}}_{\Gamma'} & \text{by rule } (\to\text{E}), \\
 & \Rightarrow & x\vec{M}N \in [\![C]\!]^{\mathcal{S}}_{\Gamma'} & \text{by induction on (i)}, \\
 & \Rightarrow & x\vec{M} \in [\![B \to C]\!]^{\mathcal{S}}_{\Gamma} & \text{by Definition 18.3.10.}
\end{array}
$$

(ii) Case $A \equiv B{\to}C$ and $\mathcal{S} \neq$ AO or $C{\neq_{\mathrm{AO}}}\top$. Let $\Gamma' = \Gamma, z{:}\,B$ with z fresh, and suppose $M \in [\![B \to C]\!]^{\mathcal{S}}_{\Gamma}$; as $z \in [\![B]\!]^{\mathcal{S}}_{\Gamma,z:B}$ by induction on (i), we have

$$\begin{array}{llll} M \in [\![B \to C]\!]^{\mathcal{S}}_{\Gamma} \text{ and } z \in [\![B]\!]^{\mathcal{S}}_{\Gamma,z:B} & \Rightarrow & Mz \in [\![C]\!]^{\mathcal{S}}_{\Gamma'} & \text{by Definition 18.3.10,}\\ & \Rightarrow & Mz \in [C]^{\mathcal{S}}_{\Gamma'} & \text{by induction on (ii),}\\ & \Rightarrow & M \in [B{\to}C]^{\mathcal{S}}_{\Gamma} & \text{by Lemma 18.3.9.} \end{array}$$

Case $A \equiv B{\cap}C$. This follows from induction hypothesis and from $[\![B \cap C]\!]^{\mathcal{S}}_{\Gamma} = [\![B]\!]^{\mathcal{S}}_{\Gamma} \cap [\![C]\!]^{\mathcal{S}}_{\Gamma}$ by Lemma 18.3.6(ii). ∎

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

18.3.13. Lemma. (i) *Let $\mathcal{S} \in \{\mathrm{CDZ}, \mathrm{DHM}\}$. Then*

$$M \in [A]^{\mathcal{S}}_{\Gamma,z:\omega} \;\&\; N \in [\![\omega]\!]^{\mathcal{S}}_{\Gamma} \;\Rightarrow\; M[z := N] \in [A]^{\mathcal{S}}_{\Gamma}.$$

(ii) *Let $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Park}, \mathrm{HR}\}$. Then*

$$\forall A, B \in \mathbb{T}^{\mathcal{S}}[A \leq_{\mathcal{S}} B \;\Rightarrow\; [\![A]\!]^{\mathcal{S}}_{\Gamma} \subseteq [\![B]\!]^{\mathcal{S}}_{\Gamma}].$$

Proof. (i) We may assume $A{\neq_{\mathcal{S}}}\top$. For $\mathcal{S} = \mathrm{CDZ}$ if $M \in [A]^{\mathrm{CDZ}}_{\Gamma,z:\omega}$ there is $P \in \mathcal{A}_{\mathrm{CDZ}}(M)$ such that $\Gamma, z{:}\omega \vdash^{\mathrm{CDZ}}_{\cap\top} P : A$. The proof is by induction on P.

Case $P \equiv \bot$. Trivial.

Case $P \equiv \lambda x.P'$. Then $M \twoheadrightarrow_{\boldsymbol{\beta}} \lambda x.M'$ and $P' \in \mathcal{A}_{\mathrm{CDZ}}(M')$. From $\Gamma, z{:}\omega \vdash^{\mathrm{CDZ}}_{\cap\top} P : A$ we get $\Gamma, z{:}\omega, x{:}B_i \vdash^{\mathrm{CDZ}}_{\cap\top} P' : C_i$ and $\bigcap_{i\in I}(B_i{\to}C_i) \leq_{\mathrm{CDZ}} A$ for some I and $B_i, C_i \in \mathbb{T}^{\mathrm{CDZ}}$ by Theorem 16.1.1(iii). By induction for each $i \in I$ there is a $P_i \in \mathcal{A}_{\mathrm{CDZ}}(M'[z := N])$ such that $\Gamma, x{:}B_i \vdash^{\mathrm{CDZ}}_{\cap\top} P_i : C_i$. Let $P_i = \dagger(M_i)$, where $M'[z := N] \twoheadrightarrow_{\boldsymbol{\beta}} M_i$ and let M'' be a common reduct of the M_i and $P'' \equiv \dagger(M'')$. Then $P'' \in \mathcal{A}_{\mathrm{CDZ}}(M'[z := N])$ and $\Gamma, x{:}B_i \vdash^{\mathrm{CDZ}}_{\cap\top} P'' : C_i$, for all $i \in I$, by lemma 18.3.6(i). Clearly $\lambda x.P'' \in \mathcal{A}_{\mathrm{CDZ}}(M[z := N])$ and by construction $\Gamma \vdash^{\mathrm{CDZ}}_{\cap\top} \lambda x.P'' : A$.

Case $P \equiv x\vec{P}$, then $M \twoheadrightarrow_{\boldsymbol{\beta}} x\vec{M}$ and $\vec{P} \in \mathcal{A}_{\mathrm{CDZ}}(\vec{M})$. From $\Gamma, z{:}\omega \vdash^{\mathrm{CDZ}}_{\cap\top} P : A$ we get $\Gamma, z{:}\omega \vdash^{\mathrm{CDZ}}_{\cap\top} x : \vec{B} \to A$ and $\Gamma, z{:}\omega \vdash^{\mathrm{CDZ}}_{\cap\top} \vec{P} : \vec{B}$ by Theorem 16.1.10(ii) and Lemma 15.1.16. By induction there are $\vec{P'} \in \mathcal{A}_{\mathrm{CDZ}}(\vec{M}[z := N])$ such that $\Gamma \vdash^{\mathrm{CDZ}}_{\cap\top} \vec{P'} : \vec{B}$. If $x \neq z$ we are done since $x\vec{P'} \in \mathcal{A}_{\mathrm{CDZ}}(M[z := N])$ and we can derive $\Gamma \vdash^{\mathrm{CDZ}}_{\cap\top} x\vec{P'} : A$ using ($\to$E). Otherwise $\Gamma, z{:}\omega \vdash^{\mathrm{CDZ}}_{\cap\top} z : \vec{B} \to A$ implies $\omega \leq_{\mathrm{CDZ}} \vec{B} \to A$ by Theorem 16.1.10(i). Being CDZ $\boldsymbol{\beta}$-sound by Theorem 16.1.8(ii) from $\omega =_{\mathrm{CDZ}} \vec{\varphi} \to \omega$ we obtain $\vec{B} \leq_{\mathrm{CDZ}} \vec{\varphi}$ and $\omega \leq_{\mathrm{CDZ}} A$ by Lemma 15.1.16. So we get $\Gamma \vdash^{\mathrm{CDZ}}_{\cap\top} \vec{P'} : \vec{\varphi}$, i.e. $\vec{M}[z := N] \in [\varphi]^{\mathrm{CDZ}}_{\Gamma}$. By Definition 18.3.10 $N \in [\![\omega]\!]^{\mathrm{CDZ}}_{\Gamma}$ and $\vec{M}[z := N] \in [\varphi]^{\mathrm{CDZ}}_{\Gamma}$ imply $M[z := N] \in [\omega]^{\mathrm{CDZ}}_{\Gamma}$. Since $\omega \leq_{\mathrm{CDZ}} A$ we get $M[z := N] \in [A]^{\mathrm{CDZ}}_{\Gamma}$.

The proof for $\mathcal{S} = \mathrm{DHM}$ is similar but easier than that for $\mathcal{S} = \mathrm{CDZ}$. In the case $P \equiv z\vec{P}$ it follows from Definition 18.3.10 that $N \in [\![\omega]\!]^{\mathrm{DHM}}_{\Gamma}$ implies $M[z := N] \in [A]^{\mathrm{DHM}}_{\Gamma}$.

(ii) We treat the cases related to $A{\to}B{\leq}\top{\to}\top$ in AO, $(\omega{\to}\varphi){=}\varphi, \omega{=}(\varphi{\to}\omega)$ in CDZ, $(\varphi{\to}\varphi) \cap (\omega{\to}\omega){=}\varphi$ in HR, and $(\omega{\to}\omega){=}\omega$ in Park.

Proof of $[\![A{\rightarrow}B]\!]^{\mathrm{AO}}_{\Gamma} \subseteq [\![\top{\rightarrow}\top]\!]^{\mathrm{AO}}_{\Gamma}$. If $B =_{\mathrm{AO}} \top$, then

$$[\![A{\rightarrow}B]\!]^{\mathrm{AO}}_{\Gamma} = [A{\rightarrow}B]^{\mathrm{AO}}_{\Gamma} \subseteq [\top{\rightarrow}\top]^{\mathrm{AO}}_{\Gamma} = [\![\top{\rightarrow}\top]\!]^{\mathrm{AO}}_{\Gamma}.$$

If on the other hand $B \neq_{\mathrm{AO}} \top$, then $M \in [\![A{\rightarrow}B]\!]^{\mathrm{AO}}_{\Gamma}$. Write $\Gamma' = \Gamma, z{:}A$. Then $z \in [A]^{\mathrm{AO}}_{\Gamma'}$ by Lemma 18.3.12(i), hence $z \in [\![A]\!]^{\mathrm{AO}}_{\Gamma'}$. So $Mz \in [\![B]\!]^{\mathrm{AO}}_{\Gamma'} \subseteq [B]^{\mathrm{AO}}_{\Gamma'}$ by Lemma 18.3.12(ii), and therefore by Lemma 18.3.9 we have $M \in [A{\rightarrow}B]^{\mathrm{AO}}_{\Gamma} \subseteq [\top{\rightarrow}\top]^{\mathrm{AO}}_{\Gamma} = [\![\top{\rightarrow}\top]\!]^{\mathrm{AO}}_{\Gamma}$.

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

$$\begin{array}{rcll} [\![\omega{\rightarrow}\varphi]\!]^{\mathrm{CDZ}}_{\Gamma} & \subseteq & [\omega{\rightarrow}\varphi]^{\mathrm{CDZ}}_{\Gamma}, & \text{by Lemma 18.3.12(ii)},\\ & = & [\varphi]^{\mathrm{CDZ}}_{\Gamma}, & \text{since } \omega{\rightarrow}\varphi =_{\mathrm{CDZ}} \varphi,\\ & = & [\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma}, & \text{by Definition 18.3.10.} \end{array}$$

Proof of $[\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma} \subseteq [\![\omega{\rightarrow}\varphi]\!]^{\mathrm{CDZ}}_{\Gamma}$. Suppose $\Gamma' \supseteq \Gamma$, $M \in [\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma}$ and $N \in [\![\omega]\!]^{\mathrm{CDZ}}_{\Gamma'}$ in order to show $MN \in [\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma'}$. By Definition 18.3.10 $[\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma} = [\varphi]^{\mathrm{CDZ}}_{\Gamma}$. If $M \in [\varphi]^{\mathrm{CDZ}}_{\Gamma}$, then there is $P \in \mathcal{A}_{\mathrm{CDZ}}(M)$ such that $\Gamma \vdash^{\mathrm{CDZ}}_{\cap\top} P : \varphi$. We will show $MN \in [\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma'}$ by cases of P

Case $P \equiv \bot$. By Proposition 18.3.5(i) one has $\varphi =_{\mathrm{CDZ}} \top$, and this contradicts Proposition 15.1.20. So this case is impossible. Then $[\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma'} = [\varphi]^{\mathrm{CDZ}}_{\Gamma'} = \Lambda$. Hence $MN \in [\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma'}$.

Case $P \equiv \lambda z.P'$. Then $M \twoheadrightarrow_{\boldsymbol{\beta}} \lambda z.M'$ and $P' \in \mathcal{A}_{\mathrm{CDZ}}(M')$. From $\Gamma \vdash^{\mathrm{CDZ}}_{\cap\top} P : \varphi$ we get $\Gamma, z{:}\omega \vdash^{\mathrm{CDZ}}_{\cap\top} P' : \varphi$ by Theorem 16.1.10(iii), since $\varphi =_{\mathrm{CDZ}} \omega{\rightarrow}\varphi$. This implies $M' \in [\varphi]^{\mathrm{CDZ}}_{\Gamma,z:\omega}$. We may assume that $z \notin \mathrm{dom}(\Gamma')$. Then also $M' \in [\varphi]^{\mathrm{CDZ}}_{\Gamma',z:\omega}$. Therefore

$$\begin{array}{rcll} MN & \twoheadrightarrow_{\boldsymbol{\beta}} & (\lambda z.M')N & \\ & \rightarrow_{\boldsymbol{\beta}} & M'[z := N] & \\ & \in & [\varphi]^{\mathrm{CDZ}}_{\Gamma'}, & \text{by (i)},\\ & = & [\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma'}. & \end{array}$$

Case $P \equiv x\vec{P}$. Notice that $\Gamma \vdash^{\mathrm{CDZ}}_{\cap\top} P : \varphi$ implies $\Gamma \vdash^{\mathrm{CDZ}}_{\cap\top} P : \omega{\rightarrow}\varphi$, since $\varphi =_{\mathrm{CDZ}} \omega{\rightarrow}\varphi$. By Lemma 18.3.12(ii) $[\![\omega]\!]^{\mathrm{CDZ}}_{\Gamma'} \subseteq [\omega]^{\mathrm{CDZ}}_{\Gamma'}$, hence there is $P' \in \mathcal{A}_{\mathrm{CDZ}}(N)$ such that $\Gamma' \vdash^{\mathrm{CDZ}}_{\cap\top} P' : \omega$. We get $\Gamma' \vdash^{\mathrm{CDZ}}_{\cap\top} PP' : \varphi$. As $PP' \in \mathcal{A}_{\mathrm{CDZ}}(MN)$ we conclude that $MN \in [\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma'}$.

Proof of $[\![\varphi{\rightarrow}\omega]\!]^{\mathrm{CDZ}}_{\Gamma} \subseteq [\![\omega]\!]^{\mathrm{CDZ}}_{\Gamma}$. We have $[\![\varphi{\rightarrow}\omega]\!]^{\mathrm{CDZ}}_{\Gamma} \subseteq [\varphi{\rightarrow}\omega]^{\mathrm{CDZ}}_{\Gamma}$ by Lemma 18.3.12(ii) and $[\varphi{\rightarrow}\omega]^{\mathrm{CDZ}}_{\Gamma} = [\omega]^{\mathrm{CDZ}}_{\Gamma}$, as $\varphi{\rightarrow}\omega =_{\mathrm{CDZ}} \omega$, using Definition 18.3.7. Moreover using Definition 18.3.10 it follows that

$$\begin{array}{rcl} [\![\varphi{\rightarrow}\omega]\!]^{\mathrm{CDZ}}_{\Gamma} & = & \{M \mid \forall \Gamma' \supseteq \Gamma, \forall N \in [\![\varphi]\!]^{\mathrm{CDZ}}_{\Gamma'}.MN \in [\![\omega]\!]^{\mathrm{CDZ}}_{\Gamma'}\}\\ & = & \{M \mid \forall \Gamma' \supseteq \Gamma, \forall N \in [\varphi]^{\mathrm{CDZ}}_{\Gamma'}.MN \in [\![\omega]\!]^{\mathrm{CDZ}}_{\Gamma'}\}\\ & \subseteq & \{M \mid \forall \Gamma' \supseteq \Gamma, \forall N, \vec{N} \in [\varphi]^{\mathrm{CDZ}}_{\Gamma'}.MN\vec{N} \in [\omega]^{\mathrm{CDZ}}_{\Gamma'}\} \end{array}$$

From $[\![\varphi{\to}\omega]\!]_\Gamma^{\rm CDZ} \subseteq [\omega]_\Gamma^{\rm CDZ}$ and

$$[\![\varphi{\to}\omega]\!]_\Gamma^{\rm CDZ} \subseteq \{M \mid \forall\Gamma' \supseteq \Gamma, \forall N, \vec{N} \in [\varphi]_{\Gamma'}^{\rm CDZ}.MN\vec{N} \in [\omega]_{\Gamma'}^{\rm CDZ}\}$$

we can conclude

$$[\![\varphi{\to}\omega]\!]_\Gamma^{\rm CDZ} \subseteq \{M \mid \forall\Gamma' \supseteq \Gamma, \vec{N} \in [\varphi]_{\Gamma'}^{\rm CDZ}.M\vec{N} \in [\omega]_{\Gamma'}^{\rm CDZ}\} = [\![\omega]\!]_\Gamma^{\rm CDZ}.$$

Proof of $[\![\omega]\!]_\Gamma^{\rm CDZ} \subseteq [\![\varphi{\to}\omega]\!]_\Gamma^{\rm CDZ}$. Again using Definition 18.3.10 one has

$$\begin{array}{rcl} M \in [\![\omega]\!]_\Gamma^{\rm CDZ} & \Rightarrow & \forall\Gamma' \supseteq \Gamma, \forall N, \vec{N} \in [\varphi]_{\Gamma'}^{\rm CDZ}.MN\vec{N} \in [\omega]_{\Gamma'}^{\rm CDZ} \\ & \Rightarrow & \forall\Gamma' \supseteq \Gamma, \forall N \in [\varphi]_{\Gamma'}^{\rm CDZ}.MN \in [\![\omega]\!]_{\Gamma'}^{\rm CDZ} \\ & \Rightarrow & M \in [\![\varphi{\to}\omega]\!]_\Gamma^{\rm CDZ}. \end{array}$$

Proof of $[\![(\varphi{\to}\varphi)\cap^\top(\omega{\to}\omega)]\!]_\Gamma^{\rm HR} \subseteq [\![\varphi]\!]_\Gamma^{\rm HR}$. By Lemma 18.3.12(ii) one has

$$\begin{array}{rcl} [\![(\varphi{\to}\varphi)\cap^\top(\omega{\to}\omega)]\!]_\Gamma^{\rm HR} & \subseteq & [(\varphi{\to}\varphi)\cap^\top(\omega{\to}\omega)]_\Gamma^{\rm HR} \\ & = & [\varphi]_\Gamma^{\rm HR} \\ & = & [\![\varphi]\!]_\Gamma^{\rm HR} \end{array}$$

by Definition 18.3.7, $(\varphi{\to}\varphi)\cap^\top(\omega{\to}\omega) = \varphi$ and Definition 18.3.10.

Proof of $[\![\varphi]\!]_\Gamma^{\rm HR} \subseteq [\![(\varphi{\to}\varphi)\cap^\top(\omega{\to}\omega)]\!]_\Gamma^{\rm HR}$. Let $\Gamma' \supseteq \Gamma$.

$$\begin{array}{l} M \in [\![\varphi]\!]_\Gamma^{\rm HR} \Rightarrow M \in [\varphi]_\Gamma^{\rm HR} \\ \qquad\qquad \Rightarrow \exists P \in \mathcal{A}_{\rm HR}(M)\ \Gamma \vdash_{\cap^\top}^{\rm HR} P : \varphi, \text{ by Definition 18.3.7.} \end{array} \tag{18.3}$$

$$\begin{array}{l} N \in [\![\varphi]\!]_{\Gamma'}^{\rm HR} \Rightarrow N \in [\varphi]_{\Gamma'}^{\rm HR} \\ \qquad\qquad \Rightarrow \exists P' \in \mathcal{A}_{\rm HR}(N)\ \Gamma' \vdash_{\cap^\top}^{\rm HR} P' : \varphi, \text{ by Definition 18.3.7.} \end{array} \tag{18.4}$$

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

$$\begin{array}{rcll} \text{(18.3) and (18.4)} & \Rightarrow & \Gamma' \vdash^{\rm HR} \hat{P} : \varphi, & \text{by (Ax-}\Phi\text{-HR), } (\leq_{\mathcal{T}}\ ),\ ({\to}\text{E}) \\ & \Rightarrow & MN \in [\varphi]_{\Gamma'}^{\rm HR}, & \text{since } \hat{P} \in \mathcal{A}_{\rm HR}(MN) \\ & \Rightarrow & MN \in [\![\varphi]\!]_{\Gamma'}^{\rm HR} & \\ & \Rightarrow & M \in [\![\varphi \to \varphi]\!]_\Gamma^{\rm HR} & \end{array}$$

$$\begin{array}{l} N \in [\![\omega]\!]_{\Gamma'}^{\rm HR} \Rightarrow N \in [\omega]_{\Gamma'}^{\rm HR} \\ \qquad\qquad \Rightarrow \exists P' \in \mathcal{A}_{\rm HR}(N)\ \Gamma' \vdash_{\cap^\top}^{\rm HR} P' : \omega, \text{ by Definition 18.3.7.} \end{array} \tag{18.5}$$

$$\begin{array}{l} \vec{N} \in [\![\varphi]\!]_{\Gamma'}^{\rm HR} \Rightarrow \vec{N} \in [\varphi]_{\Gamma'}^{\rm HR} \\ \qquad\qquad \Rightarrow \exists \vec{P} \in \mathcal{A}_{\rm HR}(\vec{N})\ \Gamma' \vdash_{\cap^\top}^{\rm HR} \vec{P} : \vec{\varphi}, \qquad \text{by Definition 18.3.7.} \end{array} \tag{18.6}$$

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

$$\begin{array}{rcll}
(18.3), (18.5) \text{ and } (18.6) & \Rightarrow & \Gamma' \vdash^{\mathrm{HR}} \hat{P} : \omega, & \text{by (Ax-}\Phi\text{-HR)}, (\leq_{\mathcal{T}}), (\to\mathrm{E}) \\
& \Rightarrow & M N \vec{N} \in [\omega]^{\mathrm{HR}}_{\Gamma'} & \text{since } \hat{P} \in \mathcal{A}_{\mathrm{HR}}(M N \vec{N}) \\
& \Rightarrow & M N \in [\![\omega]\!]^{\mathrm{HR}}_{\Gamma'} & \\
& \Rightarrow & M \in [\![\omega \to \omega]\!]^{\mathrm{HR}}_{\Gamma}. &
\end{array}$$

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

$$\begin{array}{rcll}
z \in [\omega]^{\mathrm{Park}}_{\{z:\omega\}} & \Rightarrow & z \in [\![\omega]\!]^{\mathrm{Park}}_{\{z:\omega\}} & \\
& \Rightarrow & M z \in [\![\omega]\!]^{\mathrm{Park}}_{\Gamma'} & \\
& \Rightarrow & M z \in [\omega]^{\mathrm{Park}}_{\Gamma'} & \\
& \Rightarrow & M \in [\omega]^{\mathrm{Park}}_{\Gamma}, & \text{by Lemma 18.3.9 and } (\omega \to \omega) \leq_{\mathrm{Park}} \omega, \\
& \Rightarrow & M \in [\![\omega]\!]^{\mathrm{Park}}_{\Gamma}. &
\end{array}$$

Proof of $[\![\omega]\!]^{\mathrm{Park}}_{\Gamma} \subseteq [\![\omega{\to}\omega]\!]^{\mathrm{Park}}_{\Gamma}$. Let $\Gamma' \supseteq \Gamma$. Then we have

$$\begin{array}{rl}
M \in [\![\omega]\!]^{\mathrm{Park}}_{\Gamma} & \Rightarrow M \in [\omega]^{\mathrm{Park}}_{\Gamma} \\
& \Rightarrow \exists P \in \mathcal{A}_{\mathrm{Park}}(M)\ \Gamma \vdash^{\mathrm{Park}}_{\cap^{\top}} P : \omega, \text{ by Definition 18.3.7.}
\end{array} \tag{18.7}$$

$$\begin{array}{rl}
N \in [\![\omega]\!]^{\mathrm{Park}}_{\Gamma'} & \Rightarrow N \in [\omega]^{\mathrm{Park}}_{\Gamma'} \\
& \Rightarrow \exists P' \in \mathcal{A}_{\mathrm{Park}}(N)\ \Gamma' \vdash^{\mathrm{Park}}_{\cap^{\top}} P' : \omega, \text{ by Definition 18.3.7.}
\end{array} \tag{18.8}$$

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

$$\begin{array}{rcll}
(18.7) \text{ and } (18.8) & \Rightarrow & \Gamma' \vdash^{\mathrm{Park}}_{\cap^{\top}} \hat{P} : \omega, & \text{by (Ax-}\Phi\text{)}, (\leq_{\mathrm{Park}}), (\to\mathrm{E}) \\
& \Rightarrow & M N \in [\omega]^{\mathrm{Park}}_{\Gamma'}, & \text{since } \hat{P} \in \mathcal{A}_{\mathrm{Park}}(M N) \\
& \Rightarrow & M N \in [\![\omega]\!]^{\mathrm{Park}}_{\Gamma'} & \\
& \Rightarrow & M \in [\![\omega \to \omega]\!]^{\mathrm{Park}}_{\Gamma}. \blacksquare &
\end{array}$$

18.3.14. Definition (Semantic Satisfiability). Let ρ be a mapping from term variables to terms and write $[\![M]\!]_{\rho} = M[\vec{x} := \rho(\vec{x})]$, where $\vec{x} = \mathrm{FV}(M)$.

$$\begin{array}{rrcl}
\text{(i)} & \mathcal{S}, \rho, \Gamma \models M : A & \Leftrightarrow & [\![M]\!]_{\rho} \in [\![A]\!]_{\Gamma}. \\
\text{(ii)} & \mathcal{S}, \rho, \Gamma' \models \Gamma & \Leftrightarrow & \mathcal{S}, \rho, \Gamma' \models x : B, \text{ for all } (x{:}B) \in \Gamma; \\
\text{(iii)} & \Gamma \models^{\mathcal{S}} M : A & \Leftrightarrow & \mathcal{S}, \rho, \Gamma' \models \Gamma \Rightarrow \mathcal{S}, \rho, \Gamma' \models M : A, \text{ for all } \rho, \Gamma'.
\end{array}$$

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

18.3.15. Lemma. *Let* $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Park}, \mathrm{HR}\}$. *Then*

$$\Gamma \vdash^{\mathcal{S}}_{\cap^{\top}} M : A \Rightarrow \Gamma \models^{\mathcal{S}} M : A.$$

Proof. The proof is by induction on the derivation of $\Gamma \vdash^{\mathcal{T}}_{\cap\top} M : A$.

Cases (Ax), (Ax-$\top$). Immediate.

Cases ($\to$E), ($\cap$I). By induction.

Case ($\leq$). By Lemma 18.3.13(ii).

Case ($\to$I). Suppose $M \equiv \lambda y.R$, $A \equiv B{\to}C$ and $\Gamma, y : B \vdash^{\mathcal{S}}_{\cap\top} R : C$.

Subcase $\mathcal{S} \neq$ AO or $C{\neq}_{\mathrm{AO}}\top$. Suppose $\mathcal{S}, \rho, \Gamma' \models \Gamma$ in order to show $[\![\lambda y.R]\!]_\rho \in [\![B{\to}C]\!]^{\mathcal{S}}_{\Gamma'}$. Let $\Gamma'' \Supseteq \Gamma'$ and $T \in [\![B]\!]^{\mathcal{S}}_{\Gamma''}$. Then by the induction hypothesis $[\![R]\!]_{\rho[y:=T]} \in [\![C]\!]^{\mathcal{S}}_{\Gamma''}$. We may assume $y \notin \rho(x)$ for all $x \in \mathrm{dom}(\Gamma)$. Then by Proposition 18.3.11 $[\![\lambda y.R]\!]_\rho T \to_{\boldsymbol{\beta}} [\![R]\!]_{\rho[y:=T]}$ and hence $[\![\lambda y.R]\!]_\rho T \in [\![C]\!]^{\mathcal{S}}_{\Gamma''}$. So $[\![\lambda y.R]\!]_\rho \in [\![B{\to}C]\!]^{\mathcal{S}}_{\Gamma'}$.

Subcase $\mathcal{S} =$ AO and $C =_{\mathrm{AO}} \top$. The result follows easily observing that $\lambda x.\bot \in \mathcal{A}_{\mathrm{AO}}([\![\lambda y.R]\!]_\rho)$ for all ρ and we can derive $\vdash^{\mathrm{AO}}_{\cap\top} \lambda x.\bot : B{\to}C$ using (Ax-$\top$), ($\to$I) and ($\leq_{\mathrm{AO}}$). We conclude $[\![M]\!]_\rho \in [A]^{\mathcal{S}}_\Gamma$ which implies $[\![M]\!]_\rho \in [\![A]\!]^{\mathcal{S}}_\Gamma$ by Definition 18.3.10. ∎

Now we can prove the converse of Proposition 18.3.8.

18.3.16. Proposition. *Let* $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Park}, \mathrm{HR}\}$. *Then*

$$\Gamma \vdash^{\mathcal{S}}_{\cap\top} M : A \Rightarrow M \in [A]^{\mathcal{S}}_\Gamma.$$

Proof. Let $\rho_0(x) = x$. By Lemma 18.3.12(i) $\mathcal{S}, \rho_0, \Gamma \models \Gamma$. Then $\Gamma \vdash^{\mathcal{T}}_{\cap\top} M : A$ implies $M = [\![M]\!]_{\rho_0} \in [\![A]\!]^{\mathcal{S}}_\Gamma$ by Lemma 18.3.15. So we conclude $M \in [A]^{\mathcal{S}}_\Gamma$ by Lemma 18.3.12(ii). ∎

18.3.17. Theorem (Approximation Theorem). *Let* $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Park}, \mathrm{HR}\}$. *Then*

$$\Gamma \vdash^{\mathcal{S}}_{\cap\top} M : A \;\Leftrightarrow\; \exists P \in \mathcal{A}_{\mathcal{S}}(M).\Gamma \vdash^{\mathcal{S}}_{\cap\top} P : A.$$

Proof. By Propositions 18.3.8 and 18.3.16. ∎

18.3.18. Corollary. *Let* $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}, \mathrm{AO}, \mathrm{Park}, \mathrm{HR}\}$. *Let* M *be an untyped lambda term and write* $\Gamma_\rho = \{x{:}B \mid B \in \rho(x)\}$. *Then*

$$[\![M]\!]^{\mathcal{F}^{\mathcal{S}}}_\rho = \{A \mid \Gamma_\rho \vdash^{\mathcal{S}}_{\cap\top} P : A \textit{ for some } P \in \mathcal{A}_{\mathcal{S}}(M) \textit{ and some } \Gamma \models \rho\}.$$

Proof. By Theorem 18.2.8 Exercise ?? (for $\mathcal{S} =$ AO) and the Approximation Theorem. ∎

Another way of writing this is

$$\begin{aligned}
[\![M]\!]^{\mathcal{F}^{\mathcal{S}}}_\rho &= \bigcup_{P \in \mathcal{A}_{\mathcal{S}}(M)} [\![P]\!]^{\mathcal{F}^{\mathcal{S}}}_\rho \\
&= \bigcup_{P \in \mathcal{A}_{\mathcal{S}}(M)} \{A \mid \Gamma \vdash^{\mathcal{S}}_{\cap\top} P : A \text{ for some } \Gamma \models \rho\}.
\end{aligned}$$

This gives the motivation for the name 'Approximation Theorem'. Theorem 18.3.17 was first proved for $\mathcal{S} =$ BCD in Barendregt et al. [1983], for $\mathcal{S} =$ Scott in Ronchi Della Rocca [1988], for $\mathcal{S} =$ CDZ in Coppo et al. [1987], for $\mathcal{S} =$ AO in Abramsky and Ong [1993], for $\mathcal{S} =$ Park and $\mathcal{S} =$ HR in Honsell and Ronchi Della Rocca [1992].

Some applications of the Approximation Theorem

As discussed in Section 18.2 type theories give rise in a natural way to *filter λ-models*. Properties of $\mathcal{F}^{\mathcal{S}}$ with $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{AO}, \mathrm{BCD}, \mathrm{Park}, \mathrm{HR}\}$ can be easily derived using Theorem 18.3.17. For instance, one can check the following.

- The models $\mathcal{F}^{\mathrm{Scott}}, \mathcal{F}^{\mathrm{CDZ}}, \mathcal{F}^{\mathrm{DHM}}, \mathcal{F}^{\mathrm{BCD}}$ are sensible (see Barendregt [1984]).
- The top element in $\mathcal{F}^{\mathrm{AO}}$ is the interpretation of the terms of order ∞.
- The model $\mathcal{F}^{\mathrm{Park}}$ characterizes the terms reducible to closed terms.
- The model $\mathcal{F}^{\mathrm{HR}}$ characterizes the terms reducible to λI-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. Barendregt et al. [1983], Coppo et al. [1987], Ronchi Della Rocca [1988], Abramsky and Ong [1993], Honsell and Ronchi Della Rocca [1992].

18.3.19. Theorem. *The models $\mathcal{F}^{\mathcal{S}}$ with $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{BCD}\}$ are sensible, i.e. for all unsolvable terms M, N one has*

$$[\![M]\!]_{\rho}^{\mathcal{F}^{\mathcal{S}}} = [\![N]\!]_{\rho}^{\mathcal{F}^{\mathcal{S}}}.$$

Proof. It follows immediately from Corollary 18.3.18 of the Approximation Theorem, the fact that $\perp$ is the only approximant of an unsolvable term for the mapping $\boxtimes$, and Proposition 18.3.5(i). ■

Let us recall the definition of term of order ∞, see Longo [1983].

18.3.20. Definition. An untyped lambda term M is *of order ∞* iff

$$\forall n \exists M'.M \twoheadrightarrow_{\boldsymbol{\beta}} \lambda x_1 \ldots \lambda x_n.M'.$$

18.3.21. Theorem. *Let M be an untyped lambda term. Then the following are equivalent.*

(i) *M is of order ∞.*
(ii) *$\vdash^{\mathrm{AO}}_{\cap\top} M : A$ for all types $A \in \mathbb{T}^{\mathrm{AO}}$.*
(iii) *$[\![M]\!]_{\rho}^{\mathcal{F}^{\mathrm{AO}}} = \top = \mathbb{T}^{\mathrm{AO}} \in \mathcal{F}^{\mathrm{AO}}$ for all valuations ρ.*

Proof. Write $\vdash, \leq$ for $\vdash^{\mathrm{AO}}_{\cap\top}, \leq_{\mathrm{AO}}$, respectively. ((i)$\Leftrightarrow$(ii)). It is easy to check by structural induction on types (see Exercise 16.3.1) that

$$\forall A \in \mathbb{T}^{\mathrm{AO}} \exists n \in \mathbb{N}.(\top^n \to \top) \leq A.$$

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

$$\vdash P : (\top^n \to \top) \;\Leftrightarrow\; P \equiv \lambda x_1 \ldots \lambda x_n.P' \text{ for some } P'.$$

($\Leftarrow$) By axiom ($\top$-universal) and rule ($\to$I). ($\Rightarrow$) Assume towards a contradiction that $P \equiv \lambda x_1 \ldots \lambda x_m.P'$ for $m < n$ and P' is of the form $\bot$ or $x\vec{P}$. Then by Theorem 16.1.10(iii)

$$\vdash P : (\top^n \to \top) \;\Rightarrow\; \{x_1{:}\top, \ldots, x_m{:}\top\} \vdash P' : (\top^{n-m} \to \top).$$

But this latter judgment can neither be derived if $P' \equiv \bot$, by Proposition 18.3.5(i) and Lemma 15.1.17, nor if $P' \equiv x\vec{P}$, by Theorem 16.1.1(i) and (ii) and Lemma 15.1.17.

((ii)$\Leftrightarrow$(iii)). Suppose $\vdash M : A$ for all A. Then by Theorem 18.2.8

$$[\![M]\!]^{\mathcal{F}^{\mathrm{AO}}}_{\rho} = \{A \mid \Gamma \vdash M : A \text{ for some } \Gamma \models \rho\} = \mathbb{T}^{\mathrm{AO}},$$

for all ρ. This is the top element in $\mathcal{F}^{\mathrm{AO}}$. Conversely, if $[\![M]\!]^{\mathcal{F}^{\mathrm{AO}}}_{\rho} = \top = \mathbb{T}^{\mathrm{AO}}$, for all ρ, then take $\rho_0(x) = \uparrow\!\top$. Then $\Gamma_{\rho_0} = \{x{:}\top \mid x \in \mathrm{Var}\}$. Hence

$$\begin{array}{rcll} \mathbb{T}^{\mathrm{AO}} & = & [\![M]\!]^{\mathcal{F}^{\mathrm{AO}}}_{\rho_0} & \\ & = & \{A \mid \Gamma \vdash M : A \text{ for some } \Gamma \models \rho_0\}, & \text{by Theorem 18.2.8,} \\ & = & \{A \mid \vdash M : A\}, & \text{since the assumptions } x{:}\top \text{ are superfluous.} \end{array}$$

Therefore $\vdash M : A$ for all $A \in \mathbb{T}^{\mathrm{AO}}$. ∎

18.3.22. THEOREM. *A term M reduces to a closed term iff $\vdash^{\mathrm{Park}}_{\cap^\top} M : \omega$.*

PROOF. By the Approximation Theorem it suffices to check that if $P \in \Lambda\Phi$ is an anf and $\mathcal{V}$ is a finite set of term variables:

$$\{x{:}\omega \mid x \in \mathcal{V}\} \vdash^{\mathrm{Park}}_{\cap^\top} P : \omega \text{ iff } FV(P) \subseteq \mathcal{V}.$$

($\Leftarrow$) By an easy induction on P, using that $\omega = \omega{\to}\omega$.

($\Rightarrow$) By induction on P. Lemmas 15.1.16 shows $\vec{B}{\to}\omega \neq \top$.

Case $P \equiv \lambda y.P'$. By Theorem 16.1.10(iii) and the induction hypothesis for P'.

Case $P \equiv y\vec{P}$. By Theorem 16.1.10(ii) we have $\Gamma \vdash^{\mathrm{Park}}_{\cap^\top} y : \vec{B}{\to}\omega, \Gamma \vdash^{\mathrm{Park}}_{\cap^\top} \vec{P} : \vec{B}$. Hence by Theorem 16.1.10(i) one has $y \in \mathcal{V}$ and $\omega \leq \vec{B}{\to}\omega$. By $\boldsymbol{\beta}$-soundness and $\omega = \omega{\to}\omega$ we get $B_i \leq \omega$. Thus $\Gamma \vdash^{\mathrm{Park}}_{\cap^\top} P_i : \omega$ and hence $\mathrm{FV}(P_i) \subseteq \mathcal{V}$, by the induction hypothesis.

Case $P \equiv \Phi\vec{P}$. Similar to the previous case. ∎

Lastly we work out the characterization of terms reducible to λI-terms. We define the set of terms we want to characterize and the set obtained by adding the constant Φ to it.

18.3.23. DEFINITION. (i) The set $\Lambda I\beta$ of $\lambda I\beta$-terms is the set of lambda-terms that reduce to a λI-term.

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

In the following key lemma we show that each approximate normal form $P \in \Lambda\mathsf{I}\beta\Phi$ is typable with φ from the HR-basis all whose predicates are φ. To properly deal 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 HR-basis containing $x{:}\omega$ and all whose other predicates are φ. This is useful since $\varphi = (\varphi{\to}\varphi)\cap^{\top}(\omega{\to}\omega)$.

18.3.24. Lemma. *Let $P \in \Lambda\Phi$ be an anf, and $\Gamma_\varphi = \{y{:}\varphi \mid y \in \mathrm{Var}\}$. Then*

(i) $P \in \Lambda\mathsf{I}\beta\Phi \;\Leftrightarrow\; \Gamma_\varphi \vdash^{\mathrm{HR}}_{\cap^{\top}} P : \varphi.$

(ii) *Let $P \in \Lambda\mathsf{I}\beta\Phi$. Then $x \in P \;\Leftrightarrow\; \Gamma_\varphi \uplus \{x{:}\omega\} \vdash^{\mathrm{HR}}_{\cap^{\top}} P : \omega.$*

Proof. (i) and (ii) can be simultaneously proved by induction on P. Write $\vdash, \leq$ for $\vdash^{\mathrm{HR}}_{\cap^{\top}}, \leq_{\mathrm{HR}}$, respectively.

Case $P \equiv y$ is trivial.

Case $P \equiv \lambda z.P'$. ($\Rightarrow$) 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 $\Gamma_\varphi \vdash P' : \varphi$ and $\Gamma_\varphi \uplus \{z{:}\omega\} \vdash P' : \omega$, so by rules ($\to$I), ($\cap$I) and ($\leq$) we get $\Gamma_\varphi \vdash P : \varphi$. If $x \in P$, then $x \in P'$ and by induction $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P' : \omega$, so by rules ($\to$I) and ($\leq$) we get $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P : \omega$. For ($\Leftarrow$) $\Gamma_\varphi \vdash P : \varphi$ implies $\Gamma_\varphi \vdash P' : \varphi$ and $\Gamma_\varphi \uplus \{z{:}\omega\} \vdash P' : \omega$ by Theorem 16.1.10(iii) since $\varphi = (\varphi{\to}\varphi)\cap^{\top}(\omega{\to}\omega)$ and HR is β-sound. By induction $P' \in \Lambda\mathsf{I}\beta\Phi$ and $z \in P'$ so we get $P \in \Lambda\mathsf{I}\beta\Phi$. Moreover $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P : \omega$ implies $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P' : \omega$ so we get by induction $x \in P'$, i.e. $x \in P$.

Case $P \equiv \chi P' \vec{P}$ where $\chi \in \{\Phi, z\}$. For ($\Rightarrow$) 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 $\Gamma_\varphi \vdash P' : \varphi$ and $\Gamma_\varphi \vdash P_i : \varphi$ for all i, so by axiom either (Ax) or (Ax-Φ-HR) and rules ($\to$E), ($\leq$) we get $\Gamma_\varphi \vdash P : \varphi$. If $x \in P$, then either $\chi \equiv x$ or $x \in P'\vec{P}$. In the first case $x{:}\omega \vdash x : \omega$ so by rules (*weakening*), ($\to$E), ($\leq$) we get $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P : \omega$. In the second case by induction either $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P' : \omega$ or $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P_i : \omega$ for some i, so by axiom either (Ax) or (Ax-Φ-HR) and rules ($\to$E), ($\leq$) we get $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P : \omega$.

($\Leftarrow$) Notice that $\vec{B}{\to}\varphi \neq \top$, by Lemma 15.1.16(iii) and (iv).

Subcase ($\Leftarrow$(i)). By Theorem 16.1.10(ii) we get

$$\Gamma_\varphi \vdash \chi : B'{\to}\vec{B}{\to}\varphi \;\&\; \Gamma_\varphi \vdash P' : B' \;\&\; \Gamma_\varphi \vdash \vec{P} : \vec{B}.$$

So $\varphi \leq B'{\to}\vec{B}{\to}\varphi$, by Theorem 16.1.10(i) and Proposition 18.3.5(iii). From $\omega \leq \varphi$ and $\omega = (\vec{\varphi}{\to}\omega)$ we get $\vec{\varphi}{\to}\omega \leq B'{\to}\vec{B}{\to}\varphi$. Hence by β-soundness $B' \leq \varphi$ and $B_i \leq \varphi$, for some i. Therefore $\Gamma_\varphi \vdash P' : \varphi$ and $\Gamma_\varphi \vdash P_i : \varphi$. So $P' \in \Lambda\mathsf{I}\beta\Phi$ and $P_i \in \Lambda\mathsf{I}\beta\Phi$, by the induction hypothesis and hence $P \in \Lambda\mathsf{I}\beta\Phi$.

Subcase ($\Leftarrow$(ii)). Suppose $P \equiv \chi P'\vec{P} \in \Lambda\mathsf{I}\beta\Phi$ and $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P : \omega$ in order to prove that $x \in P$. This clearly holds if $\chi = x$, so let $\chi = \Phi$. Similar to the case ($\Leftarrow$(i)) above we get

$$\begin{array}{rcl} \Gamma_\varphi \uplus \{x{:}\omega\} & \vdash & \Phi : B'{\to}\vec{B}{\to}\omega, \\ \Gamma_\varphi \uplus \{x{:}\omega\} & \vdash & P' : B', \\ \Gamma_\varphi \uplus \{x{:}\omega\} & \vdash & \vec{P} : \vec{B}, \\ \varphi & \leq & B'{\to}\vec{B}{\to}\omega, \\ B' & \leq & \varphi, \\ B_i & \leq & \varphi. \end{array}$$

Now $(\varphi{\to}\varphi) \cap (\omega{\to}\omega) \leq B'{\to}\vec{B}{\to}\omega$. Hence by $\boldsymbol{\beta}$-soundness we have

$$(B' \leq \varphi \;\&\; \varphi \leq \vec{B}{\to}\omega) \vee (B' \leq \omega \;\&\; \omega \leq \vec{B}{\to}\omega).$$

So from $\varphi \leq B'{\to}B{\to}\omega$ we get $B' \leq \omega$ or $\varphi \leq \vec{B}{\to}\omega$. Now $(\varphi{\to}\varphi) \cap (\omega{\to}\omega) \leq B_n{\to}\omega$ implies $B_n \leq \omega$, by $\boldsymbol{\beta}$-soundness and Proposition 15.1.20. So we get $B' \leq \omega$ or $B_i \leq \omega$, for some i. Hence $\Gamma_\varphi \uplus \{x{:}\omega\} \vdash P' : \omega \vee \Gamma_\varphi \uplus \{x{:}\omega\} \vdash P_i : \omega$, for some i. Therefore $x \in P'$ or $x \in P_i$ for some i, by the induction hypothesis. ∎

18.3.25. Lemma. *If $\varphi \leq A_1 \to \ldots A_m \to \omega$, then there exists i such that $A_i = \omega$.*

Proof. By induction on m.
Case $m = 1$. $\varphi = (\varphi \to \varphi) \cap (\omega \to \omega) \leq A_1 \to \omega$. By $\boldsymbol{\beta}$-soundness, the only possible case is $A_1 \leq \omega$. Since ω is the least element, we have that $A_1 = \omega$.
Case $m > 1$. Similar, using induction. ∎

18.3.26. Lemma. *Let Γ such that $A = \varphi$ for all $x : A$ in Γ, then there is no anf P such that $\Gamma \vdash P : \omega$.*

Proof. Suppose there exists P towards a contradiction. By induction on P using the Inversion Lemmas.
If $P = \lambda z.P'$, then $\Gamma \vdash P : \omega$ implies $\Gamma, z : \varphi \vdash P' : \omega$ and induction applies.
If $P = P_0 P_1 ... P_m$ where P_0 is either a variable z or Φ, then

$$\Gamma \vdash P_0 : A_1 \to \ldots A_m \to \omega \tag{18.9}$$

and $\Gamma \vdash P_i : A_i$ for some A_i. (18.9) implies $\varphi \leq A_1 \to \ldots A_m \to \omega$. By Lemma 18.3.25, there exists i such that $A_i = \omega$ and then $\Gamma \vdash P_i : \omega$. By induction hypothesis, this is impossible.

We define $\Gamma^P_\varphi = \{x{:}\varphi \mid x \in FV(P)\}$.

18.3.27. Corollary. *If $\Gamma^P_\varphi \uplus \{x{:}\omega\} \vdash P : \omega$, then $x \in FV(P)$.*

18.3.28. Lemma. *If $\Gamma^P_\varphi \vdash P : \varphi$, then P is a λI-term.*

PROOF. By induction on P.

Case $P \equiv y$. Trivial.

Case $P \equiv \lambda z.P'$. By Inversion Theorem 16.1.10(iii), $\Gamma_\varphi^{P'} \vdash P' : \varphi$ and $\Gamma_\varphi^{P'} \uplus z : \omega \vdash P' : \omega$. By induction hypothesis, we have that P' is a λI-term. It remains to prove that $z \in FV(P')$. Suppose that $z \notin FV(P')$. Then we could remove it from the context and get $\Gamma_\varphi^{P'} \vdash P' : \omega$. But this contradicts Lemma 18.3.26.

Case $P \equiv P_0 P_1 \dots P_n$ where $P_0 = x$ or $P_0 = \Phi$. By Inversion Theorem 16.1.10 (ii), $\Gamma_\varphi^P \vdash P_0 : A_1 \to \dots \to A_n \to \varphi$ and $\Gamma_\varphi^P \vdash P_i : A_i$ for all $i > 0$. By Inversion Theorem 16.1.10(i) for $P_0 = x$ or Proposition 18.3.5(iii) for $P_0 = \Phi$, we get $\varphi \le A_1 \to \dots \to A_n \to \varphi$. Since $\omega \le \varphi$ and $\omega = \varphi \to \omega$, we get that $\varphi \to \dots \to \varphi \to \omega \le A_1 \to \dots \to A_n \to \varphi$. By β-soundness, $A_i \le \varphi$. Hence, by rules (*strengthening*) and $(\le)$, $\Gamma_\varphi^{P_i} \vdash P_i : \varphi$ for all $i > 0$. By induction hypothesis, all P_i's are λI-terms. ■

18.3.29. LEMMA. *Let $P \in \Lambda\Phi$ be an anf.*

(i) *If P is a λI-term, then $\Gamma_\varphi^P \vdash P : \varphi$.*

(ii) *If P is a λI-term and $x \in FV(P)$, then $\Gamma_\varphi^P \uplus \{x : \omega\} \vdash P : \omega$.*

PROOF. Simultaneously by induction on P.

(i) Case $P \equiv y$. Trivial.

Case $P \equiv \lambda z.P'$. If P is a λI-term, so is P' and $z \in FV(P')$. By induction hypothesis (i), we have that $\Gamma_\varphi^{P'} \vdash P' : \varphi$. By induction hypothesis (ii), we have that $\Gamma_\varphi^{P'} \uplus \{z : \omega\} \vdash P' : \omega$. Since $\varphi \cap \omega = \omega$ and $\varphi = (\varphi \to \varphi) \cap (\omega \to \omega)$ and by rules $(\to\mathrm{I})$,$(\cap\mathrm{I})$, $(\le)$ we get $\Gamma_\varphi^P \vdash P : \varphi$.

Case $P = P'P''$. By induction hypothesis (i), we have that $\Gamma_\varphi^{P'} \vdash P' : \varphi \le \varphi \to \varphi$ and $\Gamma_\varphi^{P''} \vdash P'' : \varphi$. By rules (*weakening*), $(\le)$ and $(\to\mathrm{E})$, we get that $\Gamma_\varphi^P \vdash P : \varphi$.

(ii) Case $P \equiv x$. Trivial.

Case $P \equiv \lambda z.P'$. Since $x \in FV(P)$, $x \in FV(P')$. By induction hypothesis (ii), $\Gamma_\varphi^{P'} \uplus \{x : \omega\} \vdash P' : \omega$. By rules $(\to\mathrm{I})$, $(\le)$ with $\varphi \to \omega = \omega$ we get $\Gamma_\varphi^P \uplus \{x : \omega\} \vdash P : \omega$.

Case $P \equiv P'P''$. We have two subcases:

$x \in FV(P')$. By induction hypothesis (ii), $\Gamma_\varphi^{P'} \uplus \{x : \omega\} \vdash P' : \omega = \varphi \to \omega$. By induction hypothesis (i) $\Gamma_\varphi^{P''} \vdash P'' : \varphi$. Hence by (*weakening*) and $(\to\mathrm{E})$ we conclude $\Gamma_\varphi^P \uplus \{x : \omega\} \vdash P : \omega$.

$x \in FV(P'')$. By induction hypothesis (ii), $\Gamma_\varphi^{P''} \uplus \{x : \omega\} \vdash P'' : \omega$. By induction hypothesis (i) $\Gamma_\varphi^{P'} \vdash P' : \varphi \le \omega \to \omega$. Hence by (*weakening*) and $(\to\mathrm{E})$ we conclude $\Gamma_\varphi^P \uplus \{x : \omega\} \vdash P : \omega$. ■

Finally we prove the following result characterizing λI-terms in a type theoretic way.

18.3.30. THEOREM. *Let M be an untyped lambda term, Define for $\vec{x} = x_1, \dots, x_n$ the context $\Gamma_{\vec{x}} = \{x_1{:}\varphi, \dots, x_n{:}\varphi\}$. then*

$$M \text{ reduces to a } \lambda\text{I-term} \Leftrightarrow \Gamma_\varphi^M \vdash_{\cap^\top}^{\mathrm{HR}} M : \varphi.$$

PROOF. Easy from Theorem 18.3.17 and Lemma 18.3.29(i), observing that if $\boxdot(N)$ is a λI-term, then N is a λI-term too. ■

Theorem 18.3.30 was first proved in Honsell and Ronchi Della Rocca [1992] by purely semantic means.

18.4. Exercises

18.4.1. Check the following equalities:

$$\begin{array}{rcl} [\![\lambda x.y]\!]^{\mathrm{CDV}}_{\rho_0} & = & \emptyset; \\ [\![\lambda x.y]\!]^{\mathrm{HL}}_{\rho_1} & = & \uparrow\omega; \\ [\![\lambda x.y]\!]^{\mathrm{AO}}_{\rho_0} & = & \uparrow(\top\to\top). \\ [\![\lambda x.y]\!]_{\rho_0} & = & \uparrow\nu. \end{array}$$

where $\rho_0(y) = \uparrow\emptyset$ and $\rho_1(y) = \uparrow\omega$.

18.4.2. Using the Approximation Theorem shows that

- there is no type deducible for $\Delta\Delta$ in the system $\vdash^{\mathrm{CDV}}_{\cap}$;
- $\vdash^{\mathrm{BCD}}_{\cap\top} \Delta\Delta : A$ iff $A =_{\mathrm{BCD}} \top$;
- $\vdash^{\mathrm{Park}}_{\cap\top} \Delta\Delta : A$ iff $\omega \leq_{\mathrm{Park}} A$.

18.4.3. Using the Approximation Theorem shows that in the system $\lambda^{\mathrm{AO}}_{\cap\top}$ the set of types deducible for $\Delta\Delta$ is stricty included in the set of types deducible for $\mathsf{K}(\Delta\Delta)$.

18.4.4. Using the Approximation Theorem shows that in the system $\lambda^{\mathrm{Scott}}_{\cap\top}$ the set of types deducible for J and I coincide.

18.4.5. (i) Define $\mathsf{K}^\infty \equiv \mathsf{YK}$. This term is called the “ogre”. Find a type for it in the system $\lambda{\cap^\top}^{\mathrm{AO}}$.

(ii) Is there a type which cannot be derived for the “ogre” in $\lambda{\cap^\top}^{\mathrm{AO}}$? [Hint: use (i) and Exercise 15.5.7.]

18.4.6. Prove using the results of Exercise 18.4.5 that $[\![\mathsf{K}^\infty]\!]_\rho = \mathcal{F}^{\mathrm{AO}}$.

18.4.7. Define $\mathsf{t} : \mathbb{T}(\{\top,\varphi,\omega\})\to\mathbb{T}(\{\top,\varphi,\omega\})$ inductively:

$$\begin{array}{rcll} \mathsf{t}(\alpha) & = & \alpha, & \text{where } \alpha\in\{\top,\omega\}; \\ \mathsf{t}(\varphi) & = & \Omega; & \\ \mathsf{t}(A\to B) & = & A\to\mathsf{t}(B); & \\ \mathsf{t}(A\cap B) & = & \mathsf{t}(A)\cap\mathsf{t}(B). & \end{array}$$

The intersection type theory ♣ is axiomatized by rule $(\to)$ and axioms $(\to\cap)$, $(\top)$, $(\top\to)$, $(\omega\varphi)$, $(\varphi\to\omega)$, $(\omega\to\varphi)$, see Fig. 15.1.7, and (♣), (♣→), where

$$\begin{array}{ll} (\clubsuit) & A \leq \mathsf{t}(A). \\ (\clubsuit\!\to) & A\to B \leq \mathsf{t}(A)\to\mathsf{t}(B). \end{array}$$

If $\Gamma = \{x_1{:}A_1,\ldots,x_n{:}A_n\}$, then write $\mathsf{t}(\Gamma) = \{x_1{:}\mathsf{t}(A_1),\ldots,x_n{:}\mathsf{t}(A_n)\}$. Show the following.

(i) The map $\mathtt{t}$ is idempotent, i.e. $\mathtt{t}(\mathtt{t}(A)) = \mathtt{t}(A)$.

(ii) $A{\to}\mathtt{t}(B) =_{\clubsuit} \mathtt{t}(A){\to}\mathtt{t}(B)$.

(iii) $A \leq_{\clubsuit} B \Rightarrow \mathtt{t}(A) \leq_{\clubsuit} \mathtt{t}(B)$.

(iv) $\Gamma \vdash^{\clubsuit} M : A \Rightarrow \mathtt{t}(\Gamma) \vdash^{\clubsuit} M : \mathtt{t}(A)$.

(v) $\Gamma, \Gamma' \vdash^{\clubsuit} M : A \Rightarrow \; \Gamma, \mathtt{t}(\Gamma') \vdash^{\clubsuit} M : \mathtt{t}(A)$.

(vi) $\forall i \in I.\, \Gamma, x{:}A_i \vdash^{\clubsuit} M : B_i \;\&\; \bigcap_{i\in I}(A_i{\to}B_i) \leq_{\clubsuit} C{\to}D \;\Rightarrow$ $\Gamma, x{:}C \vdash^{\clubsuit} M : D$.

(vii) $\clubsuit$ is not β-sound. [Hint. $\varphi{\to}\omega \leq_{\clubsuit} \Omega{\to}\omega$.]

(viii) $\mathcal{F}^{\clubsuit}$ is a filter λ-model. [Hint. Modify Theorem 18.2.24, and Lemma 18.2.23(vi).]

(ix) The step function $\uparrow\varphi \Rightarrow \uparrow\omega$ is not representable in $\mathcal{F}^{\clubsuit}$.

Actually, $\mathcal{F}^{\clubsuit}$ is the inverse limit solution of the domain equation $\mathcal{D} \simeq [\mathcal{D}{\to}\mathcal{D}]$ taken in the category of $\clubsuit$-*lattices*, whose objects are ω-algebraic lattices $\mathcal{D}$ endowed with a finitary additive projection $\delta : \mathcal{D}{\to}\mathcal{D}$ and whose morphisms $f : (\mathcal{D},\delta){\to}(\mathcal{D}',\delta')$ are continuous functions such that $\delta' \circ f \sqsubseteq f \circ \delta$. See Alessi [1993], Alessi, Barbanera and Dezani-Ciancaglini [2004] for details.

18.4.8. Let $\mathcal{S}$ be a type structure such that $\nu \in \mathbb{A}^{\mathcal{S}}$ and $(A{\to}B) \leq_{\mathcal{S}} \nu$.

(i) Show $\uparrow\nu \cdot X = \emptyset$ for all $X \in \mathcal{F}^{\mathcal{S}}$ $\Leftrightarrow$ $\mathcal{S}$ is ν-sound.

(ii) Show $F_{\mathcal{S}}(G_{\mathcal{S}}(\bot \mapsto \bot)) = \bot \mapsto \bot$ $\Leftrightarrow$ $\mathcal{S}$ is ν-sound.

(iii) Show that if each type is equivalent to some constant $\psi \in \mathbb{A}^{\mathcal{S}}$, then

$$F_{\mathcal{S}}(\uparrow\nu) = \bigsqcup\{\psi \mapsto \psi' \mid \nu \leq_{\mathcal{S}} \psi{\to}\psi'\}.$$

(iv) Examine whether the condition in (iii) is necessary. **Comment:** Do you know the answer? No.

Chapter 19

Applications 16.10.2006:1032

The type assignment systems introduced in Section 15.2 will be put to use in the present chapter, where we show how to characterize a number of properties of λ-terms using intersection types. In particular, in Section 19.1 we characterize the soundness and completeness of the intersection type assignments for several variants of a natural semantics. In Section 19.2 we shall discuss several normalization properties. In Section 19.3 we see how to solve domain equations by building suitable filter structures. Finally in Section 19.5 it will be shown that given Γ, A one cannot predict inhabitation, i.e. the existence of an M such that (1).

19.1. Realizability interpretation of types

The natural set-theoretic semantics for type assignment in $\lambda_{\rightarrow}$ based on untyped λ-models is given in Scott [1975] where it was shown that

$$\Gamma \vdash_{\lambda_{\rightarrow}} M : A \;\Rightarrow\; \Gamma \models M : A.$$

Scott asked whether the converse (completeness) holds. In Barendregt et al. [1983] the notion of semantics was extended to intersection types and completeness was proved for $\lambda_{\cap}^{\mathrm{BCD}}$ via the corresponding filter model. Completeness for $\lambda_{\rightarrow}$ follows by a conservativity result. In Hindley [1983] an alternative proof of completeness for $\lambda_{\rightarrow}$ was given, using a term model. Variations of the semantics are presented in Dezani-Ciancaglini et al. [2003].

19.1.1. Definition. (Type Interpretation Domain)

(i) A *quasi λ-model* is a triple $\mathcal{D} = \langle \mathcal{D}, \cdot, [\![\ ]\!] \rangle$ with

(1) $\langle \mathcal{D}, \cdot \rangle$ is a free applicative structure;

(2) $[\![\ ]\!] : \Lambda \times \mathsf{Env}_{\mathcal{D}} \rightarrow \mathcal{D}$, where $\mathsf{Env}_{\mathcal{D}} = [Var \rightarrow \mathcal{D}]$, is a mapping (*interpretation function* for λ-terms) which satisfies the following properties

a. $[\![x]\!]_{\rho} = \rho(x)$;

b. $[\![MN]\!]_{\rho} = [\![M]\!]_{\rho} \cdot [\![N]\!]_{\rho}$;

c. $[\![\lambda x.M]\!]_{\rho} = [\![\lambda y.M[x := y]]\!]_{\rho}$ if $y \notin \mathrm{FV}(M)$;

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

(ii) For $X, Y \subseteq \mathcal{D}$ write $X \rightarrow Y = \{d \in \mathcal{D} \mid \forall e \in X. d \cdot e \in Y\}$.

Comment: we used ξ (macro ten) for type environment in Chapter 14

19.1.2. Definition (Type Interpretation). Let $\mathcal{D} = \langle \mathcal{D}, \cdot, [\![\;]\!] \rangle$ be a quasi λ-model and let $\mathcal{T}$ be an intersection type theory over the constants $\mathbb{A}^{\mathcal{T}}$. The *type interpretation* induced by the type environment $\xi : \mathbb{A}^{\mathcal{T}} \to \mathsf{P}(\mathcal{D})$ is defined by:

1. $[\![\top]\!]^{\mathcal{D}}_{\xi} = \mathcal{D}$;
2. $[\![\alpha]\!]^{\mathcal{D}}_{\xi} = \xi(\alpha)$ if $\alpha \in \mathbb{A}^{\mathcal{T}}$ and $\alpha \neq \top$;
3. $[\![A \to B]\!]^{\mathcal{D}}_{\xi} = [\![A]\!]^{\mathcal{D}}_{\xi} \to [\![B]\!]^{\mathcal{D}}_{\xi}$;
4. $[\![A \cap B]\!]^{\mathcal{D}}_{\xi} = [\![A]\!]^{\mathcal{D}}_{\xi} \cap [\![B]\!]^{\mathcal{D}}_{\xi}$,

where $X \to Y = \{d \in \mathcal{D} \mid \forall e \in X. d \cdot e \in Y\}$. ■

The above definition is the extension to intersection-types of the *simple semantics* for simple types of Scott [1975], generalized by allowing $\langle \mathcal{D}, \cdot \rangle$ to be just a quasi λ-terms instead of a λ-model.

In discussing *sound* type assignment systems we consider only quasi λ-models and type environments which are good (the notion of goodness will depend on the current intersection type theory) and which preserve with the inclusion relation between types in the following sense:

In order to prove soundness, we have to check that the interpretation preserves the typability rules. We already know that the interpretation preserves the typing rules for application and intersection. This is because $\cap$ is interpreted as intersection on sets and $\to$ is interpreted as the arrow induced by the application $\cdot$ on $\mathcal{D}$. The following definition is necessary to require that the interpretation preserves the remaining two typability rules: abstraction and subtyping.

19.1.3. Definition. Let a quasi λ-model $\mathcal{D} = \langle \mathcal{D}, \cdot, [\![\;]\!] \rangle$ and a type environment $\xi : \mathbb{A}^{\mathcal{T}} \to \mathsf{P}(\mathcal{D})$ be given.

(i) $(\mathcal{D}, \xi)$ are $\to$-*good* if for all $A, B \in \mathbb{T}^{\mathcal{T}}$ for all environments ρ, terms M and variables x

$$[\forall d \in [\![A]\!]_{\xi}.\ [\![M]\!]_{\rho[x:=d]} \in [\![B]\!]_{\xi}] \;\Rightarrow\; [\![\lambda x.M]\!]_{\rho} \in [\![A]\!]_{\xi} \to [\![B]\!]_{\xi};$$

(ii) $(\mathcal{D}, \xi)$ is $\mathcal{T}$-F-*good* if they are $\mathcal{T}$-good and moreover for all environments ρ and $A \in \mathbb{T}^{\mathcal{T}}$

$$[\![x]\!]_{\rho} \in [\![A]\!]_{\xi} \;\Rightarrow\; [\![\lambda y.xy]\!]_{\rho} \in [\![A]\!]_{\xi};$$

(iii) $(\mathcal{D}, \xi)$ *preserve* $\leq_{\mathcal{T}}$ iff for all $A, B \in \mathbb{T}^{\mathcal{T}}$:

$$A \leq_{\mathcal{T}} B \;\Rightarrow\; [\![A]\!]_{\xi} \subseteq [\![B]\!]_{\xi}.$$

We now introduce the semantics of type assignment.

19.1.4. Definition (Semantic Satisfiability). Let $\mathcal{T}$ be a $\mathrm{TT}^{\top}$.

(i) Let $\mathcal{D} = \langle \mathcal{D}, \cdot, [\![\ ]\!] \rangle$ be a quasi λ-model. Define

$$\begin{aligned} \mathcal{D}, \rho, \xi \models M : A \quad &\Leftrightarrow \quad [\![M]\!]_\rho \in [\![A]\!]_\xi; \\ \mathcal{D}, \rho, \xi \models \Gamma \quad &\Leftrightarrow \quad \mathcal{D}, \rho, \xi \models x : B, \text{for all } (x{:}B) \in \Gamma. \end{aligned}$$

(ii)

$$\begin{aligned} \Gamma \models^{\mathcal{T}}_{\cap^\top} M : A \quad \Leftrightarrow \quad & \mathcal{D}, \rho, \xi \models \Gamma \ \Rightarrow \ \mathcal{D}, \rho, \xi \models M : A, \text{ for all} \\ & \text{for all } \mathcal{D}, \xi, \rho \text{ such that } (\mathcal{D}, \xi) \text{ are } \to\text{-good and preserve } \leq_{\mathcal{T}}. \end{aligned}$$

In order to distinguish it from other notions of semantics $\models^{\mathcal{T}}$ is called the *simple* semantics.

Derivability in the type system implies semantic satisfiability, as shown in the next theorem.

19.1.5. Theorem (Soundness). *For all $\mathcal{T}$ one has*

$$\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M : A \ \Rightarrow \ \Gamma \models^{\mathcal{T}}_{\cap^\top} M : A.$$

Proof. By induction on the derivation of $\Gamma \vdash^{\mathcal{T}}_{\cap^\top} M : A$. Rules $(\to\text{E})$, $(\cap\text{I})$ and $(\top\text{-universal})$ are sound by the definition of type interpretation (Definition 19.1.2).

As to the soundness of rule $(\to\text{I})$, assume $\Gamma, x{:}A \vdash^{\mathcal{T}}_{\cap^\top} M : B$ in order to show $\Gamma \models^{\mathcal{T}} (\lambda x.M) : (A{\to}B)$. Assuming $\mathcal{D}, \rho, \xi \models \Gamma$ we have to show

$$[\![\lambda x.M]\!]^{\mathcal{D}}_\rho \in [\![A]\!]^{\mathcal{D}}_\xi \to [\![B]\!]^{\mathcal{D}}_\xi.$$

Let $d \in [\![A]\!]^{\mathcal{D}}_\xi$. We are done if we can show

$$[\![M]\!]^{\mathcal{D}}_{\rho[x:=d]} \in [\![B]\!]^{\mathcal{D}}_\xi,$$

because $(\mathcal{D}, \xi)$ are $\to$-good. Now $\mathcal{D}, \rho[x := d], \xi \models \Gamma, x{:}A$, hence $[\![M]\!]^{\mathcal{D}}_{\rho[x:=d]} \in [\![B]\!]^{\mathcal{D}}_\xi$, by the induction hypothesis for $\Gamma, x{:}A \vdash^{\mathcal{T}}_{\cap^\top} M : B$.

Rule $(\leq)$ is sound, as we consider only $(\mathcal{D}, \xi)$ that preserve $\leq_{\mathcal{T}}$. ∎

As to completeness, first we observe that only lazy type structures (see Definition ??) can be complete.

19.1.6. Proposition (Adequacy implies laziness). *Suppose*

$$\forall \Gamma, M, A [\Gamma \models^{\mathcal{T}}_{\cap^\top} M : A \ \Rightarrow \ \Gamma \vdash^{\mathcal{T}}_{\cap^\top} M : A].$$

Then $\mathcal{T} \in$ $\mathbf{LTS}^\top$.

Comment: the previous schema was sensible considering more semantics, now we have only the simple semantics, therefore I put the proof of this proposition inside that one of the completeness theorem

Completeness

Now we characterize those theories which are complete. with respect to the given semantics.

19.1.7. Definition. Let $\mathcal{T}$ be a NTT.

(i) Define $\mathcal{D}^{\mathcal{T}} = \langle \mathcal{F}^{\mathcal{T}}, \cdot, [\![\ ]\!]^{\mathcal{T}} \rangle$, with $[\![\ ]\!]^{\mathcal{T}}$ as in Definition 18.1.4(i).

(ii) Let $\xi^{\mathcal{T}} : \mathbb{A}^{\mathcal{T}} \to \mathsf{P}(\mathcal{F}^{\mathcal{T}})$ be the type environment defined by

$$\xi^{\mathcal{T}}(\alpha) = \{X \in \mathcal{F}^{\mathcal{T}} \mid \alpha \in X\}.$$

(iii) Let $[\![\,]\!]^{\mathcal{T}} : \mathbb{T}^{\mathcal{T}} \to \mathsf{P}(\mathcal{F}^{\mathcal{T}})$ be the mapping $[\![\,]\!]_{\xi^{\mathcal{T}}}$.

The mapping $[\![\,]\!]^{\mathcal{T}} : \mathbb{T}^{\mathcal{T}} \to \mathsf{P}(\mathcal{F}^{\mathcal{T}})$ has the property of associating to each type A the set of filters which contain A (thus preserving the property which defines $\xi^{\mathcal{T}}$ in the basic case of type constants).

19.1.8. Proposition. *Let $\mathcal{T}$ be a NTT. Then we have*

$$[\![A]\!]^{\mathcal{T}} \quad = \quad \{X \in \mathcal{F}^{\mathcal{T}} \mid A \in X\}.$$

Proof. By induction on A. The only interesting case is when A is an arrow type. If $A \equiv B \to C$ we have

$$\begin{array}{rcll}
[\![B\to C]\!]^{\mathcal{T}} & = & \{X \in \mathcal{F}^{\mathcal{T}} \mid \forall Y \in [\![B]\!]^{\mathcal{T}}.\, X \cdot Y \in [\![C]\!]^{\mathcal{T}}\} & \text{by definition,} \\
 & = & \{X \in \mathcal{F}^{\mathcal{T}} \mid \forall Y.\, B \in Y \Rightarrow C \in X \cdot Y\} & \text{by induction,} \\
 & = & \{X \in \mathcal{F}^{\mathcal{T}} \mid C \in X \cdot \uparrow B\} & \text{by monotonicity,} \\
 & = & \{X \in \mathcal{F}^{\mathcal{T}} \mid C \in \uparrow\{C' \mid \exists B' \in \uparrow B. B' \to C' \in X\}\}, & \text{by the definition of} \\
 & & & \text{filter application,} \\
 & = & \{X \in \mathcal{F}^{\mathcal{T}} \mid B \to C \in X\}, & \text{by } (\to) \text{ and } (\top\to). \blacksquare
\end{array}$$

19.1.9. Lemma. *Let $\mathcal{T}$ be a NTT. Then $(\mathcal{D}^{\mathcal{T}}, \xi^{\mathcal{T}})$ are $\to$-good and preserve $\leq_{\mathcal{T}}$.*

Proof. For condition (i) of Definition 19.1.3 let $X \in [\![A]\!]^{\mathcal{T}}$ be such that

$$[\![M]\!]_{\rho[x:=X]} \in [\![B]\!]^{\mathcal{T}}.$$

Then, by Proposition 19.1.8, $B \in [\![M]\!]_{\rho[x:=X]}$, hence $B \in f(X)$, where we have put $f = \boldsymbol{\lambda} d.[\![M]\!]_{\rho[x:=d]}$. Since**Comment:** the reason is that for each filter structure F, G give a Galois connection, this was stated before but I do not find it anymore! one has $f \sqsubseteq F^{\mathcal{T}}(G^{\mathcal{T}}(f))$, it follows that $B \in F^{\mathcal{T}}(G^{\mathcal{T}}(f))(X)$. So $F^{\mathcal{T}}(G^{\mathcal{T}}(f))(X) \in [\![B]\!]^{\mathcal{T}}$, by Proposition 19.1.8. We are done since $F^{\mathcal{T}}(G^{\mathcal{T}}(f))(X) = [\![\lambda x.M]\!]_\rho \cdot X$, by Definition 18.1.4(i).

Lastly notice that as an immediate consequence of the Proposition 19.1.8 we get

$$A \leq_{\mathcal{T}} B \Leftrightarrow \forall X \in \mathcal{F}^{\mathcal{T}}.[A \in X \Rightarrow B \in X] \Leftrightarrow [\![A]\!]^{\mathcal{T}} \subseteq [\![B]\!]^{\mathcal{T}}_{,}$$

and therefore $(\mathcal{D}^{\mathcal{T}}, \xi^{\mathcal{T}})$ preserve $\leq_{\mathcal{T}}$. $\blacksquare$

Finally we can prove the desired completeness result.

19.1.10. Theorem. *(Completeness for the Simple Semantics)*

(i) $[\Gamma \models^{\mathcal{T}}_{\cap^{\top}} M : A \;\Rightarrow\; \Gamma \vdash^{\mathcal{T}}_{\cap^{\top}} M : A]$ *iff* $\mathcal{T}$ *is a* NTT.

(ii) *Let* $\mathcal{T}$ *be a NTT. Then*

$$\Gamma \models^{\mathcal{T}}_{\cap^{\top}} M : A \;\Leftrightarrow\; \Gamma \vdash^{\mathcal{T}}_{\cap^{\top}} M : A.$$

Proof. (i) ($\Rightarrow$) From Proposition 19.1.6 it follows that $\mathcal{T} \in \mathbf{LTS}^{\top}$. It is easy to verify that all type interpretations validate rule ($\to$) and the axioms ($\to\cap$), and ($\top$). and ($\top_{\mathrm{lazy}}$). For instance, as regards to axiom ($\to\cap$), consider the $\mathcal{T}$-basis $\Gamma = \{x{:}(A \to B) \cap (A \to C)\}$. From Definition 19.1.2 we get $\Gamma \models^{\mathcal{T}}_{\cap^{\top}} x : A{\to}(B \cap C)$. Hence, by hypothesis, we have $\Gamma \vdash^{\mathcal{T}}_{\cap^{\top}} x : A{\to}(B \cap C)$. Using Theorem 16.1.10(i) it follows that $(A{\to}B){\cap}^{\top}(A{\to}C) \leq_{\mathcal{T}} A{\to}B \cap C$. Therefore axiom ($\to\cap$) holds. As to ($\top\to$) we have $\models^{\mathcal{T}}_{\cap^{\top}} x : \top \to \top$, since $[\![x]\!]_{\rho} \cdot d \in \mathcal{D}$ for all $\mathcal{D}, d \in \mathcal{D}$ and $\rho : \mathsf{Env}_{\mathcal{D}} \to \mathcal{D}$. Now by assumption

$$\begin{array}{lll}
\models^{\mathcal{T}}_{\cap^{\top}} x : \top \to \top & \Rightarrow & \vdash^{\mathcal{T}}_{\cap^{\top}} x : (\top{\to}\top) \\
& \Rightarrow & x{:}\top \vdash^{\mathcal{T}}_{\cap^{\top}} x : (\top{\to}\top) \\
& \Rightarrow & \top \leq_{\mathcal{T}} (\top{\to}\top), \qquad \text{by Theorem 16.1.10(i).}
\end{array}$$

This proves ($\Rightarrow$).

($\Leftarrow$) To show that $\Gamma \models^{\mathcal{T}} M : A$ implies $\Gamma \vdash^{\mathcal{T}}_{\cap^{\top}} M : A$ under the given conditions we use the quasi λ-model $\langle \mathcal{F}^{\mathcal{T}}, \cdot, [\![\;]\!] \rangle$ induced by the lambda structure $\langle \mathcal{F}^{\mathcal{T}}, F^{\mathcal{T}}, G^{\mathcal{T}} \rangle$. So by Lemma 19.1.9 we have that $\Gamma \models^{\mathcal{T}}_{\cap^{\top}} M : A$ implies $[\![M]\!]_{\rho_{\Gamma}} \in [\![A]\!]^{\mathcal{T}}$, where $\rho_{\Gamma}(x) = \begin{cases} \uparrow A & \text{if } x{:}A \in \Gamma, \\ \uparrow \top & \text{otherwise.} \end{cases}$

We conclude $\Gamma \vdash^{\mathcal{T}}_{\cap^{\top}} M : A$ using Proposition 19.1.8.

(ii) By Proposition 19.1.5 and (i). ∎

The strict story

In this subsection $\mathcal{T}$ is always a proper intersection type theory, with or without top element. That is $\mathcal{T}$ is a PTT, c.f. Definition 15.1.9. Filters are strict filters, so they may be empty, c.f. Definition 15.4.5, and we do not have the axiom ($\top$-universal) for type assignment. The definitions are very similar to the non-strict case. At many places one adds an index 's' and everywhere one erases $\top$. Also the statements are very similar and proofs will be omitted mostly.

We keep Definitions 19.1.1-19.1.4, except that in Definition 19.1.2 we now define $[\![\alpha]\!]^{\mathcal{D}}_{\xi,s} = \xi(\alpha)$ for $\alpha \in \mathbb{A}^{\mathcal{T}}$ and we omit the clause $[\![\top]\!]^{\mathcal{D}}_{\xi,s} = \mathcal{D}$.

19.1.11. Proposition (Soundness). *For all PTT* $\mathcal{T}$ *one has*

$$\Gamma \vdash^{\mathcal{T}}_{\cap} M : A \;\Rightarrow\; \Gamma \models^{\mathcal{T}}_{\cap} M : A.$$

Proof. As for Proposition 19.1.5, without the case ($\top$-universal). ∎

19.1.12. PROPOSITION (Adequacy implies strict naturality). *Suppose*

$$\forall \Gamma, M, A\, [\Gamma \models^{\mathcal{T}}_{i,s} M : A \;\Rightarrow\; \Gamma \vdash^{\mathcal{T}}_{\cap^{\mathcal{T}}} M : A].$$

Then $\mathcal{T} \in \mathcal{TT}^{\mathsf{l}}$.

PROOF. As for Proposition 19.1.6, without the cases (Ω) and (Ω-lazy). ∎

19.1.13. DEFINITION. Let $\mathcal{T}$ be a PTT.

(i) Define $\mathcal{D}^{\mathcal{T}}_s = \langle \mathcal{F}^{\mathcal{T}}_s, \cdot, [\![\ ]\!]^{\mathcal{F}^{\mathcal{T}}_s} \rangle$, with $[\![\ ]\!]^{\mathcal{F}^{\mathcal{T}}_s}_s$ as in Definition 18.1.4(ii).

(ii) Let $\xi^{\mathcal{T}}_s : \mathbb{A}^{\mathcal{T}} \to \mathsf{P}(\mathcal{F}^{\mathcal{T}}_s)$ be the type environment defined by the map

$$\xi^{\mathcal{T}}_s(\alpha) = \{X \in \mathcal{F}^{\mathcal{T}}_s \mid \alpha \in X\}.$$

(iii) Let $[\![\ ]\!]^{\mathcal{T}}_s : \mathbb{T}^{\mathcal{T}} \to \mathsf{P}(\mathcal{F}^{\mathcal{T}}_s)$ be the mapping $[\![\ ]\!]_{\xi^{\mathcal{T}}_s, s}$.

19.1.14. PROPOSITION. *Let $\mathcal{T}$ be a PTT. Then we have*

$$[\![A]\!]^{\mathcal{T}}_s = \{X \in \mathcal{F}^{\mathcal{T}}_s \mid A \in X\}.$$

PROOF. Similar to the proof of Proposition 19.1.8. ∎

19.1.15. LEMMA. *Let $\mathcal{T}$ be a PTT. Then $\mathcal{D}^{\mathcal{T}}_s$, $\xi^{\mathcal{T}}_s$ are $\to$-good and preserve $\leq_{\mathcal{T}}$.*

PROOF. Similar to the proof of Lemma 19.1.9. ∎

19.1.16. THEOREM. *(Completeness for the strict Simple Semantics)*

(i) $[\Gamma \models^{\mathcal{T}}_{\cap} M : A \;\Rightarrow\; \Gamma \vdash^{\mathcal{T}}_{\cap} M : A]$ *iff $\mathcal{T}$ is a PTT.*

(ii) *Let $\mathcal{T}$ be a PTT. Then*

$$\Gamma \models^{\mathcal{T}}_{\cap} M : A \;\Leftrightarrow\; \Gamma \vdash^{\mathcal{T}}_{\cap} M : A.$$

PROOF. (i) Similar to the proof of Theorem 19.1.10.

(ii) By Proposition 19.1.11 and (i). ∎

19.2. Characterizing syntactic properties

In this section we will see the intersection type systems at work in the characterization of properties of λ-terms. Since types are preserved by reduction, we can characterize only properties which induce equivalences that are preserved by reduction. In particular we will consider some normalization properties of λ-terms, i.e. the standard properties of having a head normal form or a normal form, and of being strongly normalizable.

First we recall some basic definitions.

19.2.1. DEFINITION (Severi [1996]). The set SN is the least set of terms closed under the following rules.

$$\frac{M_1 \in \mathsf{SN}, \ldots, M_n \in \mathsf{SN}}{x M_1 \ldots M_n \in \mathsf{SN}}\; n \geq 0$$

$$\frac{M \in \mathsf{SN}}{\lambda x.M \in \mathsf{SN}}$$

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

19.2.2. Lemma. *$M \in \mathsf{SN} \Leftrightarrow M$ is strongly normalizing.*

Proof. ($\Rightarrow$) By induction on the generation of SN.

($\Leftarrow$) By induction on M it follows that if M is a nf, then $M \in \mathsf{SN}$. Now suppose that M is strongly normalizing. Let $||M||$, the *norm* of M, be the length of the longest reduction path starting with M. Then by induction on $||M||$ one can show that $M \in \mathsf{SN}$. The case $||M|| = n > 0$ is done by induction on the structure of M, being $x\vec{M}, \lambda x.N$ or $(\lambda x.P)Q\vec{M}$. In the first two cases the result follows from the induction hypothesis for the structure of M, in the third case from the induction hypothesis for $||M||$. ■

19.2.3. Definition. (i) A term M is *persistently head normalizing* iff $M\vec{N}$ has a head normal form for all terms $\vec{N}$.

(ii) A term M is *persistently normalizing* iff $M\vec{N}$ has a normal form for all normalizable terms $\vec{N}$.

The notion of persistently normalizing terms has been introduced in Böhm and Dezani-Ciancaglini [1975]. The following predicates will come in handy in the sequel.

19.2.4. Definition.

(i) $M \in \mathsf{HN} \Leftrightarrow M$ has a head normal form.

(ii) $M \in \mathsf{PHN} \Leftrightarrow M$ is persistently head normalizing.

(iii) $M \in \mathsf{N} \Leftrightarrow M$ has a normal form.

(iv) $M \in \mathsf{PN} \Leftrightarrow M$ is persistently normalizing.

(v) $M \in \mathsf{SN} \Leftrightarrow M$ is strongly normalizing.

(vi) $M \in \mathsf{NF} \Leftrightarrow M$ is a normal form. **Comment:** I think we do not use NF any more and I suggest to erase them, right?

The following inclusions follow immediately by definition, but the inclusion $\mathsf{PN} \subseteq \mathsf{PHN}$, see Exercise 19.6.3.

$$\begin{array}{l} \mathsf{NF} \subseteq \mathsf{SN} \subseteq \mathsf{N} \subseteq \mathsf{HN} \\ \mathsf{PN} \subseteq \mathsf{N} \subseteq \mathsf{HN} \\ \mathsf{PN} \subseteq \mathsf{PHN} \subseteq \mathsf{HN} \end{array}$$

The above inclusions are visualized in Figure 19.1.

19.2.5. Example. (i) $\lambda x.x$ and $\lambda y.y(\lambda x.xx)$ are in N and in HN but they are not in PN nor in PHN.

(ii) $\lambda y.x\Omega$ is in PHN but not in PN.

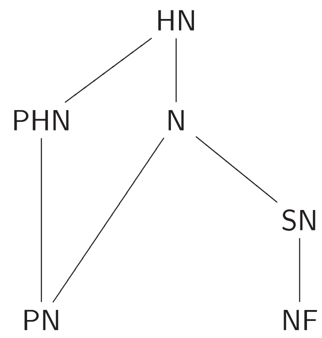

Figure 19.1: Inclusions between sets of λ-terms

Stable sets

Comment: moved here: it was after the main theorem

We will use the standard proof technique of type stable sets (Krivine [1990]).

We take as quasi λ-model $\mathcal{D}_\Lambda = \langle \Lambda, \cdot, [\![\ ]\!] \rangle$, where Λ is the set of λ-terms modulo $\boldsymbol{\beta}$-conversion, i.e. the term model of $\boldsymbol{\beta}$-equality, and therefore $[\![M]\!]_\rho = \{N \mid N =_{\boldsymbol{\beta}} M[\vec{x} := \rho(\vec{x})] \text{ where } \vec{x} = FV(M)\}$, $[\![M]\!]_\rho \cdot [\![N]\!]_\rho = [\![MN]\!]_\rho$. Notice that $X {\to} Y = \{M \in \Lambda \mid \forall N \in X\ MN \in Y\}$.

19.2.6. Definition. (i) A set $X \subseteq \Lambda$ is called *closed under head expansion of redexes* if

$$M[x := N]\vec{M} \in X \text{ implies } (\lambda x.M)N\vec{M} \in X.$$

The term N is called the *argument* of the head expansion.

(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 redexes whose argument is in SN.

From the above definition and Definition 19.2.1 we easily get the following.

19.2.7. Proposition. *Let* $\mathsf{S} \in \{\mathsf{HN}, \mathsf{N}, \mathsf{SN}\}$.

(i) *Each* S *is* S-type-*stable.*

(ii) PHN *is* HN-*stable and* PN *is* N-*stable.*

(iii) *If* X, Y *are* S-type-*stable, then* $X \Rightarrow Y$ *is* S-type-*stable.*

(iv) *If* Y *is* HN-type-*stable, then* $Z \Rightarrow Y$ *is* HN-type-*stable, for* $\emptyset \neq Z \subseteq \Lambda$.

(v) *If* X, Y *are* S-type-*stable then* $X \cap Y$ *is* S-type-*stable.* ∎

19.2.8. Lemma. (i) a. *If* $A \neq \top$, *then* $[\![A]\!]_{\mathcal{V}^1_{\mathrm{BCD}}}$ *is* HN-type-*stable.*

b. *If* $A \neq \top$, *then* $[\![A]\!]_{\mathcal{V}_{\mathrm{DHM}}}$ *is* HN-type-*stable.*

(ii) a. *If* $\top \notin A$, *then* $[\![A]\!]_{\mathcal{V}^2_{\mathrm{BCD}}}$ *is* N-type-*stable.*

b. *If* $\top \notin A$, *then* $[\![A]\!]_{\mathcal{V}_{\mathrm{CDZ}}}$ *is* N-type-*stable.*

(iii) $[\![A]\!]_{\mathcal{V}_{\mathrm{CDV}}}$ *is* SN-*type-stable.*

PROOF. All points follow easily from Proposition 19.2.7. ∎

We need some definitions of type interpretation and semantic satisfability that will be used also in Section 19.1 in a slight variation.

19.2.9. DEFINITION (Type Interpretation). The *interpretation* of types in $\mathbb{T}^{\mathcal{T}}$ induced by the type environment $\xi : \mathbb{A}^{\mathcal{T}} \to \mathcal{P}(\Lambda)$ is defined as follows.

$$
\begin{array}{rcll}
[\![A]\!]_{\xi} & = & \xi(A) & \text{if } A \in \mathbb{A}^{\mathcal{T}}; \\
[\![A \to B]\!]_{\xi} & = & [\![A]\!]_{\xi} \Rightarrow [\![B]\!]_{\xi}; & \\
[\![A \cap^{\top} B]\!]_{\xi} & = & [\![A]\!]_{\xi} \cap [\![B]\!]_{\xi}. &
\end{array}
$$

19.2.10. DEFINITION (Type environments). (i) The *type environment* $\xi = \xi^1_{\mathrm{BCD}}$ is defined as follows.

$$
\begin{array}{rcll}
[\![A]\!]_{\xi} & = & \mathsf{HN}, & \text{if } A \in \mathbb{A}_{\infty}; \\
[\![\top]\!]_{\xi} & = & \Lambda. &
\end{array}
$$

(ii) The *type environment* $\xi = \xi^2_{\mathrm{BCD}}$ is defined as follows.

$$
\begin{array}{rcll}
[\![A]\!]_{\xi} & = & \mathsf{N}, & \text{if } A \in \mathbb{A}_{\infty}; \\
[\![\top]\!]_{\xi} & = & \Lambda. &
\end{array}
$$

(iii) The *type environment* $\xi = \xi_{\mathrm{DHM}}$ is defined as follows.

$$
\begin{array}{rcl}
[\![\omega]\!]_{\xi} & = & \mathsf{PHN}; \\
[\![\varphi]\!]_{\xi} & = & \mathsf{HN}; \\
[\![\top]\!]_{\xi} & = & \Lambda.
\end{array}
$$

(iv) The *type environment* $\xi = \xi_{\mathrm{CDZ}}$ is defined as follows.

$$
\begin{array}{rcl}
[\![\omega]\!]_{\xi} & = & \mathsf{PN}; \\
[\![\varphi]\!]_{\xi} & = & \mathsf{N}; \\
[\![\top]\!]_{\xi} & = & \Lambda.
\end{array}
$$

(v) The *type environment* $\xi = \xi_{\mathrm{CDV}}$ is defined as follows.

$$
[\![A]\!]_{\xi} \;=\; \mathsf{SN}, \qquad \text{if } A \in \mathbb{A}_{\infty}.
$$

19.2.11. DEFINITION. A type environment ξ is called *nice* for $\mathcal{T}$ iff

(i) $[\![\Omega]\!]_{\xi} = \Lambda$.

(ii) $A \leq_{\mathcal{T}} B \;\Rightarrow\; [\![A]\!]_{\xi} \subseteq [\![B]\!]_{\xi}$.

(iii) $M[x := N] \in [\![A]\!]_{\xi}, N \in [\![B]\!]_{\xi} \;\Rightarrow\; (\lambda x.M)N \in [\![A]\!]_{\xi}$.

We shall show that each type environment of Definition 19.2.10 is nice for its implicit $\mathcal{T}$. The proof occupies 19.2.12-19.2.14.

We shall show that for each type environment $\xi_{\mathcal{T}}$ of Definition 19.2.10 $(\mathcal{D}_\Lambda, \xi_{\mathcal{T}})$ are $\mathcal{T}$-good and preserve $\leq_{\mathcal{T}}$. The proof occupies 19.2.12-??.

19.2.12. Lemma. (i) $M \in \mathsf{N}, N \in \mathsf{PN} \;\Rightarrow\; M[x := N] \in \mathsf{N}$.

(ii) $M \in \mathsf{N}, N \in \mathsf{PN} \;\Rightarrow\; MN \in \mathsf{N}$.

(iii) $M \in \mathsf{HN}, N \in \mathsf{PHN} \;\Rightarrow\; M[x := N] \in \mathsf{N}$.

(iv) $M \in \mathsf{HN}, N \in \mathsf{PHN} \;\Rightarrow\; MN \in \mathsf{HN}$.

Proof. By an easy induction on the (head) normal form of M. ■

19.2.13. Proposition. (i) $\mathsf{PN} = (\mathsf{N} {\mapsto} \mathsf{PN})$.

(ii) $\mathsf{N} = (\mathsf{PN} {\mapsto} \mathsf{N})$.

(iii) $\mathsf{PHN} = (\mathsf{HN} {\mapsto} \mathsf{PHN})$.

(iv) $\mathsf{HN} = (\mathsf{PHN} {\mapsto} \mathsf{HN})$.

Proof. All cases are immediate except the inclusions $\mathsf{N} \subseteq (\mathsf{PN} {\mapsto} \mathsf{N})$ and $\mathsf{HN} \subseteq (\mathsf{PHN} {\mapsto} \mathsf{HN})$. These follow easily from Lemma 19.2.12(ii) and (iv). ■

19.2.14. Lemma. *For all $\xi_{\mathcal{T}}$ of Definition 19.2.10 we have the following.*

(i) $\forall N \in [\![B]\!]_{\xi_{\mathcal{T}}}, M[x := N] \in [\![A]\!]_{\xi_{\mathcal{T}}}$ *implies* $\lambda x.M \in [\![B \to A]\!]_{\xi_{\mathcal{T}}}$.

(ii) $A \leq_{\mathcal{T}} B \;\Rightarrow\; [\![A]\!]_{\xi_{\mathcal{T}}} \subseteq [\![B]\!]_{\xi_{\mathcal{T}}}$.

I.e. for all $\xi_{\mathcal{T}}$ of Definition 19.2.10 $(\mathcal{D}_\Lambda, \xi_{\mathcal{T}})$ are $\to$-good and preserve $\leq_{\mathcal{T}}$.

Proof. (i) If either $\mathcal{T} \neq \mathrm{CDV}$ or $\mathcal{T} = \mathrm{CDV}$ and $N \in \mathsf{SN}$ one easily shows that $M[x := N] \in [\![A]\!]_{\xi_{\mathcal{T}}}$ implies $(\lambda x.M)N \in [\![A]\!]_{\xi_{\mathcal{T}}}$ by induction on A using Proposition 19.2.7. The conclusion from the definition of $\to$.

(ii) By induction on the generation of $\leq_{\mathcal{T}}$, using Proposition 19.2.13. ■

19.2.15. Definition (Semantic Satisfiability). Let $\rho : \mathrm{V} \to \Lambda$.

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

(ii) $\rho, \xi \models M : A \;\Leftrightarrow\; [\![M]\!]_\rho \in [\![A]\!]_\xi$.

(iii) $\rho, \xi \models \Gamma \;\Leftrightarrow\; \rho, \xi \models x : B$, for all $(x{:}B) \in \Gamma$.

(iv) $\Gamma \models^{\mathcal{T}} M : A \;\Leftrightarrow\; \rho, \xi \models \Gamma$ implies $\rho, \xi \models M : A$, for all nice ξ and all ρ.

19.2.16. Theorem (Soundness). $\Gamma \vdash^{\mathcal{T}} M : A \;\Rightarrow\; \Gamma \models^{\mathcal{T}} M : A$.

Proof. By induction on the derivation of $\Gamma \vdash M : A$. ■

19.2.17. Lemma. (i) *a.* $A \not\sim \Omega \;\Rightarrow\; [\![A]\!]_{\mathcal{V}^1_{\mathrm{BCD}}}$ *is* HN*-type-stable.*

b. $A \sim \Omega \;\Rightarrow\; [\![A]\!]_{\mathcal{V}^1_{\mathrm{BCD}}} = \Lambda$.

(ii) *a.* $\Omega \notin A \;\Rightarrow\; [\![A]\!]_{\mathcal{V}^2_{\mathrm{BCD}}}$ *is* N*-type-stable.*

b. $x \in \mathsf{PN}$ *for* $x \in \mathrm{Var}$.

(iii) $[\![A]\!]_{\mathcal{V}_{\mathrm{CDV}}}$ *is* SN*-type-stable.*

Proof. (i)a., (ii)a. and (iii) follow easily from Lemma 19.2.7. (i)b. follows directly from Lemma 15.1.16. (ii)b. holds trivially. ■

In the following result several important syntactic properties of lambda terms are characterized by typability in some assignment system related to some intersection type theory. $\mathrm{FV}(M){:}\omega \vdash$ stands for $x_1{:}\omega, \ldots, x_n{:}\omega \vdash$, where $\{x_1,\ldots,x_n\} = \mathrm{FV}(M)$.**Comment:** we used Γ^M_ω in section 18, we need to choose 1 notation! and $\Lambda\mathrm{I}$ is the set of terms reducing to a $\lambda\mathrm{I}$-term; finally $\mathcal{T}\ni\Omega$ indicates that Ω is one of the constants of $\mathcal{T}$.

19.2.18. Theorem (Characterization Theorems). *Let $M\in\Lambda$ and let $\mathcal{T}$ be an arbitrary intersection type theory.*

$$
\begin{array}{lll}
\text{(i)} \quad M\in\mathsf{N} & \Leftrightarrow & \forall\ TT^{\top}\mathcal{T}\,\exists\Gamma, A.\Gamma\vdash^{\mathcal{T}}_{\cap^\top} M : A \ \&\ \top\notin\Gamma, A \\
& \Leftrightarrow & \exists\Gamma, A.\Gamma\vdash^{\mathrm{BCD}}_{\cap^\top} M : A \ \&\ \top \notin \Gamma, A. \\
& \Leftrightarrow & \mathrm{FV}(M) : \omega\vdash^{\mathrm{CDZ}}_{\cap^\top} M : \varphi. \\
\text{(ii)} \quad M\in\mathsf{HN} & \Leftrightarrow & \forall\ TT^{\top}\mathcal{T}\,\exists\Gamma\,\exists n, m\in\mathbb{N}.\Gamma\vdash^{\mathcal{T}}_{\cap^\top} M : (\top^m{\to}A)^n{\to}A \\
& \Leftrightarrow & \exists\Gamma, A.\Gamma\vdash^{\mathrm{BCD}}_{\cap^\top} M : A \ \&\ A \neq_{\mathrm{BCD}} \top \\
& \Leftrightarrow & \mathrm{FV}(M) : \omega\vdash^{\mathrm{DHM}}_{\cap^\top} M : \varphi. \\
\text{(iii)} \quad M\in\mathsf{SN} & \Leftrightarrow & \forall\mathcal{T}\ \exists\Gamma, A.\Gamma\vdash^{\mathcal{T}}_{\cap} M : A. \\
& \Leftrightarrow & \exists\Gamma, A.\Gamma\vdash^{\mathrm{CDV}}_{\cap} M : A.
\end{array}
$$

Proof. $(\Rightarrow)$.

(i) By Corollary 16.2.5(ii) it suffices to consider M in nf. The proof is by induction on M. The only interesting case is $M\equiv x\vec{M}$ where $\vec{M}\equiv M_1\ldots M_m$. By the induction hypothesis we have $\Gamma_j\vdash^{\mathcal{T}} M_j : A_j$, for some Γ_j, A_j not containing $\top$ and for $j\le m$. This implies: $\uplus_{j\le m}\Gamma_j\uplus\{x{:}A_1{\to}\ldots{\to}A_m{\to}A\}\vdash^{\mathcal{T}}_{\cap^\top} x\vec{M} : A$. Therefore $\forall\ \mathrm{TT}^{\top}\mathcal{T}\,\exists\Gamma, A.\Gamma\vdash^{\mathcal{T}}_{\cap^\top} M : A\ \&\ \top\notin\Gamma, A$ and in particular this holds for $\mathcal{T}=\mathrm{BCD}$.

For $\lambda^{\mathrm{CDZ}}_{\cap^\top}$ we show by induction on M that if $\Gamma=\{x{:}\omega\mid x\in FV(M)\}$, then $\Gamma\vdash M : \varphi$. If $M\equiv x\vec{M}$ then by the induction hypothesis and weakening we have $\Gamma\vdash^{\mathrm{CDZ}}_{\cap^\top} M_j : \varphi$. As $\omega=\varphi\to\omega$ in CDZ, this implies $\Gamma\vdash^{\mathrm{CDZ}}_{\cap^\top} x\vec{M} : \omega$.
By rule $(\le_{\mathrm{CDZ}})$ we conclude $\Gamma\vdash^{\mathrm{CDZ}}_{\cap^\top} M : \varphi$. If $M\equiv\lambda y.N$ then by the induction hypothesis we have $\Gamma, y:\omega\vdash^{\mathrm{CDZ}}_{\cap^\top} N : \varphi$ and this implies $\Gamma\vdash^{\mathrm{CDZ}}_{\cap^\top} M : \omega\to\varphi$. By rule $(\le_{\mathrm{CDZ}})$ we conclude $\Gamma\vdash^{\mathrm{CDZ}}_{\cap^\top} M : \varphi$.

(ii) As in the proof of (i), we can consider M in hnf. Let $M\equiv\lambda y_1\ldots y_n.xM_1\ldots M_m$. We have $x{:}\top^m{\to}A\vdash^{\mathcal{T}}_{\cap^\top} xM_1\ldots M_m : A$ by $({\to}\mathrm{E})$. By rules (*weakening*) and $({\to}\mathrm{I})$ this implies $x{:}\top^m{\to}A\vdash^{\mathcal{T}}_{\cap^\top} M : (\top^m{\to}A)^n{\to}A$.
Being A arbitrary we can choose it different from $\top$ in $\lambda^{\mathrm{BCD}}_{\cap^\top}$.

For $\lambda^{\mathrm{DHM}}_{\cap^\top}$ by choosing $A\equiv\omega$ we get from above $x{:}\Omega^m\to\omega\vdash^{\mathrm{DHM}}_{\cap^\top} M : (\Omega^m\to\omega)^n\to\omega$. By rules $(\le_{\mathrm{DHM}})$ and $(\le)$ this implies $x{:}\omega\vdash^{\mathrm{DHM}}_{\cap^\top} M : \varphi$ since $\omega=_{\mathrm{DHM}}\top\to\omega$, $\omega\le_{\mathrm{DHM}}\varphi$ and $\varphi=_{\mathrm{DHM}}\omega\to\varphi$.

(iii) By induction on the structure of strongly normalizing terms following Definition 19.2.1. We only consider the case $M\equiv(\lambda x.R)N\vec{M}$ with $\vec{M}\equiv M_1\ldots M_n$ and both $R[x:=N]\vec{M}$ and N are strongly normalizing. By the induction hypothesis there are Γ, A, Γ', B such that $\Gamma\vdash^{\mathcal{T}}_{\cap} R[x:=N]\vec{M} : A$ and

$\Gamma' \vdash_{\cap}^{\mathcal{T}} N : B$. We get $\Gamma \uplus \Gamma' \vdash_{\cap}^{\mathcal{T}} R[x := N]\vec{M} : A$ and $\Gamma \uplus \Gamma' \vdash_{\cap}^{\mathcal{T}} N : B$, so if $n = 0$ we are done by Theorem 16.2.4(i). If $n > 0$ by iterated applications of Theorem 16.1.1(ii) to $\Gamma \vdash_{\cap}^{\mathcal{T}} R[x := N]\vec{M} : A$ we have

$$\Gamma \vdash_{\cap}^{\mathcal{T}} R[x := N] : B_1^{(i)} \to \ldots \to B_n^{(i)} \to B^{(i)} \quad \Gamma \vdash_{\cap}^{\mathcal{T}} M_j : B_j^{(i)},\ (j \leq n)$$

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

The proof of ($\Leftarrow$) will occupy the rest of this section.

Now we are able to finish the proof of Theorem 19.2.18. Proof of Theorem 19.2.18

($\Leftarrow$). Take $\rho_0(x) = x$ for $x \in \mathrm{Var}$.

(i) Let $\Gamma \vdash_{\cap^\top}^{\mathrm{BCD}} M : A$ and $\top \notin A, \Gamma$. By soundness (Theorem 19.1.5) $\Gamma \models_{\cap^\top}^{\mathrm{BCD}} M : A$. By Lemmas 19.2.14 and 19.2.8(ii)a one has $\rho_0, \xi^2_{\mathrm{BCD}} \models \Gamma$, hence $M \in [\![A]\!]_{\xi^2_{\mathrm{BCD}}} \subseteq \mathsf{N}$, again by that Lemma. Let

$$FV(M){:}\omega \vdash_{\cap^\top}^{\mathrm{CDZ}} M : \varphi.$$

By Lemmas 19.2.14 and 19.2.8(ii)b one has $\rho_0, \xi_{\mathrm{CDZ}} \models \Gamma$, hence $M \in [\![\varphi]\!]_{\xi_{\mathrm{CDZ}}} = \mathsf{N}$, by Definition 19.2.10(iv).

(ii) Let $\Gamma \vdash_{\cap^\top}^{\mathrm{BCD}} M : A \neq \top$. By Lemmas 19.2.14 and 19.2.8(i)a one has $\rho_0, \xi^1_{\mathrm{BCD}} \models \Gamma$, hence by the same Lemma it follows that $M \in [\![A]\!]_{\xi^1_{\mathrm{BCD}}} \subseteq \mathsf{HN}$. Let

$$FV(M){:}\omega \vdash_{\cap^\top}^{\mathrm{DHM}} M : \varphi.$$

By Lemmas 19.2.14 and 19.2.8(i)b one has $\rho_0, \xi_{\mathrm{DHM}} \models \Gamma$, hence $M \in [\![\varphi]\!]_{\xi_{\mathrm{DHM}}} = \mathsf{HN}$, by Definition 19.2.10(iii).

(iii) Let $\Gamma \vdash_{\cap}^{\mathrm{CDV}} M : A$. By Lemmas 19.2.14 and 19.2.8(iii) one has $\rho_0, \xi_{\mathrm{CDV}} \models \Gamma$, hence $M \in [\![A]\!]_{\xi_{\mathrm{CDV}}} \subseteq \mathsf{SN}$, by the same Lemma. ∎

19.2.19. Remark. (i) For a $\mathrm{TT}^\top$ $\mathcal{T}$ one has

$$\exists A, \Gamma. \Gamma \vdash_{\cap^\top}^{\mathcal{T}} M : A \;\&\; \top \neq_{\mathcal{T}} A, \Gamma \not\Rightarrow M \in \mathsf{HN}.$$

Take for example $\mathcal{T} = \mathrm{Park}$, then $\vdash_{\cap^\top}^{\mathrm{Park}} (\lambda x.xx)(\lambda x.xx) : \omega \neq_{\mathrm{Park}} \top$, by Theorem 18.3.22, but this term is unsolvable.

(ii) There are many proofs of Point Theorem 19.2.18(iii) in the literature Pottinger [1981], Leivant [1986], van Bakel [1992], Krivine [1990], Ghilezan [1996], Amadio and Curien [1998]. As observed in Venneri [1996] all but Amadio and Curien [1998] contain some bugs, which in the case of Krivine [1990] can be easily remedied with a suitable non-standard notion of length of reduction path.

In Coppo et al. [1987] persistently normalizing normal forms have been given a similar characterization using the notion of replaceable variable (Coppo et al. [1987]). Other classes of terms are characterized in Dezani-Ciancaglini et al. [2005].

19.3. D_∞ models as filter models

In this section we give some applications of the results of Section 17.2, focusing on natural type structures and natural lambda-structures.

The first application is concerned with the lambda models D_∞, see Scott [1972] or Barendregt [1984]. These models are constructed from an initial D and it is not much of a restriction to consider them to be in **ALG**, i.e. the category of ω-algebraic lattices and Scott-continuous maps. To begin with, Corollary 17.2.14(ii)**Comment:** this is for $\mathbf{ALG}_a$, we need more! states that forall $\mathcal{D} \in \mathbf{NLS}$

$$\mathcal{D} \;\cong\; \mathcal{F}^{\mathcal{K}(\mathcal{D})},$$

(isomorphism within the category **NLS**). For a given lambda model D_∞ obtained from a $D_0 \in \mathbf{ALG}$ we have that $D_\infty \in \mathbf{NLS}$. We can construct directly (without having to construct first D_∞) a $\mathcal{S}_\infty \in \mathbf{NTS}$ such that

$$K(D_\infty) \cong \mathcal{S}_\infty,$$

obtaining for such D_∞

$$D_\infty \cong \mathcal{F}^{\mathcal{S}_\infty}.$$

The construction of D_∞ does not only depend on the initial D_0, but also on the projection pair $\mathsf{i}_0, \mathsf{j}_0$ which gives the start of the D_∞ construction:

$$\mathsf{i}_0 : D_0 \to D_1,\ \mathsf{j}_0 : D_1 \to D_0,$$

where $D_1 = [D_0 \to D_0]$. By making various variations on the theme $(D_0, \mathsf{i}_0, \mathsf{j}_0)$ one obtains the following versions of D_∞

$$D_\infty^{\text{Scott}}, D_\infty^{\text{Park}}, D_\infty^{\text{CDZ}}, D_\infty^{\text{DHM}} \text{ and } D_\infty^{\text{HR}},$$

each having some specific property. For $\mathcal{S} \in \{\text{Scott}, \text{Park}, \text{CDZ}, \text{DHM}, \text{HR}\}$???? we construct a **NTS** $\mathcal{S}_\infty$ such that

$$\mathcal{S}_\infty \cong \mathcal{S}.$$

Hence, writing $\mathcal{F}^{\mathcal{S}}$ for $\mathcal{F}^{\mathcal{S}}$ as usual, hence we obtain the isomorphisms

$$D_\infty^{\mathcal{S}} \cong \mathcal{F}^{\mathcal{S}}.$$

The pleasant fact is that $\mathcal{S}$ and the triple $t = (D_0, \mathsf{i}_0, \mathsf{j}_0)$ correspond to each other in a canonical way. For $\mathcal{S} \in \{\text{Scott}, \text{Park}\}$ one has that the model $D_\infty^{\mathcal{S}}$ was constructed first and the natural type structure $\mathcal{S}$ came later. For $\mathcal{S} \in \{\text{CDZ}, \text{DHM}, \text{HR}\}$ one first constructed the natural type structure $\mathcal{S}$ in order to obtain the model $D_\infty^{\mathcal{S}}$ satisfying a certain property.

The second application illustrates the expressiveness and flexibility of intersection types: following Alessi et al. [2001] we show that the term $\top \equiv (\lambda x.xx)(\lambda x.xx)$ is *easy* in the sense of Jacopini [1975], as has been shown in various ways, see e.g. Baeten and Boerboom [1979] or Mitschke's proof in Barendregt [1984], Proposition 15.3.9. Given any λ-term M, we inductively build natural intersection type structures $\mathcal{S}^n$ in such a way that the union of these structures, call it $\mathcal{S}^*$, forces the interpretation of M to coincide with the interpretation of $\top$.

Further applications of intersection types consist of necessary conditions for filter λ-models to be *sensible* or *semi-sensible*. We will not consider this issue, see Zylberajch [1991]. Comment: Berline students?

Let $\mathcal{D}$ be an ω-algebraic lattice that is reflexive (i.e. there are continuous maps $F : \mathcal{D} \to [\mathcal{D} \to \mathcal{D}]$ and $G : [\mathcal{D} \to \mathcal{D}] \to \mathcal{D}$ with $F \circ G = \mathrm{id}_{[\mathcal{D} \to \mathcal{D}]}$, see Barendregt [1984] Definition 5.4.1) and as such an extensional lambda-model (implying $G \circ F = \mathrm{id}_{\mathcal{D}}$, ibidem Theorem 5.4.4). Then $\mathcal{D}$ is clearly a natural lambda structure (in which only $F \circ G \sqsupseteq \mathrm{id}_{[\mathcal{D} \to \mathcal{D}]}$ and $G \circ F \sqsubseteq \mathrm{id}_{\mathcal{D}}$ are required). Now Theorem **??****Comment:** where is now this isomorphism result? disappeared? implies that any such λ-model $\mathcal{D}$ is isomorphic to a filter λ-model $\mathcal{F}^{\mathcal{S}}$, provided that we have $\mathcal{S} \cong \mathcal{K}(\mathcal{D})$. In this section we show that in the special case of $\mathcal{D} = D_\infty$, one can obtain a concise type theoretic description of $\mathcal{K}(\mathcal{D}_\infty)$ via a suitable natural type structure $\mathcal{S}_\infty$. Remarkably this type theory $\mathcal{S}_\infty$ is exactly the natural type structure *freely generated* by defining $\mathbb{A}^\infty = \mathcal{K}(D_0)$ and, axiomatizing $\mathcal{S}_\infty$ by postulating on $\mathbb{T}^\infty = \mathbb{T}_\cap(\mathcal{K}(\mathcal{D}_0))$ the equalities which arise from encoding the initial embedding-projection pair $\langle \mathsf{i}_0, \mathsf{j}_0 \rangle$. In this section we follow Alessi [1991], Alessi, Dezani-Ciancaglini and Honsell [2004].

We fix some notations and recall the standard D_∞ construction.

19.3.1. Definition. (i) Let D_0 be an ω-algebraic complete lattice and

$$\langle \mathsf{i}_0, \mathsf{j}_0 \rangle$$

be an *embedding-projection* pair between D_0 and $[D_0 \to D_0]$, i.e.

$$\begin{array}{rcl} \mathsf{i}_0 & : & D_0 \to [D_0 \to D_0] \\ \mathsf{j}_0 & : & [D_0 \to D_0] \to D_0 \end{array}$$

are Scott continuous maps satisfying

$$\begin{array}{rcl} \mathsf{i}_0 \circ \mathsf{j}_0 & \sqsubseteq & Id_{[D_0 \to D_0]} \\ \mathsf{j}_0 \circ \mathsf{i}_0 & = & Id_{D_0}. \end{array}$$

(ii) Define a *tower* $\langle \mathsf{i}_n, \mathsf{j}_n \rangle : D_n \to D_{n+1}$ in the following way:

- $D_{n+1} = [D_n \to D_n]$;
- $\mathsf{i}_n(f) = \mathsf{i}_{n-1} \circ f \circ \mathsf{j}_{n-1}$ for any $f \in D_n$;
- $\mathsf{j}_n(g) = \mathsf{j}_{n-1} \circ g \circ \mathsf{i}_{n-1}$ for any $g \in D_{n+1}$.

(iii) For $d \in \Pi_{n \in \mathbb{N}} D_n$ write $d_n = d(n)$. The set D_∞ is defined by

$$D_\infty = \{ d \in \Pi_{n \in \mathbb{N}} D_n \mid \forall n \in \mathbb{N}.\ d_n \in D_n \ \&\ \mathsf{j}_n(d_{n+1}) = d_n \},$$

Notation. d_n denotes the projection on D_n, while d^n is an element of D_n.

19.3.2. Lemma. *$\bigsqcup X$ exists for all $X \subseteq D_\infty$.*

PROOF. Clearly $\bigsqcup X = \boldsymbol{\lambda} n \in \mathbb{N}.d^n$, where $d^n = \sqcup\{e_n \mid e \in X\}$.

19.3.3. DEFINITION. (i) The ordering on D_∞ is given by
$d \sqsubseteq e \quad \Leftrightarrow \quad \forall k \in \mathbb{N}.\ d_k \sqsubseteq e_k.$

(ii) Let $\langle \Phi_{m\infty}, \Phi_{\infty m}\rangle$ denote the standard embedding-projection pair from D_m to D_∞ defined as follows. For any element $d^m \in D_m$, $d \in D_\infty$,

$$\Phi_{mn}(d^m) = \begin{cases} j_n(\ldots(j_{m-1}(d^m))) & \text{if } m > n \\ d^m & \text{if } m = n \\ i_{n-1}(\ldots(i_m(d^m))) & \text{if } m < n \end{cases}$$

$$\begin{aligned} \Phi_{m\infty}(d^m) &= \langle \Phi_{m1}(d^m), \Phi_{m2}(d^m)\ldots, \Phi_{mn}(d^m), \ldots\rangle \\ \Phi_{\infty m}(d) &= d_m = d(m) \end{aligned}$$

19.3.4. DEFINITION. Let

$$\begin{aligned} F_\infty &: D_\infty \to [D_\infty \to D_\infty] \\ G_\infty &: [D_\infty \to D_\infty] \to D_\infty \end{aligned}$$

be defined as follows

$$\begin{aligned} F_\infty(d) &= \bigsqcup_{n\in\mathbb{N}} (\Phi_{n\infty} \circ d_{n+1} \circ \Phi_{\infty n}); \\ G_\infty(f) &= \bigsqcup_{n\in\mathbb{N}} \Phi_{(n+1)\infty}(\Phi_{\infty n} \circ f \circ \Phi_{n\infty}). \blacksquare \end{aligned}$$

19.3.5. LEMMA. (i) $\mathsf{i}_n \circ \mathsf{j}_n \sqsubseteq Id_{[D_n \to D_n]}$, $\mathsf{j}_n \circ \mathsf{i}_n = Id_{D_n}$.
(ii) $\forall p, q \in D_n\ [\mathsf{i}_{n+1}(p \mapsto q) = (\mathsf{i}_n(p) \mapsto \mathsf{i}_n(q))\ \&$
$\mathsf{j}_{n+1}(\mathsf{i}_n(p) \mapsto \mathsf{i}_n(q)) = (p \mapsto q)]$.
(iii) *$\Phi_{m\infty} \circ \Phi_{\infty m} \sqsubseteq Id_\infty$ and $\Phi_{\infty m} \circ \Phi_{m\infty} = Id_{D_m}$.*
(iv) $\forall e \in \mathcal{K}(D_n)\ [\mathsf{i}_n(e) \in \mathcal{K}(D_{n+1})]$.
(v) $\forall e \in \mathcal{K}(D_n)\ [m \geq n \Rightarrow \Phi_{nm}(e) \in \mathcal{K}(D_m)]$.
(vi) $\forall e \in \mathcal{K}(D_n)\ [\Phi_{n\infty}(e) \in \mathcal{K}(D_\infty)]$.
(vii) *If $n \leq k \leq m$ and $d \in D_n$, $e \in D_k$, then*

$$\Phi_{nk}(d) \sqsubseteq e \ \Leftrightarrow\ \Phi_{nm}(d) \sqsubseteq \Phi_{km}(e) \ \Leftrightarrow\ \Phi_{n\infty}(d) \sqsubseteq \Phi_{k\infty}(e).$$

(viii) *$\Phi_{mn} : D_m \to D_n = \Phi_{\infty n} \circ \Phi_{m\infty}$.*
(ix) $\forall a, b \in \mathcal{D}_n\ [(\Phi_{n\infty}(a) \mapsto \Phi_{n\infty}(b)) = \Phi_{n\infty} \circ (a \mapsto b) \circ \Phi_{\infty n}]$.

PROOF. (i) follows by induction on n.
(iv) and (v) and (vi) follow from Lemma 17.4.4(ii) observing that the following pairs are all Galois connections:

1. $\langle i_n, j_n\rangle$
2. $\langle \Phi_{nm}, \Phi_{mn}\rangle$ for $n \leq m$
3. $\langle \Phi_{n\infty}, \Phi_{\infty n}\rangle$

(viii) follows from a variation of Lemma 17.4.9 (where the Z does not play a role) observing that $\langle \Phi_{n\infty}, \Phi_{\infty n}\rangle$ is a Galois connection. ■

19.3.6. LEMMA. $\bigsqcup_{n\in\mathbb{N}} \Phi_{n\infty} \circ \Phi_{\infty n} = Id_{D_\infty}$.

PROOF. Since $\langle \Phi_{n\infty}, \Phi_{\infty n}\rangle$ is an embedding-projection pair, we have for all $n\in\mathbb{N}$ $\Phi_{n\infty} \circ \Phi_{\infty n} \sqsubseteq \mathrm{Id}_{D_\infty}$, hence for all $d \in D_\infty$

$$\bigsqcup_{n\in\mathbb{N}} \Phi_{n\infty} \circ \Phi_{\infty n}(d) \sqsubseteq d.$$

On the other hand, for all $k\in\mathbb{N}$, we have

$$\begin{array}{lll} (\bigsqcup_{n\in N} \Phi_{n\infty} \circ \Phi_{\infty n}(d))_k & \sqsupseteq (\Phi_{k\infty} \circ \Phi_{\infty k}(d))_k & \text{because } (\_)_k \text{ is monotone} \\ & = \Phi_{\infty k}(d) & \text{because } (\Phi_{k\infty}(x))_k = x \text{ for all } x \\ & = d_k & \end{array}$$

Therefore also

$$\bigsqcup_{n\in\mathbb{N}} \Phi_{n\infty} \circ \Phi_{\infty n}(d) \sqsupseteq d,$$

and we are done. ■

Next lemma characterizes the compact elements of D_∞ and $[D_\infty \to D_\infty]$.

19.3.7. LEMMA. (i) $d\in\mathcal{K}(D_\infty) \Leftrightarrow \exists k, e \in \mathcal{K}(D_k).\Phi_{k\infty}(e) = d$.
(ii) $f \in \mathcal{K}([D_\infty \to D_\infty]) \Leftrightarrow \exists k, g \in \mathcal{K}(D_{k+1}).f = \Phi_{k\infty} \circ g \circ \Phi_{\infty k}$.
(iii) *If* $f = \Phi_{k\infty} \circ g \circ \Phi_{\infty k}$ *with* $g \in D_{k+1}$, *then* $G_\infty(f) = \Phi_{(k+1)\infty}(g)$.

PROOF. (i) ($\Rightarrow$) Let $d \in \mathcal{K}(D_\infty)$. Then $d = \bigsqcup_{n\in\mathbb{N}} \Phi_{n\infty}(d_n)$, by Lemma 19.3.6. Since d is compact, there exists $k \in \mathbb{N}$ such that $d = \Phi_{k\infty}(d_k)$. Now we prove that $d_k \in \mathcal{K}(D_k)$. Let $X \subseteq D_k$ be directed. Then

$$\begin{array}{lll} d_k \sqsubseteq \bigsqcup X & \Rightarrow \quad d \sqsubseteq \Phi_{k\infty}(\bigsqcup X) & \\ & \Rightarrow \quad d \sqsubseteq \bigsqcup \Phi_{k\infty}(X), & \text{since } \Phi_{k\infty} \text{ is continuous,} \\ & \Rightarrow \quad \exists x \in X. d \sqsubseteq \Phi_{k\infty}(x), & \text{for some } k \text{ since } d \text{ is compact,} \\ & \Rightarrow \quad \Phi_{\infty k}(d) \sqsubseteq \Phi_{\infty k} \circ \Phi_{k\infty}(x) & \\ & \Rightarrow \quad d_k \sqsubseteq x. & \end{array}$$

This proves that $d_k \in \mathcal{K}(D_k)$. ($\Leftarrow$) It follows from Lemma 19.3.5(vi).
(ii) ($\Rightarrow$) By Lemma 19.3.6, we have

$$f = \bigsqcup_{n\in\mathbb{N}} \Phi_{n\infty} \circ (\Phi_{\infty n} \circ f \circ \Phi_{n\infty}) \circ \Phi_{\infty n}.$$

Using similar arguments as in the proof of (i), we have that

- $\exists k \in \mathbb{N}. f = \Phi_{k\infty} \circ (\Phi_{\infty k} \circ f \circ \Phi_{k\infty}) \circ \Phi_{\infty k}$,
- $(\Phi_{\infty k} \circ f \circ \Phi_{k\infty}) \in \mathcal{K}(D_{k+1})$.

Put $g = (\Phi_{\infty k} \circ f \circ \Phi_{k\infty})$. ($\Leftarrow$) Easy.

(iii) Let $f = \Phi_{k\infty} \circ g \circ \Phi_{\infty k}$ with $g \in D_{k+1}$. Then

$$\begin{array}{rcll}
G_\infty(f) & = & G_\infty(\Phi_{k\infty} \circ g \circ \Phi_{\infty k}) & \\
& = & \bigsqcup_{n\in\mathbb{N}} \Phi_{(n+1)\infty}(\Phi_{\infty n} \circ (\Phi_{k\infty} \circ g \circ \Phi_{\infty k}) \circ \Phi_{n\infty}) & \\
& = & \bigsqcup_{n\in\mathbb{N}} \Phi_{(n+1)\infty}(\Phi_{kn} \circ g \circ \Phi_{nk}), & \text{by definition of } \Phi_{kn}, \\
& = & \bigsqcup_{n\in\mathbb{N}} \Phi_{(n+1)\infty}(\Phi_{(k+1)(n+1)}(g)), & \text{by Lemma 19.3.6}, \\
& = & \Phi_{(k+1)\infty}(g). \blacksquare &
\end{array}$$

19.3.8. Lemma. (i) $\forall x \in \mathcal{D}_\infty . x = \sqcup\{e \in \mathcal{K}(D_\infty) \mid e \sqsubseteq x\}$.
(ii) $\mathcal{K}(D_\infty)$ *is countable.*
(iii) $\mathcal{D}_\infty \in \mathbf{ALG}$.

Proof. (i) Let $x \in D_\infty$ and $U_x = \{e \in \mathcal{K}(D_\infty) \mid e \sqsubseteq x\}$. Clearly, $\bigsqcup U_x \sqsubseteq x$. Now let $f = \bigsqcup U_x$ in order to show $x \sqsubseteq f$. By definition of sup in D_∞ we have that

$$f_n = \bigsqcup V(n,x) \quad \text{where } V(n,x) = \{e_n \in D_n \mid e \in \mathcal{K}(D_\infty) \& e \sqsubseteq x\}$$

Since D_n is algebraic, we have that

$$x_n = \bigsqcup W(n,x) \quad \text{where } W(n,x) = \{d \in \mathcal{K}(D_n) \mid d \sqsubseteq x_n\}$$

We will prove that $W(n,x) \subseteq V(n,x)$. Suppose $d \in W(n,x)$. Then $d \in \mathcal{K}(D_n)$ and $d \sqsubseteq x_n$. Let $e = \Phi_{n\infty}(d)$. Then,

(1) $d = \Phi_{\infty n} \circ \Phi_{n\infty}(d) = e_n$

(2) $e = \Phi_{n\infty}(d) \in \mathcal{K}(D_\infty)$ by Lemma 19.3.5(vi)

(3) $e = \Phi_{n\infty}(d) \sqsubseteq \Phi_{n\infty}(x_n) \sqsubseteq x$ by monotonicity of $\Phi_{n\infty}$ and Lemma 19.3.5(iii)

Hence $d \in V(n,x)$. Clearly, $x_n \sqsubseteq f_n$. Hence $x \sqsubseteq f$.

(ii) By Proposition 17.1.11 one has $D_n \in \mathbf{ALG}$ for each n. Hence $\mathcal{K}(D_n)$ is countable for each n. But then also $\mathcal{K}(D_\infty)$ is countable, by Lemma 19.3.7(i).

(iii) By (i), (ii) and (iii). $\blacksquare$

19.3.9. Theorem. (Scott [1972]) *Let D_∞ be constructed from $D_0 \in \mathbf{ALG}$ and a projection pair $\mathsf{i}_0, \mathsf{j}_0$. Then $D_\infty \in \mathbf{ALG}$ and D_∞ with F_∞, G_∞ is reflexive. Moreover,*

$$F_\infty \circ G_\infty = Id_{[D_\infty \to D_\infty]} \;\; \& \;\; G_\infty \circ F_\infty = Id_{D_\infty}.$$

It follows that D_∞ is an extensional λ-model.

Proof. See Barendregt [1984] Theorem 18.2.16 for the proof that F and G are each other's inverse. Therefore, by Proposition 18.1.10(ii), D_∞ is an extensional λ-model. $\blacksquare$

19.3.10. Corollary. *Let D_∞ be constructed from $D_0 \in \mathbf{ALG}$ and a projection pair $\mathsf{i}_0, \mathsf{j}_0$. Then $\langle D_\infty, F_\infty, G_\infty\rangle$ is in* $\mathbf{NLS}$.

Proof. Immediate from the Theorem. $\blacksquare$

D_∞ as a filter λ-model

Let D_∞ be constructed from the triple $t = (D_0, \mathsf{i}_0, \mathsf{j}_0)$. To emphasize the dependency on t we write $D_\infty = D^t_\infty$. From the previous corollary and applying various results of Chapter 17 it follows that $D^t_\infty \cong \mathcal{F}^{\mathcal{K}(D^t_\infty)}$. In this subsection we associate with $t = (D_0, \mathsf{i}_0, \mathsf{j}_0)$ a rather simple intersection type structure $\mathcal{S}^t_\infty$ such that $\mathcal{S}^t_\infty \cong \mathcal{K}(D^t_\infty)$, hence

$$D^t_\infty \cong \mathcal{F}^{\mathcal{S}^t_\infty}.$$

Later in Proposition 19.3.27 we will prove for $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{HR}\}$ that

$$\mathcal{S} \cong \mathcal{S}^t_\infty,$$

and hence $D^{\mathcal{S}}_\infty \cong \mathcal{F}^{\mathcal{S}^t_\infty} = \mathcal{F}^{\mathcal{S}}$.

19.3.11. Definition. (Definition of $\mathcal{S}^t_\infty$ and $\leq_\infty$) Let $t = (D_0, \mathsf{i}_0, \mathsf{j}_0)$ be given.

(i) The partial order $\leq_0$ on $\mathcal{K}(D_0)$ is defined by

$$d \leq_0 e \Leftrightarrow d \sqsupseteq e \quad d, e \in \mathcal{K}(D_0);$$

(ii) $\mathbb{T}^\infty = \mathcal{K}(D_0) \mid \mathbb{T}^\infty \!\to\! \mathbb{T}^\infty \mid \mathbb{T}^\infty \cap \mathbb{T}^\infty$.

(iii) Write $\top$ for $\perp_{D_0}$. Let $\leq_\infty$ be the lazy type theory on $\mathbb{T}^\infty$, i.e. having the axiom and rules of Definition 15.1.1 plus $(\to\cap), (\to), (\top)$ and $(\top \to)$ with as extra axiom

$$c \cap d =_\infty c \sqcup d$$

where $\sqcup$ is the lub for the ordering $\sqsubseteq$ on $\mathcal{K}(D_0)$
and the extra rules

$$c \leq_0 d \;\Rightarrow\; c \leq_\infty d$$

and

$$i_o(e) = (c_1 \mapsto d_1) \sqcup \ldots \sqcup (c_n \mapsto d_n) \;\Rightarrow\; e =_\infty (c_1 \to d_1) \cap \ldots \cap (c_n \to d_n),$$

where $c, d, e, , c_1, d_1, \ldots, c_n, d_n \in \mathcal{K}(D_0)$.

(iv) $\mathcal{S}^t_\infty = \mathbb{T}^\infty / =_\infty$. So we have $\mathcal{S}^t_\infty \in$ **NTS**. ∎

19.3.12. Remark. The fact that $w_n(A) \in \mathcal{K}(D^t_n)$ for all $A \in \mathbb{T}^\infty_n$ follows from Lemma 17.1.9.

The proof of the next lemma follows easily from Definition 19.3.11.

19.3.13. Lemma. $c_1 \cap \ldots \cap c_n =_\infty c_1 \sqcup \ldots \sqcup c_n$. ∎

The proof of $\mathcal{S}^t_\infty \cong \mathcal{K}(D^t_\infty)$ will occupy 19.3.16-19.3.18. First we classify the types in $\mathbb{T}^\infty$ according to the maximal number of nested arrow occurrences they may contain.

19.3.14. DEFINITION. (i) We define the map *rank* $rk : \mathbb{T}^\infty \to \mathbb{N}$ by:

$$\begin{array}{rcll} rk(c) & = & 0, & \text{for } c \in \mathcal{K}(D_0); \\ rk(A \to B) & = & \max\{rk(A), rk(B)\} + 1; & \\ rk(A \cap B) & = & \max\{rk(A), rk(B)\}. & \end{array}$$

(ii) Let $\mathbb{T}^\infty_n = \{A \in \mathbb{T}^\infty \mid rk(A) \leq n\}$. ■

Remark that $w_n(A) \in \mathcal{K}(D^t_n)$ for all $A \in \mathbb{T}^\infty_n$ by Lemma 17.1.9.

We can associate to each type in $\mathbb{T}^\infty_n$ an element in D_n: this will be crucial for defining the required isomorphism (see Definition 19.3.20).

19.3.15. DEFINITION. We define, for each $n \in \mathbb{N}$, a map $w_n : \mathbb{T}^\infty_n \to \mathcal{K}(D^t_n)$ by a double induction on n and on the construction of types in $\mathbb{T}^\infty$:

$$\begin{array}{lcl} w_n(c) & = & \Phi_{0n}(c); \\ w_n(A \cap B) & = & w_n(A) \sqcup w_n(B); \\ w_n(A \to B) & = & (w_{n-1}(A) \mapsto w_{n-1}(B)). \ ■ \end{array}$$

19.3.16. LEMMA. *Let $n \leq m$ and $A \in \mathbb{T}^\infty_n$. Then $\Phi_{m\infty}(w_m(A)) = \Phi_{n\infty}(w_n(A))$.*

PROOF. We show by induction on the definition of w_n that $w_{n+1}(A) = \mathsf{i}_n(w_n(A))$. Then the desired equality follows from the definition of the function Φ. The only interesting case is when $A \equiv B \to C$. We get

$$\begin{array}{rcll} w_{n+1}(B \to C) & = & w_n(B) \mapsto w_n(C), & \text{by definition,} \\ & = & \mathsf{i}_{n-1}(w_{n-1}(B)) \mapsto \mathsf{i}_{n-1}(w_{n-1}(C)), & \text{by induction,} \\ & = & \mathsf{i}_n(w_{n-1}(B) \mapsto w_{n-1}(C)), & \text{by Lemma 19.3.5(ii),} \\ & = & \mathsf{i}_n(w_n(B \to C)), & \text{by Definition 19.3.15.} \ ■ \end{array}$$

The maps w_n reverse the order between types.

19.3.17. LEMMA. *Let $rk(A \cap B) \leq n$. Then*

$$A \leq_\infty B \ \Rightarrow \ w_n(B) \sqsubseteq w_n(A).$$

PROOF. The proof is by induction on the definition of $\leq_\infty$. We consider only two cases.

Case $(\to)$. Let $A \leq_\infty B$ because $A \equiv C \to D$, $B \equiv E \to F$, $E \leq_\infty C$ and $D \leq_\infty F$. Then

$$\begin{array}{rcl} E \leq_\infty C \ \& \ D \leq_\infty F & \Rightarrow & w_{n-1}(C) \sqsubseteq w_{n-1}(E) \ \& \ w_{n-1}(F) \sqsubseteq w_{n-1}(D), \\ & & \text{by induction} \\ & \Rightarrow & w_{n-1}(E) \mapsto w_{n-1}(F) \sqsubseteq w_{n-1}(C) \mapsto w_{n-1}(D) \\ & \Rightarrow & w_n(B) \sqsubseteq w_n(A). \end{array}$$

Case $e =_\infty (c_1 \to d_1) \cap \ldots \cap (c_k \to d_k)$ as $i_0(e) = (c_1 \mapsto d_1) \sqcup \ldots \sqcup (c_k \mapsto d_k)$. We show by induction on $n \geq 1$ the following.

$$w_n(e) = (w_{n-1}(c_1) \mapsto w_{n-1}(d_1)) \sqcup \ldots \sqcup (w_{n-1}(c_k) \mapsto w_{n-1}(d_k)).$$

It trivially holds for $n = 1$, so let $n > 1$.

$$\begin{array}{rcl} w_n(e) & = & \mathsf{i}_{n-1}(w_{n-1}(e)) \\ & = & \mathsf{i}_{n-1}((w_{n-2}(c_1) \mapsto w_{n-2}(d_1)) \sqcup \ldots \sqcup (w_{n-2}(c_k) \mapsto w_{n-2}(d_k))) \\ & = & \mathsf{i}_{n-1}(w_{n-2}(c_1) \mapsto w_{n-2}(d_1)) \sqcup \ldots \sqcup \mathsf{i}_{n-1}(w_{n-2}(c_k) \mapsto w_{n-2}(d_k)) \\ & = & (\mathsf{i}_{n-2}(w_{n-2}(c_1)) \mapsto \mathsf{i}_{n-2}(w_{n-2}(d_1))) \sqcup \ldots \sqcup (\mathsf{i}_{n-2}(w_{n-2}(c_k)) \mapsto \mathsf{i}_{n-2}(w_{n-2}(d_k)) \\ & = & (w_{n-1}(c_1) \mapsto w_{n-1}(d_1)) \sqcup \ldots \sqcup (w_{n-1}(c_k) \mapsto w_{n-1}(d_k)). \blacksquare \end{array}$$

Also the reverse implication of Lemma 19.3.17 holds.

19.3.18. Lemma. *Let $rk(A \cap B) \leq n$. Then*

$$w_n(B) \sqsubseteq w_n(A) \;\Rightarrow\; A \leq_\infty B.$$

Proof. By induction on $rk(A \cap B)$.
If $rk(A \cap B) = 0$ we have $A \equiv \bigcap_{i \in I} c_i$, $B = \bigcap_{j \in J} d_j$. Then

$$\begin{array}{rcl} w_n(B) \sqsubseteq w_n(A) & \Rightarrow & \bigsqcup_{j \in J} \Phi_{0n}(d_j) \sqsubseteq \bigsqcup_{i \in I} \Phi_{0n}(c_i) \\ & \Rightarrow & \Phi_{n0}(\bigsqcup_{j \in J} \Phi_{0n}(d_j)) \sqsubseteq \Phi_{n0}(\bigsqcup_{i \in I} \Phi_{0n}(c_i)) \\ & \Rightarrow & \bigsqcup_{j \in J} (\Phi_{n0} \circ \Phi_{0n})(d_j) \sqsubseteq \bigsqcup_{i \in I} (\Phi_{n0} \circ \Phi_{0n})(c_i) \\ & \Rightarrow & \bigsqcup_{j \in J} d_j \sqsubseteq \bigsqcup_{i \in I} c_i \\ & \Rightarrow & A \leq_\infty B. \end{array}$$

Otherwise, let

$$\begin{array}{rcl} A & \equiv & (\bigcap_{i \in I} c_i) \cap (\bigcap_{l \in L} (C_l \to D_l)), \\ B & \equiv & (\bigcap_{h \in H} d_h) \cap (\bigcap_{m \in M} (E_m \to F_m)). \end{array}$$

By Proposition 17.1.11 we have

$$c_i =_\infty \bigcap_{j \in J_i} (a_j \to b_j), \quad d_h =_\infty \bigcap_{k \in K_h} (e_k \to f_k),$$

where $a_j, b_j, e_k, f_k \in \mathcal{K}(D_0)$. Now for all $n \geq 1$

$$\begin{array}{rcl} w_n(c_i) & = & \bigsqcup_{j \in J_i} (w_{n-1}(a_j) \mapsto w_{n-1}(b_j))), \\ w_n(d_h) & = & (\bigsqcup_{k \in K_h} (w_{n-1}(e_k) \mapsto w_{n-1}(f_k))), \end{array}$$

since by Lemma 19.3.17 the function w_n identifies elements in the equivalence classes of $=_\infty$. So we get

$$\begin{array}{l}\bigsqcup_{h\in H}(\bigsqcup_{k\in K_h} w_{n-1}(e_k)\mapsto w_{n-1}(f_k))\sqcup(\bigsqcup_{m\in M} w_{n-1}(E_m)\mapsto w_{n-1}(F_m)) \quad\sqsubseteq\\ \qquad \bigsqcup_{i\in I}(\bigsqcup_{j\in J_i} w_{n-1}(a_j)\mapsto w_{n-1}(b_j))\sqcup(\bigsqcup_{l\in L} w_{n-1}(C_l)\mapsto w_{n-1}(D_l)).\end{array}$$

Hence for each $h\in H$, $k\in K_h$ we have

$$\begin{array}{rcl}(w_{n-1}(e_k)\mapsto w_{n-1}(f_k)) & \sqsubseteq & \bigsqcup_{i\in I}(\bigsqcup_{j\in J_i} w_{n-1}(a_j)\mapsto w_{n-1}(b_j))\sqcup\\ & & (\bigsqcup_{l\in L} w_{n-1}(C_l)\mapsto w_{n-1}(D_l)).\end{array}$$

Suppose $w_{n-1}(f_k)\neq\bot_{D_n}$. Then by Lemma 17.1.10 there exist $I'\subseteq I$, $J_i'\subseteq J_i$, $L'\subseteq L$ such that

$$\begin{array}{rcl}\bigsqcup_{i\in I'}(\bigsqcup_{j\in J_i'} w_{n-1}(a_j))\sqcup(\bigsqcup_{l\in L'} w_{n-1}(C_l)) & \sqsubseteq & w_{n-1}(e_k),\\ \bigsqcup_{i\in I'}(\bigsqcup_{j\in J_i'} w_{n-1}(b_j))\sqcup(\bigsqcup_{l\in L'} w_{n-1}(D_l)) & \sqsupseteq & w_{n-1}(f_k).\end{array}$$

Notice that all types involved in the two above judgments have ranks strictly less than $rk(A\cap B)$:

1. the rank of a_j, b_j, e_k, f_k is 0, since they are all constants in $\mathcal{K}(D_0)$ and that
2. the rank of C_l, D_l is stricly smaller than the one of $A\cap B$, since they are subterms of an arrow in A.

Then by induction and by Lemma 19.3.13 we obtain

$$\begin{array}{rcl}e_k & \leq_\infty & \bigcap_{i\in I'}(\bigcap_{j\in J_i'} a_j)\cap\bigcap_{l\in L'} C_l,\\ f_k & \geq_\infty & \bigcap_{i\in I'}(\bigcap_{j\in J_i'} b_j)\cap\bigcap_{l\in L'} D_l.\end{array}$$

Therefore we have by $(\to)$ and Proposition 15.1.13 $A\leq_\infty e_k\to f_k$.
If $w_{n-1}(f_k) = \bot_{D_n}$, then $w_{n-1}(f_k) = \Phi_{0n}(f_k)$ since $f_k\in\mathcal{K}(D_0)$. This gives $f_k = \Phi_{n0}\circ\Phi_{0n}(f_k) = \Phi_{n0}(\bot_{D_n}) = \bot_{D_0}$ because $j_n(\bot_{D_{n+1}}) = \bot_{D_n}$. We conclude since $f_k = \bot_{D_0}$ implies $A\leq_\infty e_k\to f_k$.
In a similar way we can prove that $A\leq_\infty E_m\to F_m$, for any $m\in M$. Putting together these results we get $A\leq_\infty B$. ■

19.3.19. Proposition. *$\langle\mathcal{K}(D^t_\infty),\to_\infty,\cap,\top\rangle$ is a natural type structure where $a\to_\infty b = G_\infty(a\mapsto b)$, $\cap$ is the least upper bound of D^t_∞ and $\top$ is the bottom of D^t_∞.*

Proof. By Theorem 19.4.10 we have that $\Xi_\infty = \langle D^t_\infty, F_\infty, G_\infty\rangle$ is a natural lambda structure. By Proposition 17.4.29, $\mathcal{L}(\Xi_\infty) = \langle D^t_\infty, Z_\infty\rangle$ is a natural zip structure, where $Z_\infty(a, b) = G_\infty(a{\mapsto}b)$. Using Proposition 17.3.13, we know that $\mathsf{Cmp} \circ \mathcal{L}(\Xi_\infty) = \langle \mathcal{K}(D^t_\infty), \to_\infty, \cap, \top\rangle$ is a natural type structure, where the arrow $\to_\infty$ is given by Z_∞.

We can now prove the isomorphism in **NTS** between $\mathcal{K}(\mathcal{D}_\infty)$ and $\mathcal{S}_\infty$

19.3.20. Definition. For $A \in \mathbb{T}^\infty$ write

$$\mathsf{m}([A]) = \Phi_{r\infty}(w_r(A)),$$

where $r \geq rk(A)$.

19.3.21. Theorem. *In* **NTS** *one has* $\mathcal{S}^t_\infty \cong \mathcal{K}(D^t_\infty)$ *via* m.

Proof. First of all notice that m is well defined, in the sense the it does not depend on either the type chosen in $[A]$ or the rank r. In fact let $B, B' \in [A]$, and let $p \geq rk(B)$, $p' \geq rk(B')$. Fix any $q \geq p, p'$. Then we have

$$\begin{array}{rcll}
\Phi_{p\infty}(w_p(B)) & = & \Phi_{q\infty}(w_q(B)), & \text{by Lemma 19.3.16,}\\
 & = & \Phi_{q\infty}(w_q(B')), & \text{by Lemma 19.3.17,}\\
 & = & \Phi_{p'\infty}(w_{p'}(B')), & \text{by Lemma 19.3.16.}
\end{array}$$

Write $\mathsf{m}(A)$ for $\mathsf{m}([A])$. We prove that m satisfies all the conditions of Proposition ??. m is injective by Lemma 19.3.18 and monotone by Lemma 19.3.17.

From Lemma 17.1.11(ii) we get immediately

$$\mathcal{K}(D_{n+1}) = \{c_1{\mapsto}d_1 \sqcup \ldots \sqcup c_n{\mapsto}d_n | c_i, d_i \in \mathcal{K}(D_n)\}.$$

it is easily proved by induction on n that w_n is surjective on $\mathcal{K}(D_n)$, hence m is surjective by Lemma 19.3.7(i). The function m^{-1} is monotone by Lemma 19.3.18. Finally, we have to prove that m satisfies condition (??) of Proposition ??. Taking into account that the order $\leq_\infty$ on $\mathcal{K}(D^t_\infty)$ is the reversed of $\sqsubseteq$ of D^t_∞ and that by Definition ??, we have $d{\to_\infty}e = G_\infty(d{\mapsto}e)$, we need to show:

$$A \leq_\infty B \to C \;\;\Leftrightarrow\;\; \mathsf{m}(A) \sqsupseteq G_\infty(\mathsf{m}(B){\mapsto}\mathsf{m}(C)) \tag{19.1}$$

In order to prove (19.1), let $r \geq max\{rk(A), rk(B \to C)\}$ (in particular it follows $rk(B), rk(C) \leq r - 1$). We have

$$\begin{array}{rcl}
G_\infty(\mathsf{m}(B){\mapsto}\mathsf{m}(C)) & = & G_\infty(\Phi_{(r-1)\infty}(w_{r-1}(B)){\mapsto}\Phi_{(r-1)\infty}(w_{r-1}(C)))\\
 & = & G_\infty(\Phi_{(r-1)\infty} \circ (w_{r-1}(B){\mapsto}w_{r-1}(C)) \circ \Phi_{\infty(r-1)}),\\
 & & \text{by Lemma 19.3.5(viii),}\\
 & = & \Phi_{r\infty}(w_{r-1}(B){\mapsto}w_{r-1}(C)), \quad \text{by Lemma 19.3.7(iii),}\\
 & = & \Phi_{r\infty}(w_r(B \to C)), \quad \text{by definition of } w_r.
\end{array}$$

Finally we have

$$\begin{array}{rcll} A \leq_\infty B \to C & \Leftrightarrow & w_r(A) \sqsupseteq w_r(B \to C), & \\ & & \text{by Lemmas 19.3.17 and 19.3.18,} & \\ & \Leftrightarrow & \Phi_{r\infty}(w_r(A)) \sqsupseteq \Phi_{r\infty}(w_r(B \to C)) & \\ & & \text{since } \Phi_{r\infty} \text{ is an embedding} & \\ & \Leftrightarrow & \Phi_{r\infty}(w_r(A)) \sqsupseteq G_\infty(\mathsf{m}(B) \mapsto \mathsf{m}(C)), & \text{as above,} \\ & \Leftrightarrow & \mathsf{m}(A) \sqsupseteq G_\infty(\mathsf{m}(B) \mapsto \mathsf{m}(C)). & \end{array}$$

So we have proved (19.1) and the proof is complete. ■

19.3.22. THEOREM. *$\mathcal{F}^{\mathcal{S}^t_\infty} \cong D^t_\infty$ in* **NLS**, *via the map*

$$\underline{\mathsf{m}}(X) = \bigsqcup\{\mathsf{m}(B) \mid B \in X\}$$

satisfying $\underline{\mathsf{m}}(\uparrow A) = \mathsf{m}(A)$.

PROOF. One has $\mathcal{F}^{\mathcal{S}^t_\infty} \cong \mathcal{F}^{\mathcal{K}(D^t_\infty)} \cong D^t_\infty$, by Theorems 19.3.21 and ??(i). Notice that if $\mathcal{S} \cong \mathcal{S}'$ via the map m, then also via the approximable map μ_{m} and hence $\mathcal{F}^{\mathcal{S}} \cong \mathcal{F}^{\mathcal{S}'}$ via the map

$$\mathsf{Flt}(\mu_{\mathsf{m}})(X) = \{B' \mid \exists B \in X. B \mu_{\mathsf{m}} B'\}.$$

Therefore by Theorem 19.3.21 the map

$$\overline{\mu_{\mathsf{m}}} : \mathcal{F}^{\mathcal{S}^t_\infty} \to \mathcal{F}^{\mathcal{K}(D^t_\infty)}$$

is an isomorphism. By Proposition ?? the map

$$\bigsqcup : \mathcal{F}^{\mathcal{K}(D^t_\infty)} \to D^t_\infty$$

is an isomorphism. Therefore the required isomorphism is

$$\begin{array}{rcl} \underline{\mathsf{m}}(X) & = & \bigsqcup \mathsf{Flt}\mu_{\mathsf{m}}(X) \\ & = & \bigsqcup\{B' \mid \exists B \in X. B \mu_{\mathsf{m}} B'\} \\ & = & \bigsqcup\{B' \mid \exists B \in X. \mathsf{m}(B) \leq B'\} \\ & = & \bigsqcup\{B' \mid \exists B \in X. B' \sqsubseteq \mathsf{m}(B)\} \\ & = & \bigsqcup\{\mathsf{m}(B) \mid B \in X\}. \end{array}$$

In particular $\underline{\mathsf{m}}(\uparrow A) = \bigsqcup\{\mathsf{m}(B) \mid B \sqsupseteq A\} = \mathsf{m}(A)$. Let $\mathsf{m} : S_\infty \to \mathcal{K}(D_\infty)$ be the isomorphism in NTS. By Proposition 17.3.14 we know that Flt is a functor from NTS to NLS. Then

$$\mathsf{Flt}(\mathsf{m}) : \mathcal{F}^{S_\infty} \to \mathcal{F}^{\mathcal{K}(D_\infty)}$$

is an isomorphism in NLS where $\mathsf{Flt}(\mathsf{m})(X) = \{B \mid \exists A \in X. \mathsf{m}(A) \sqsubseteq B\}$.

By Proposition 17.3.15(ii) we have that

$$\bigodot : \mathcal{F}^{\mathcal{K}(D_\infty)} \to D_\infty$$

is an isomorphism in NLS where $\circleddash(X) = \sqcup X$.

The composition of $\circleddash$ and $\mathsf{Flt}(\mathsf{m})$ is an isomorpshim from $\mathcal{F}^{\mathcal{S}_\infty}$ to D_∞ explicitley given by

$$\circleddash \circ \mathsf{Flt}(\mathsf{m})(X) = \sqcup\{B \mid \exists A \in X.\mathsf{m}(A) \sqsubseteq B\} \sqcup \{\mathsf{m}(A) \mid A \in X\}$$

∎

Specific models D_∞ as filter models

In this subsection we specialize Theorem 19.3.22 to D^t_∞ models constructed from specific triples $t = (D_0, \mathsf{i}_0, \mathsf{j}_0)$, five in total, each satisfying a specific modeltheoretic property. For each of these the corresponding type structure $\mathcal{S}^t_\infty$ can be described in a relatively simple way. This will be done as follows. Five type structures $\mathcal{S}$ will be defined with $\mathcal{S} \in \{\text{Scott}, \text{Park}, \text{CDZ}, \text{DHM}, \text{HR}\}$. Also five corresponding choices of the triple $t_\mathcal{S} = (D_0, \mathsf{i}_0, \mathsf{j}_0)$ will be defined, giving rise to $D^{t_\mathcal{S}}_\infty$ for $\mathcal{S} \in \{\text{Scott}, \text{Park}, \text{CDZ}, \text{DHM}, \text{HR}\}$. Moreover, we will show that $\mathcal{S}^{t_\mathcal{S}}_\infty = \mathcal{S}$, hence

$$D^{t_\mathcal{S}}_\infty \cong \mathcal{F}^{\mathcal{S}^{t_\mathcal{S}}_\infty} \cong \mathcal{F}^\mathcal{S},$$

by Theorem 19.3.22. We will write $\mathcal{F}^\mathcal{S} := \mathcal{F}^{\mathcal{S}_\mathcal{S}}$, $D^\mathcal{S}_\infty := D^{t_\mathcal{S}}_\infty$ and $\mathcal{S}^\mathcal{S}_\infty := \mathcal{S}^{t_\mathcal{S}}_\infty$. Then the equation becomes

$$D^\mathcal{S}_\infty \cong \mathcal{F}^{\mathcal{S}^\mathcal{S}_\infty} \cong \mathcal{F}^\mathcal{S}.$$

19.3.23. Definition. (i) Remember the following type theoretic axioms.

(ω_{Scott})	$(\top{\to}\omega) = \omega$
(ω_{Park})	$(\omega{\to}\omega) = \omega$
$(\omega\varphi)$	$\omega \leq \varphi$
$(\varphi{\to}\omega)$	$(\varphi{\to}\omega) = \omega$
$(\omega{\to}\varphi)$	$(\omega{\to}\varphi) = \varphi$
(I)	$(\varphi{\to}\varphi) \cap (\omega{\to}\omega) = \varphi$

(ii) In Definition 15.1.7 the natural type structures[1] $\mathcal{S} \in \{\text{Scott}, \text{Park}, \text{CDZ}, \text{DHM}, \text{HR}\}$ have been defined by specifying its set of constants $\mathbb{A}^\mathcal{S}$ and some extra axioms (besides the axioms $({\to}\cap)$, $(\top)$ and $(\top\to)$ and the rule $(\to)$).

$\mathcal{S}$	Constants $\mathbb{A}^\mathcal{S}$	Axioms of $\mathcal{S}$
Scott	$\{\top, \omega\}$	(ω_{Scott})
Park	$\{\top, \omega\}$	(ω_{Park})
CDZ	$\{\top, \omega, \varphi\}$	$(\omega\varphi), (\varphi{\to}\omega), (\omega{\to}\varphi)$
HR	$\{\top, \omega, \varphi\}$	$(\omega\varphi), (\varphi{\to}\omega), (I)$
DHM	$\{\top, \omega, \varphi\}$	$(\omega\varphi), (\omega{\to}\varphi), (\omega_{\text{Scott}})$

[1]To be precise Definition 15.1.7 introduces compatible type theories, but we can view them as structures exploiting Remark 15.4.7.

(iii) As usual we write $\mathcal{S}$ for $\mathbb{T}^{\mathcal{S}}/=_{\mathcal{S}}$.

19.3.24. DEFINITION. (i) For $\mathcal{S}\in\{\text{Scott},\text{Park},\text{CDZ},\text{DHM},\text{HR}\}$ the following triples $(D_0,\mathsf{i}_0,\mathsf{j}_0)$ are defined. For $\mathcal{S}\in\{\text{Scott},\text{Park}\}$, we define $D_0^{\mathcal{S}}$ as the two point chain $\{\top\sqsubseteq\omega\}$. For $\mathcal{S}\in\{\text{CDZ},\text{HR},\text{DHM}\}$, we define $D_0^{\mathcal{S}}$ as the three point chain $\{\top\sqsubseteq\varphi\sqsubseteq\omega\}$.

(ii) We define $\mathsf{i}_0^{\mathcal{S}}:D_0^{\mathcal{S}}\to[D_0^{\mathcal{S}}\to D_0^{\mathcal{S}}]$ and $\mathsf{j}_0^{\mathcal{S}}:[D_0^{\mathcal{S}}\to D_0^{\mathcal{S}}]\to D_0^{\mathcal{S}}$ as follows. First $\mathsf{i}_0^{\mathcal{S}}$.

$$
\begin{array}{rcl}
\mathsf{i}_0^{\mathcal{S}}(\top) &=& \top\mapsto\top \quad \text{for any } \mathcal{S}\in\{\text{Scott},\text{Park},\text{CDZ},\text{DHM},\text{HR}\}\\
\mathsf{i}_0^{\mathcal{S}}(\varphi) &=& \left\{\begin{array}{ll}(\varphi\mapsto\varphi)\sqcup(\omega\mapsto\omega) & \text{if } \mathcal{S}\in\{\text{HR}\}\\ \omega\mapsto\varphi & \text{if } \mathcal{S}\in\{\text{CDZ},\text{DHM}\}\end{array}\right.\\
\mathsf{i}_0^{\mathcal{S}}(\omega) &=& \left\{\begin{array}{ll}\top\mapsto\omega & \text{if } \mathcal{S}\in\{\text{Scott},\text{DHM}\}\\ \omega\mapsto\omega & \text{if } \mathcal{S}\in\{\text{Park}\}\\ \varphi\mapsto\omega & \text{if } \mathcal{S}\in\{\text{CDZ},\text{HR}\}\end{array}\right.
\end{array}
$$

Then we define, for all $\mathcal{S}$, and $f\in[D_0^{\mathcal{S}}\to D_0^{\mathcal{S}}]$

$$\mathsf{j}_0^{\mathcal{S}}(f) \;=\; \bigsqcup\{d\in D_0^{\mathcal{S}}\mid \mathsf{i}_0^{\mathcal{S}}(d)\sqsubseteq f\}.$$

(iii) Write $t_{\mathcal{S}}=(D_0^{\mathcal{S}},\mathsf{i}_0^{\mathcal{S}},\mathsf{j}_0^{\mathcal{S}})$. It is easy to prove that $\langle\mathsf{i}_0^{\mathcal{S}},\mathsf{j}_0^{\mathcal{S}}\rangle$ is an embedding-projection pair from $D_0^{\mathcal{S}}$ to $[D_0^{\mathcal{S}}\to D_0^{\mathcal{S}}]$, so we can build $D_\infty^{\mathcal{S}}=D_\infty^{t_{\mathcal{S}}}$ following the steps outlined in Definition 19.3.1. ∎

19.3.25. LEMMA. *Let* $\mathcal{S}\in\{\text{Scott},\text{Park},\text{CDZ},\text{DHM},\text{HR}\}$ *and* $c_1,\dots,c_n,d_1,\dots,d_n,e_1,\dots,e_k,f_1,\dots,f_k\in\mathcal{D}_0^{\mathcal{S}}$. *Then*

$$
\begin{array}{l}
(e_1\to f_1)\cap\dots\cap(e_k\to f_k)=_{\mathcal{S}}(c_1\to d_1)\cap\dots\cap(c_n\to d_n)\;\Leftrightarrow\\
(c_1\mapsto d_1)\sqcup\dots\sqcup(c_n\mapsto d_n)=(e_1\mapsto f_1)\sqcup\dots\sqcup(e_k\mapsto f_k).
\end{array}
$$

PROOF. It suffices to prove

$$(c\mapsto d)\sqsubseteq(e_1\mapsto f_1)\sqcup\dots\sqcup(e_k\mapsto f_k)\;\Leftrightarrow\;(e_1\to f_1)\cap\dots\cap(e_k\to f_k)\leq_{\mathcal{S}}(c\to d).$$

Now, $(c\mapsto d)\sqsubseteq(e_1\mapsto f_1)\sqcup\dots\sqcup(e_k\mapsto f_k)\;\Leftrightarrow$

$$
\begin{array}{lll}
\Leftrightarrow & \exists I\subseteq\{1,\dots,k\}\,[\sqcup_{i\in I}e_i\sqsubseteq c\;\&\;d\sqsubseteq\sqcup_{i\in I}f_i], & \text{by Lemma 17.1.10,}\\
\Leftrightarrow & \exists I\subseteq\{1,\dots,k\}\,[c\leq_{\mathcal{S}}\cap_{i\in I}e_i\;\&\;\cap_{i\in I}f_i\leq_{\mathcal{S}}d], & \\
\Leftrightarrow & (e_1\to f_1)\cap\dots\cap(e_k\to f_k)\leq_{\mathcal{S}}(c\to d), & \text{by }\beta\text{-soundness, }(\to)\\
& & \text{and }(\to\cap).\;\blacksquare
\end{array}
$$

19.3.26. COROLLARY. *The definition of* $\mathsf{i}_0^{\mathcal{S}}$ *is canonical. By this we mean that we could have given equivalently the following definition.*

$$\mathsf{i}_o^{\mathcal{S}}(e)=(c_1\mapsto d_1)\sqcup\dots\sqcup(c_n\mapsto d_n)\;\Leftrightarrow\;e=_{\mathcal{S}}(c_1\to d_1)\cap\dots\cap(c_n\to d_n).$$

PROOF. Immediate, by the definition of $\mathsf{i}_0^{\mathcal{S}}$, the axioms $(\top)$ and $(\top\to)$, the special axioms $(\omega_{\text{Scott}}),(\omega_{Park}),(\varphi\to\omega),(\omega\to\varphi),(I)$ respectively, and the previous Lemma. ∎

By Theorem 19.3.22 we get $\mathcal{F}^{\mathcal{S}^{t_\mathcal{S}}_\infty} \cong D^{t_\mathcal{S}}_\infty$ for these five cases, where $t_\mathcal{S}$ is the triple $t_\mathcal{S} = (D^\mathcal{S}_0, \mathsf{i}^\mathcal{S}_0, \mathsf{j}^\mathcal{S}_0)$. We abbreviate this as $\mathcal{F}^{\mathcal{S}^{t_\mathcal{S}}_\infty} \cong D^\mathcal{S}_\infty$.

19.3.27. Proposition. *For* $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{HR}\}$ *one has*

$$\mathcal{S} \cong \mathcal{S}^{t_\mathcal{S}}_\infty.$$

Proof. Remember that

$$\begin{array}{rcl}\mathcal{S} & = & \mathbb{T}(\mathbb{A}^\mathcal{S})/=_\mathcal{S} \\ \mathcal{S}^{t_\mathcal{S}}_\infty & = & \mathbb{T}(\mathcal{K}(D^\mathcal{S}_0))/=_\infty .\end{array}$$

Then $\mathbb{T}(\mathbb{A}^\mathcal{S}) = \mathbb{T}(D^\mathcal{S}_0) = \mathbb{T}(\mathcal{K}(D^\mathcal{S}_0))$ for all $\mathcal{S}$, since each $D^\mathcal{S}_0$ only contains compact elements. It remains to show that $=_\infty$ and $=_\mathcal{S}$ are the same. This follows from

$$A \leq_\infty B \;\Leftrightarrow\; A \leq_\mathcal{S} B.$$

As to $(\Rightarrow)$, this follows by induction on the generation of $\leq_\infty$. Now $\mathcal{S}$ satisfies the axioms $(\rightarrow\cap), (\top)$ and is closed under the rule $(\rightarrow)$, since it is a natural type structure. It remains to show that the new axioms and rules are valid in this structure.

As to the axiom $c \leq_\infty d$ because $d \sqsubseteq c$, we have $d = \top, c = \omega$ or $d = c$. Then in all cases $c \leq_\mathcal{S} d$, by the axioms $(\top)$ and $(\omega\varphi)$.

As to the axiom $c \cap d =_\infty c \sqcup d$, with $c, d \in \mathcal{K}(D_0) = D_0$, we have, say, $c \sqsubseteq d$. Then $c \sqcup d = d$. Again we have $d \leq_\mathcal{S} c$. Therefore $d \leq_\mathcal{S} c \cap d \leq_\mathcal{S} d$, and hence $c \cap d =_\mathcal{S} d = c \sqcup d$.

Finally, suppose $e =_\infty (c_1 \rightarrow d_1) \cap \ldots \cap (c_n \rightarrow d_n)$, because

$$\mathsf{i}_0(e) = (c_1 \mapsto d_1) \sqcup \ldots \sqcup (c_n \mapsto d_n).$$

Then $e =_\mathcal{S} (c_1 \rightarrow d_1) \cap \ldots \cap (c_n \rightarrow d_n)$, by Corollary 19.3.26.

As to $(\Leftarrow)$, the axioms and rules $(\top), (\top \rightarrow), (\rightarrow\cap)$ and $(\rightarrow)$ hold for $\mathcal{S}$ by definition. Moreover, all axioms extra of $\mathcal{S}$ hold in $\mathcal{S}^{t_\mathcal{S}}_\infty$, as follows from the definitions of $\mathsf{i}^\mathcal{S}_0$. ∎

Now we can obtain the following result

19.3.28. Corollary. *Let* $\mathcal{S} \in \{\mathrm{Scott}, \mathrm{Park}, \mathrm{CDZ}, \mathrm{DHM}, \mathrm{HR}\}$. *Then in the category* **NLS** *we have*

$$\mathcal{F}^\mathcal{S} \cong D^\mathcal{S}_\infty.$$

Proof.

$$\begin{array}{rcll}\mathcal{F}^\mathcal{S} & \cong & \mathcal{F}^{\mathcal{S}^{t_\mathcal{S}}_\infty}, & \text{by Proposition 19.3.27,} \\ & \cong & D^\mathcal{S}_\infty, & \text{by Theorem 19.3.22.}\end{array}$$ ∎

We will end this subsection by telling what is the interest of the various models $D^\mathcal{T}_\infty$. In Barendregt [1984], Theorem 19.2.9, the following result is proved.

19.3.29. THEOREM (Hyland and Wadsworth). *Let* $t = (D_0, \mathsf{i}_0, \mathsf{j}_0)$*, where* D_0 *is cpo (or object of* **ALG***) with at least two elements and*

$$\begin{array}{rcll} \mathsf{i}_0(d) & = & \boldsymbol{\lambda} e \in D_0.d, & \textit{for } d \in \mathcal{D}_0, \\ \mathsf{j}_0(f) & = & f(\perp_{D_0}), & \textit{for } f \in [D_0 \to D_0]. \end{array}$$

Then for $M, N \in \Lambda$ *(untyped lambda terms) and* $C[\,]$ *ranging over contexts*

$$D^t_\infty \models M = N \iff \forall C[\,].(C[M] \textit{ is solvable} \iff C[N] \textit{ is solvable}).$$

In particular, the local structure of D^t_∞ *(i.e.* $\{M = N \mid D^t_\infty \models M = N\}$*) is independent of the initial* D_0*.* ∎

19.3.30. COROLLARY. *For* t *as in the theorem one has for closed terms* M, N

$$D^t_\infty \models M = N \iff \forall A \in \mathcal{S}^{\mathrm{Scott}} \, [\vdash^{\mathrm{Scott}}_{\cap^\top} M : A \iff \vdash^{\mathrm{Scott}}_{\cap^\top} N : A].$$

PROOF. Let $M, N \in \Lambda^\emptyset$. Then

$$\begin{array}{rcll} D^t_\infty \models M = N & \iff & D^{\mathrm{Scott}}_\infty \models M = N, & \text{by Theorem 19.3.29}, \\ & \iff & \mathcal{F}^{\mathcal{S}} \models M = N, & \text{by Corollary 19.3.28}, \\ & \iff & \forall A \in \mathcal{S}^{\mathrm{Scott}} \, [\vdash^{\mathrm{Scott}}_{\cap^\top} M : A \iff \vdash^{\mathrm{Scott}}_{\cap^\top} N : A], & \end{array}$$

by Theorem 18.2.8. ∎

The model D^{Park}_∞ has been introduced to contrast the following result, see Barendregt [1984], 19.3.6.

19.3.31. THEOREM (Park). *Let* t *be as in 19.3.29. Then for the untyped* λ*-term*

$$\mathsf{Y}_{Curry} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

one has

$$[\![\mathsf{Y}_{Curry}]\!]^{D^t_\infty} = \mathsf{Y}_{Tarski},$$

where Y_{Tarski} *is the least fixed-point combinator on* D^t_∞*.* ∎

The model D^{Park}_∞ has been constructed to give $\mathsf{Y}_{\mathrm{Curry}}$ a meaning different from $\mathsf{Y}_{\mathrm{Tarski}}$.

19.3.32. THEOREM (Park). $[\![\mathsf{Y}_{Curry}]\!]^{D^{\mathrm{Park}}_\infty} \neq \mathsf{Y}_{Tarski}$. ∎

Now this model can be obtained as a simple filter model $D^{\mathrm{Park}}_\infty \cong \mathcal{F}^{\mathrm{Park}}$ and therefore, by Corollory 19.3.28, one has

$$[\![\mathsf{Y}_{\mathrm{Curry}}]\!]^{\mathcal{F}^{\mathrm{Park}}} \neq \mathsf{Y}_{\mathrm{Tarski}}.$$

The models $D^{\mathrm{Scott}}_\infty$ and D^{Park}_∞ have been introduced before the definition of the type structures $\mathcal{S}^{\mathrm{Scott}}$ and $\mathcal{S}_{\mathrm{Park}}$. In case of D^{CDZ}_∞, D^{DHM}_∞ and D^{HR}_∞, however, the natural type structures $\mathcal{S}_{\mathrm{CDZ}}, \mathcal{S}_{\mathrm{DHM}}$ and $\mathcal{S}_{\mathrm{HR}}$ were defined first, in order to create models in which the images of important classes of lambda terms become definable already said at the beginning of this section The following results are translations of the results in Theorems 18.3.22, 18.3.30 and 19.2.18.

19.3.33. Proposition. *Let $M \in \Lambda^{\emptyset}$ be a closed lambda term. Then*

(i)	*M has a normal form*	$\Leftrightarrow$	$[\![M]\!]^{D_\infty^{\mathrm{CDZ}}} \sqsupseteq \varphi$.
(ii)	*M is solvable*	$\Leftrightarrow$	$[\![M]\!]^{D_\infty^{\mathrm{DHM}}} \sqsupseteq \varphi$.
(iii)	*M reduces to a* $\lambda\mathsf{I}$*-term*	$\Leftrightarrow$	$[\![M]\!]^{D_\infty^{\mathrm{HR}}} \sqsupseteq \varphi$.
(iv)	*M is strongly normalizing*	$\Leftrightarrow$	$[\![M]\!]^{\mathcal{F}^{\mathrm{CDV}}} \neq \emptyset$.

Let $M \in \Lambda$ be an (open) lambda term. Then

(iv)	*M reduces to a closed term*	$\Leftrightarrow$	$[\![M]\!]_\rho^{D_\infty^{\mathrm{Park}}} \sqsupseteq \omega$, *for all* ρ.

Proof. (i)-(ii) We explain the situation for (i), the other case being similar.

$$
\begin{array}{llll}
M \text{ has a nf} & \Leftrightarrow & \vdash^{\mathrm{CDZ}}_{\cap\top} M : \varphi, & \text{by Theorem 19.2.18(i)},\\
& \Leftrightarrow & [\![M]\!]^{\mathcal{F}^{\mathrm{CDZ}}} \ni \varphi, & \text{by 18.2.8},\\
& \Leftrightarrow & [\![M]\!]^{\mathcal{F}^{\mathrm{CDZ}}} \sqsupseteq \uparrow\varphi, & \text{by the definition of filters},\\
& \Leftrightarrow & [\![M]\!]^{D_\infty^{\mathrm{CDZ}}} \sqsupseteq \underline{\mathsf{m}}(\uparrow\varphi) = \mathsf{m}(\varphi), & \text{by Theorem 19.3.22},\\
& \Leftrightarrow & [\![M]\!]^{D_\infty^{\mathrm{CDZ}}} \sqsupseteq \Phi_{0,\infty}(\varphi), & \text{since } \varphi \in D_0,\\
& \Leftrightarrow & [\![M]\!]^{D_\infty^{\mathrm{CDZ}}} \sqsupseteq \varphi, &
\end{array}
$$

by the identification of D_0 as a subset of D_∞.

(iii) Similarly, using Theorem 18.3.30.

(iv) As (i), but simpler as the step towards D_∞ is not made.

(iv) Similarly, using Theorem 18.3.22. ∎

As particular cases we get that Scott model as defined in Scott [1972] is isomorphic to the filter λ-model induced by the type structure Scott and Park model as defined in Park [1976] is isomorphic to the filter λ-model induced by the type structure Park (see Definition 19.3.23). The construction presented here was first discussed in Coppo et al. [1984]. Other relevant references are Coppo et al. [1987], which presents the filter λ-model induced by the type structure CDZ, Honsell and Ronchi Della Rocca [1992], where the filter λ-models induced by the type structures Park, HR and other models are considered, and Alessi [1991], Di Gianantonio and Honsell [1993], Plotkin [1993], where the relation between applicative structures and type structures is studied.

We do not have a characterization like (i) in the Theorem for $D_\infty^{\mathrm{Scott}}$, as it was shown in Wadsworth [1976] that there is a closed term J without a normal form such that $D_\infty^{\mathrm{Scott}} \models \mathsf{I} = \mathsf{J}$.

Other domain equations

Results similar to Theorem 19.3.21 can be given also for other, non-extensional, inverse limit λ-models. These are obtained as solutions of domain equations involving different functors. For instance one can solve the equations

$$\begin{array}{rcl} D & = & [D \to D] \times A \\ D & = & [D \to D] + A \\ D & = & [D \to_\perp D] \times A \\ D & = & [D \to_\perp D] + A \end{array}$$

for the analysis of models for restricted λ-calculi. In all such cases one gets concise type theoretic descriptions of the λ-models obtained as fixed points of such functors corresponding to suitable choices of the mapping G, see Coppo et al. [1983]. Solutions of these equations will be discussed below. At least the following result is worthwhile mentioning in this respect, see Coppo et al. [1984] for a proof.

19.3.34. PROPOSITION. *The filter λ-model induced by* BCD *is isomorphic to $\langle \mathcal{D}, F, G\rangle$, where $\mathcal{D}$ is the initial solution of the domain equation $[\mathcal{D} \to \mathcal{D}] \times \mathsf{P}(\mathbb{A}_\infty) \equiv \mathcal{D}$, the pair $\langle F, G\rangle$ set up a Galois connection*[2] *and G is the map which picks always the minimal element in the extensionality classes of all functions.* ∎

Lazy λ-calculus

Intersection types are flexible enough to allow for the description of λ-models which are computationally adequate for the lazy operational semantics (Abramsky and Ong [1993]), i.e. *lazy* λ-models. Following Berline [2000] we define a lazy λ-model as a λ-model in which equal terms have the same order.

19.3.35. DEFINITION. The *order* of an untyped lambda term is

$$\mathrm{order}(M) = sup\{n \mid \exists N.M \twoheadrightarrow_\beta \lambda x_1 \dots x_n.N\},$$

i.e. the upperbound of the number of its initial abstractions modulo β conversion. So $\mathrm{order}(M) \in \mathbb{N} \cup \infty$.

For example order$((\lambda x.xx)(\lambda x.xx))$=0, order$(\mathsf{K})$=2 and order$(\mathsf{YK}) = \infty$.
By 18.2.8(i) we have the following result.

19.3.36. THEOREM. *Let $\mathcal{S}$ be an ITS. Then $\mathcal{F}^\mathcal{S}$ is a lazy λ-model iff the following two conditions hold.*

(*i*) *$\mathcal{F}^\mathcal{S}$ is a λ-model;*
(*ii*) $\forall \Gamma, A.[\Gamma \vdash^{\mathcal{S}}_{\cap\top} M : A \Leftrightarrow \Gamma \vdash^{\mathcal{S}}_{\cap\top} N : A] \;\Rightarrow\; order(M) = order(N)$,
i.e. M and N have the same order if they have the same types. ∎

One of the simplest type structures one can think of is AO as defined in Figure 15.1.7. This gives a lazy λ-model, which is discussed in Abramsky and Ong [1993]. It can be also used for proving the completeness of *F-semantics* (see Dezani-Ciancaglini and Margaria [1986]).

19.3.37. THEOREM (Abramsky and Ong [1993]). *Let D_∞^{lazy} be the initial solution of the domain equation $D \cong [D \to D]_\perp$ in* **ALG**. *Then $D_\infty^{lazy} \cong \mathcal{F}^{\mathrm{AO}}$.* ∎

[2] $F \circ G \sqsupseteq Id_{[\mathcal{D}\to\mathcal{D}]}$ & $G \circ F \sqsubseteq Id_\mathcal{D}$.

The λN-calculus.

Models of the λI and the λN-calculi are considered in Honsell and Lenisa [1993], Honsell and Lenisa [1999]. These are λ-applicative structures with interpretations which equate all βI or βN-redexes to their contracta. So all filter structures induced by type structures which validate βI and βN-reduction are models of the λN-calculus. Like Theorem 19.3.21 one has the following.

19.3.38. THEOREM (Honsell and Lenisa [1999]). *Let D^{I}_∞ be the inverse limit solutions of the domain equation $[D \to_\perp D] \cong D$. Then $D^{\mathsf{I}}_\infty \cong \mathcal{F}^{\mathcal{S}}$, where $\mathcal{S}$ is a strict natural type structure, with $\mathbb{A}^{\mathcal{S}} = \mathcal{K}(D_0)$.*

Honsell and Lenisa [1999] discusses a filter structure which gives a *computationally adequate* model for the *perpetual* operational semantics and a mathematical model for the maximal sensible λI-theory.

19.4. Other models

A filter model equating an arbitrary closed term to $\Delta\Delta$

In Jacopini [1975] it has been proved by an analysis of conversion that the lambda term $\Delta\Delta$, where $\Delta = \lambda x.xx$, is easy, i.e. for any closed lambda term M the equation $\Delta\Delta = M$ is consistent. This fact was proved by a Church-Rosser argument by Mitschke, see Mitschke [1976] or Barendregt [1984], Proposition 15.3.9. A model theoretical proof was given by Baeten and Boerboom [1979] who showed that for any closed M one has

$$\mathcal{P}(\omega) \models \Delta\Delta = M,$$

for a particular way of coding pairs on the set of natural numbers ω. We will now present the proof in Alessi et al. [2001] using intersection types. For an arbitrary closed λ-term M we will build a filter model $\mathcal{F}^{\mathcal{S}_M}$ such that

$$\mathcal{F}^{\mathcal{S}_M} \models \Delta\Delta = M.$$

We first examine which types can be assigned to Δ and $\Delta\Delta$.

19.4.1. LEMMA. *Let $\mathcal{S}$ be a natural type structure that is $\boldsymbol{\beta}$-sound.*
(i) $\vdash^{\mathcal{S}}_{\cap\top} \Delta : A \to B \;\Leftrightarrow\; A \leq_{\mathcal{S}} A \to B$.
(ii) $\vdash^{\mathcal{S}}_{\cap\top} \Delta\Delta : B \;\Leftrightarrow\; \exists A \in \mathbb{T}^{\mathcal{S}}. \vdash^{\mathcal{S}}_{\cap\top} \Delta : A \leq_{\mathcal{S}} (A \to B)$.
(iii) $\vdash^{\mathcal{S}}_{\cap\top} \Delta\Delta : B \;\Leftrightarrow\; \exists A \in \mathbb{T}^{\mathcal{S}}. A \leq_{\mathcal{S}} (A \to B)$.

PROOF. (i) ($\Leftarrow$) Suppose $A \leq_{\mathcal{S}} (A\to B)$. Then

$$\begin{array}{rll} x{:}A & \vdash^{\mathcal{S}}_{\cap\top} & x : (A\to B) \\ x{:}A & \vdash^{\mathcal{S}}_{\cap\top} & xx : B \\ & \vdash^{\mathcal{S}}_{\cap\top} & \lambda x.xx : (A\to B). \end{array}$$

($\Rightarrow$) Suppose $\vdash^{\mathcal{S}}_{\cap\top} \Delta : (A \to B)$. If $B =_{\mathcal{S}} \top$, then $A \leq_{\mathcal{S}} \top =_{\mathcal{S}} (A \to B)$, by Proposition 16.1.5(ii). Otherwise, by Theorem 16.1.10,

$$\begin{array}{lcll} \vdash^{\mathcal{S}}_{\cap\top} \lambda x.xx : (A{\to}B) & \Rightarrow & x{:}A \vdash^{\mathcal{S}}_{\cap\top} xx : B, & \\ & \Rightarrow & x{:}A \vdash^{\mathcal{S}}_{\cap\top} x : C, \quad x{:}A \vdash^{\mathcal{S}}_{\cap\top} x : (C{\to}B), & \text{for some } C, \\ & \Rightarrow & A \leq_{\mathcal{S}} (C{\to}B) \leq_{\mathcal{S}} (A{\to}B), & \text{by } (\to). \end{array}$$

(ii) ($\Leftarrow$) Immediate. ($\Rightarrow$) If $B =_{\mathcal{S}} \top$, then $\vdash^{\mathcal{S}}_{\cap\top} \Delta : \top \leq_{\mathcal{S}} \top{\to}B$. If $B \neq_{\mathcal{S}} \top$, then by Theorem 16.1.10(ii) one has $\vdash^{\mathcal{S}}_{\cap\top} \Delta : (A{\to}B)$, $\vdash^{\mathcal{S}}_{\cap\top} \Delta : A$, for some A. By (i) one has $A \leq_{\mathcal{S}} A{\to}B$.

(iii) By (i) and (ii). ∎

We associate to each type the maximum number of nested arrows in the leftmost path.

19.4.2. Definition. Let $\mathcal{S}$ be a type structure. For $A \in \mathbb{T}^{\mathcal{S}}$ its *type nesting*, notation $\#(A)$, is defined inductively on types as follows:

$$\begin{array}{lcll} \#(A) & = & 0 & \text{if } A \in \mathbb{A}^{\mathcal{S}}; \\ \#(A \to B) & = & \#(A) + 1; & \\ \#(A \cap B) & = & \max\{\#(A), \#(B)\}. & \end{array}$$

Lemma 19.4.1(ii) can be strengthened using type nesting. First we need the following lemma that shows that in a strongly β type structure, Definition **??**, any type A with $\#(A) \geq 1$ is equivalent to an intersection of arrows with the same type nesting.**Comment:** the proof needs to be redone using the canonical form instead of the notion of strongly $\boldsymbol{\beta}$, is it worthwhile or is it better to erase the whole subsection?

19.4.3. Lemma. *Let $\mathcal{S}$ be strongly β. Then for all $A \in \mathbb{T}^{\mathcal{S}}$ with $\#(A) \geq 1$, there exists an $A' \equiv (\bigcap_{i \in I}(C_i \to D_i)) =_{\mathcal{S}} A$ such that $\#(A') = \#(A)$.*

Proof. Every type A is an intersection of arrow types and constants:

$$A \equiv (C \to B) \cap \ldots \cap \psi \cap \ldots.$$

Since $\mathcal{S}$ is strongly β, the constants can be replaced by an intersection of arrows between constants. As $\#(A) \geq 1$ this does not increase the type nesting. ∎

19.4.4. Lemma. *Let $\mathcal{S}$ be a natural type structure which is strongly β. Then*

$$\vdash^{\mathcal{S}}_{\cap\top} \Delta\Delta : B \;\Rightarrow\; \exists A \in \mathbb{T}^{\mathcal{S}}[\vdash^{\mathcal{S}}_{\cap\top} \Delta : A \leq_{\mathcal{S}} A \to B \;\&\; \#(A) = 0].$$

Proof. Let $\vdash^{\mathcal{S}}_{\cap\top} \Delta\Delta : B$. If $B =_{\mathcal{S}} \top$ take $A \equiv \top$. Otherwise, by Lemma 19.4.1(ii), there exists $A \in \mathbb{T}^{\mathcal{S}}$ such that $\vdash^{\mathcal{S}}_{\cap\top} \Delta : A$ and $A \leq A \to B$. We show by course of value induction on $n = \#(A)$ that we can take an alternative A' with $\#(A') = 0$. If $n = 0$ we are done, so suppose $n \geq 1$. By Lemma 19.4.3, we may assume that A is of the form $A \equiv (C_1 \to D_1) \cap \ldots \cap (C_m \to D_m)$.

Now $A \leq_{\mathcal{S}} A \to B$, hence $A \leq_{\mathcal{S}} C_{i_1} \cap \ldots \cap C_{i_p}$ and $D_{i_1} \cap \ldots \cap D_{i_p} \leq B$, with $1 \leq i_1, \ldots, i_p \leq m$, since $\mathcal{S}$ is beta. Since $B \neq_{\mathcal{S}} \top$ one has $p > 0$. Hence,

$$\begin{array}{llll}
\vdash^{\mathcal{S}}_{\cap\top} \Delta : A & \Rightarrow & \vdash^{\mathcal{S}}_{\cap\top} \Delta : (C_{i_k} \to D_{i_k}), & 1 \leq k \leq p, \\
& \Rightarrow & C_{i_k} \leq_{\mathcal{S}} (C_{i_k} \to D_{i_k}), & \text{by 19.4.1(i)}, \\
& \Rightarrow & \bigcap_k C_{i_k} \leq_{\mathcal{S}} \bigcap_k (C_{i_k} \to D_{i_k}) & \\
& & \leq_{\mathcal{S}} (\bigcap_k C_{i_k} \to \bigcap_k D_{i_k}), & \text{since } \mathcal{S} \text{ is natural}, \\
& & \leq_{\mathcal{S}} (\bigcap_k C_{i_k} \to B), & \text{as } \bigcap_k D_{i_k} \leq_{\mathcal{S}} B.
\end{array}$$

Now take $A' = \bigcap_k C_{i_k}$. Then $\#(A') < n$ and we are done by the IH. ■

Now let $M \in \Lambda^{\emptyset}$. We will build the desired model satisfying $\models \Delta\Delta = M$ by taking the union of a countable sequence of type structures $\mathcal{S}_n$ defined in a suitable way to force the final interpretation of M to coincide with the interpretation of $\Delta\Delta$. In the following $\langle\cdot,\cdot\rangle$ denotes any bijection between $\mathbb{N}\times\mathbb{N}$ and $\mathbb{N}$.

19.4.5. Definition. (i) Define the growing sequence of intersection type structures $\mathcal{S}_n$ by induction on $n \in \mathbb{N}$, specifying the constants, axioms and rules.

- $\mathbb{A}^{\mathcal{S}_0} = \{\top, \omega\}$;
- $\mathcal{S}_0 = \{(\to\cap), (\top), (\top\to), (\to)\} \cup \{\omega_{\text{Scott}}\}$;
- $\mathbb{A}^{\mathcal{S}_{n+1}} = \mathbb{A}^{\mathcal{S}_n} \cup \{\xi_{\langle n,m\rangle} \mid m \in \mathbb{N}\}$;
- $\mathcal{S}_{n+1} = \mathcal{S}_n \cup \{\xi_{\langle n,m\rangle} = (\xi_{\langle n,m\rangle} \to W_{\langle n,m\rangle})\}$,

where $\langle W_{\langle n,m\rangle}\rangle_{m\in\mathbb{N}}$ is any enumeration of the set

$$\{A \mid \vdash^{\mathcal{S}_n}_{\cap\top} M : A\}.$$

(ii) We define $\mathcal{S}_M$ as follows:

$$\mathbb{A}^{\mathcal{S}_M} = \bigcup_{n\in\mathbb{N}} \mathbb{A}^{\mathcal{S}_n}; \quad \mathcal{S}_M = \bigcup_{n\in\mathbb{N}} \mathcal{S}_n.$$

19.4.6. Proposition. *$\mathcal{S}_M$ is a strongly $\boldsymbol{\beta}$, $\eta\top$-sound and natural intersection type structure.*

Proof. It is immediate to check that $\mathcal{S}_M$ is strongly $\boldsymbol{\beta}$ and $\eta\top$-sound. Clearly the axioms $(\top), (\top\to)$ and $(\to\cap)$ are valid in $\mathcal{S}_M$, as they are already in $\mathcal{S}^{\mathcal{S}_0}$. The validity of rule $(\to)$ in $\mathcal{S}_M$ follows by a "compactness" argument: if $A' \leq_{\mathcal{S}_M} A$ & $B \leq_{\mathcal{S}_M} B'$, then $A' \leq_{\mathcal{S}_n} A$ & $B \leq_{\mathcal{S}_m} B'$; but then $(A{\to}B) \leq_{\mathcal{S}_{\max(n,m)}} (A'{\to}B')$ and hence $(A{\to}B) \leq_{\mathcal{S}_M} (A'{\to}B')$. Therefore the type structure is natural. ■

19.4.7. Theorem. *The filter structure $\mathcal{F}^{\mathcal{S}_M}$ is an extensional λ-model.*

Proof. By Propositions 19.4.6 and 18.2.13. ■

We now need to show that some types cannot be deduced for Δ.

19.4.8. Lemma. $\not\vdash^{\mathcal{S}_M}_{\cap^\top} \Delta : \omega$ *and* $\not\vdash^{\mathcal{S}_M}_{\cap^\top} \Delta : (\omega \to \omega) \to \omega \to \omega$.

Proof. Define the set $\mathcal{E}_\Omega \subseteq \mathbb{T}(\mathbb{A}^{\mathcal{S}_M})$ as the minimal set such that:

$$\begin{array}{lcl} \top \in \mathcal{E}_\Omega; & & \\ A \in \mathbb{T}^{\mathcal{S}_M}, B \in \mathcal{E}_\Omega & \Rightarrow & (A \to B) \in \mathcal{E}_\Omega; \\ A, B \in \mathcal{E}_\Omega & \Rightarrow & (A \cap B) \in \mathcal{E}_\Omega; \\ W_i \in \mathcal{E}_\Omega & \Rightarrow & \xi_i \in \mathcal{E}_\Omega. \end{array}$$

Claim: $A \in \mathcal{E}_\Omega \;\Leftrightarrow\; A =_{\mathcal{S}_M} \top$.

($\Rightarrow$) By induction on the definition of $\mathcal{E}_\Omega$, using Lemma 16.1.5(i).

($\Leftarrow$) By induction on $\leq_{\mathcal{S}_M}$ it follows that

$$\mathcal{E}_\Omega \ni B \leq_{\mathcal{S}_M} A \;\Rightarrow\; A \in \mathcal{E}_\Omega.$$

Hence if $A =_{\mathcal{S}_M} \top$, one has $\mathcal{E}_\Omega \ni \top \leq_{\mathcal{S}_M} A$ and thus $A \in \mathcal{E}_\Omega$.

As $\omega \notin \mathcal{E}_\Omega$, it follows by the claim that

$$\top \neq_{\mathcal{S}_M} \omega. \tag{19.2}$$

Suppose towards a contradiction that $\vdash^{\mathcal{S}_M}_{\cap^\top} \Delta : \omega$. Then $\vdash^{\mathcal{S}_M} \Delta : \top \to \omega$, by ω_{Scott}. By Lemma 19.4.1(i) we get $\top \leq_{\mathcal{S}_M} (\top \to \omega) =_{\mathcal{S}_M} \omega \leq_{\mathcal{S}_M} \top$, i.e. $\top =_{\mathcal{S}_M} \omega$, contradicting (19.2).

Similarly from $\vdash^{\mathcal{S}_M}_{\cap^\top} \Delta : (\omega \to \omega) \to \omega \to \omega$, by Lemma 19.4.1(i), we get $\omega \to \omega \leq_{\mathcal{S}_M} (\omega \to \omega) \to (\omega \to \omega)$, which implies $\omega \to \omega \leq_{\mathcal{S}_M} \omega \leq_{\mathcal{S}_M} \top{\to}\omega$, by $\boldsymbol{\beta}$-soundness and (ω_{Scott}). Therefore $\top =_{\mathcal{S}_M} \omega$, contradicting (19.2). ∎

We finally are able to prove the main theorem.

19.4.9. Theorem. *Let $M \in \Lambda^{\emptyset}$. Then $\mathcal{F}^{\mathcal{S}_M}$ is a non-trivial extensional λ-model such that $\mathcal{F}^{\mathcal{S}_M} \models M = \Delta\Delta$.*

Proof. The model is non-trivial since clearly $\vdash^{\mathcal{S}_M}_{\cap^\top} \mathsf{I} : (\omega \to \omega) \to \omega \to \omega$ and by Lemma 19.4.8 $\not\vdash^{\mathcal{S}_M}_{\cap^\top} \Delta : (\omega \to \omega) \to \omega \to \omega$, therefore $\mathcal{F}^{\mathcal{S}_M} \not\models \mathsf{I} = \Delta$.

We must show that $[\![M]\!]^{\mathcal{S}_M} = [\![\Delta\Delta]\!]^{\mathcal{S}_M}$. Suppose that $W \in [\![M]\!]$. Then

$$\begin{array}{llll} \vdash^{\mathcal{S}_M}_{\cap^\top} M : W & \Rightarrow & \vdash^{\mathcal{S}_n}_{\cap^\top} M : W, & \text{for some } n, \\ & \Rightarrow & \xi_i =_{\mathcal{S}_{n+1}} (\xi_i \to W), & \text{for some } i, \\ & \Rightarrow & \vdash^{\mathcal{S}_M}_{\cap^\top} \Delta\Delta : W, & \text{by Lemma 19.4.1(iii)}, \\ & \Rightarrow & W \in [\![\Delta\Delta]\!]^{\mathcal{S}_M}. & \end{array}$$

This proves $[\![M]\!] \subseteq [\![\Delta\Delta]\!]$.

Now suppose $B \in [\![\Delta\Delta]\!]$, i.e. $\vdash^{\mathcal{S}_M}_{\cap^\top} \Delta\Delta : B$. Then by Lemma 19.4.4 there exists A such that $\#(A) = 0$ and $\vdash^{\mathcal{S}_M}_{\cap^\top} \Delta : A \leq_{\mathcal{S}_M} A \to B$. Let $A \equiv \bigcap_{i \in I} \psi_i$, with $\psi_i \in \mathbb{A} = \{\top, \omega, \xi_0, \ldots\}$. By Lemma 19.4.8 $\psi_i \neq_{\mathcal{S}_M} \omega$. Hence it follows that $A =_{\mathcal{S}_M} \top$ or $A =_{\mathcal{S}_M} \bigcap_{j \in J}(\xi_j)$, for some finite $J \subseteq \mathbb{N}$. Since $\top =_{\mathcal{S}_M} (\top{\to}\top)$ and $\xi_j =_{\mathcal{S}_M} (\xi_j{\to}W)$ we get $A =_{\mathcal{S}_M} (\top \to \top)$ or $A =_{\mathcal{S}_M} \bigcap_{j \in J}(\xi_j \to W_j)$. Since $A \leq_{\mathcal{S}_M} A \to B$ it follows by β-soundness that in the first case $\top \leq_{\mathcal{S}_M} B$ or in the second case $\bigcap_{j \in L} W_j \leq_{\mathcal{S}_M} B$, for some $L \subseteq J$. Since each W_j is in $[\![M]\!]$, we have in both cases $B \in [\![M]\!]$. This shows $[\![\Delta\Delta]\!] \subseteq [\![M]\!]$ and we are done. ∎

Graph models as filter models

Scott's $\mathsf{P}(\omega)$ model

Following the original notation by Scott, in the $\mathsf{P}(\omega)$ model ω denotes the set of natural numbers. $\mathsf{P}_{\mathrm{fin}}(\omega)$ denotes the set of finite subsets of ω.

NOTATION. (i) Let $\boldsymbol{\lambda} nm.\langle n,m\rangle : \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ be a bijection, e.g. the well-known one defined by $\langle n,m\rangle = \frac{1}{2}(n+m)(n+m+1)+m$.

(ii) Let $\boldsymbol{\lambda} n.e_n : \mathbb{N}\to\mathsf{P}_{\mathrm{fin}}(\omega)$ be a bijection, e.g. the well-known one defined by

$$e_n = \{k_0,\ldots,k_{m-1}\} \text{ with } k_0 < k_1 < \ldots < k_{m-1} \;\Leftrightarrow\; n = \Sigma_{i<m}2^i.$$

19.4.10. DEFINITION. [Scott [1972]] Let $\gamma : \mathsf{P}_{\mathit{fin}}(\omega)\times\omega\to\omega$ be the bijection defined by

$$\gamma(e_n, m) = \langle n,m\rangle.$$

(i) Define $F_\omega : \mathsf{P}(\omega)\to[\mathsf{P}(\omega)\to\mathsf{P}(\omega)]$ by

$$F_\omega(X) = \bigsqcup\{u\mapsto i \mid \gamma(u,i)\in X\}.$$

(ii) $G_\omega : [\mathsf{P}(\omega)\to\mathsf{P}(\omega)]\to\mathsf{P}(\omega)$ by

$$G_\omega(f) = \{\gamma(u,i) \mid i\in f(u)\}$$

for all $f\in[\mathsf{P}(\omega)\to\mathsf{P}(\omega)]$. ∎

19.4.11. PROPOSITION. *Define for $X,Y\in\mathsf{P}(\omega)$ the application*

$$X\cdot_{\mathsf{P}(\omega)} Y = \{m \mid \exists e_n\subseteq Y\ \langle n,m\rangle\in X\}.$$

Then $F_\omega(X)(Y) = X\cdot_{\mathsf{P}(\omega)} Y$ is a (more common) equivalent definition for F_ω.

PROOF. Do exercise 19.6.18. ∎

19.4.12. THEOREM (Scott [1972]). *$\mathsf{P}(\omega)$ is a λ-model via F_ω, G_ω.*

19.4.13. THEOREM (Alessi [1991]). *Define*

$$\begin{array}{lcl}\mathbb{A}^{\nabla^\omega} & = & \omega\\ \nabla^\omega & = & \nabla^{\mathrm{Engeler}}\cup\{\bigcap_{k\in e}(k\to n)\sim\gamma(e,n)\mid e\in\mathsf{P}_{\mathit{fin}}, n\in\omega\}.\end{array}$$

Then $\mathsf{P}(\omega)\cong\mathcal{F}^{\nabla^\omega}$. ∎

Plotkin's Model

19.4.14. DEFINITION (Plotkin [1993]). Let ω be an atom.
(i) Define Pm as the least set such that

$$\mathsf{Pm} = \{\omega\} \cup (\mathsf{P}_{\mathit{fin}}(\mathsf{Pm}) \times \mathsf{P}_{\mathit{fin}}(\mathsf{Pm})).$$

(ii) Define
$$\begin{array}{rcl} F_{\mathsf{Pm}} & : & \mathsf{P(Pm)}\to[\mathsf{P(Pm)}\to\mathsf{P(Pm)}] \quad \text{by} \\ G_{\mathsf{Pm}} & : & [\mathsf{P(Pm)}\to\mathsf{P(Pm)}]\to\mathsf{P(Pm)} \end{array}$$

$$\begin{array}{rcl} F_{\mathsf{Pm}}(X) & = & \bigsqcup\{u\mapsto v \mid \langle u,v\rangle \in X\} \\ G_{\mathsf{Pm}}(f) & = & \{\langle u,v\rangle \mid v \subseteq f(u)\}. \blacksquare \end{array}$$

19.4.15. THEOREM (Plotkin [1993]). *$F_{\mathsf{Pm}}, G_{\mathsf{Pm}}$ satisfy $F_{\mathsf{Pm}} \circ G_{\mathsf{Pm}} = Id$, making $\mathsf{P(Pm)}$ a λ-model.* $\blacksquare$

19.4.16. THEOREM (Plotkin [1993]). *Let* Plotkin *be as defined in Figure 15.1.7. Then $\mathsf{P(Pm)} \cong \mathcal{F}^{\text{Plotkin}}$ are isomorphic as natural λ-structures (λ-models).* $\blacksquare$

Engeler's Model

19.4.17. DEFINITION (Engeler [1981]). Let $\mathbb{A}_\infty$ be a countable set of atoms.

(i) Define Em as the least set satisfying $\mathsf{Em} = \mathbb{A}_\infty \cup (\mathsf{P}_{\mathit{fin}}(\mathsf{Em}) \times \mathsf{Em})$
(ii) Define
$$\begin{array}{rcl} F_{\mathsf{Em}} & : & \mathsf{P(Em)} \to [\mathsf{P(Em)}\to\mathsf{P(Em)}] \quad \text{by} \\ G_{\mathsf{Em}} & : & [\mathsf{P(Em)}\to\mathsf{P(Em)}]\to\mathsf{P(Em)} \end{array}$$

$$\begin{array}{rcl} F_{\mathsf{Em}}(X) & = & \bigsqcup\{u\mapsto e \mid \langle u,e\rangle \in X\} \\ G_{\mathsf{Em}}(f) & = & \{\langle u,e\rangle \mid e \in f(u)\}. \blacksquare \end{array}$$

19.4.18. THEOREM (Engeler [1981]). *$F_{\mathsf{Em}}, G_{\mathsf{Em}}$ satisfy $F_{\mathsf{Em}} \circ G_{\mathsf{Em}} = \mathbf{1}$, making $\mathsf{P(Em)}$ a λ-model.* $\blacksquare$

19.4.19. THEOREM (Plotkin [1993]). *Let* Engeler *be as defined in Figure 15.1.7. Then $\mathsf{P(Em)} \cong \mathcal{F}^{\text{Engeler}}$ are isomorphic as natural λ-structures (λ-models).* $\blacksquare$

19.5. Undecidability of inhabitation

In this section we consider type theories with infinitely many type atoms, as described in Section 15.1. To fix ideas, we are concerned here with the theory $\mathcal{T} = \text{CDV}$. Since we do not consider other type theories, in this section the symbols $\vdash$ and $\leq$ stand for $\vdash_\cap^{\text{CDV}}$ and $\leq_{\text{CDV}}$, respectively. Moreover $\mathbb{T} = \mathbb{T}^{\text{CDV}}$.

We investigate the *inhabitation problem* for this type theory, which is to determine, for a given a type A, if there exists a closed term of type A (the inhabitant). In symbols, the problem can be presented as follows:

$$\vdash\ ?: A$$

A slightly more general variant of the problem is the inhabitation problem *relativized* to a given context Γ:

$$\Gamma \vdash ? : A$$

It is however not difficult to show that these two problems are equivalent.

19.5.1. LEMMA. *Let* $\Gamma = \{x_1{:}A_1, \ldots, x_n{:}A_n\}$. *Then the following are equivalent.*

1. *There exists a term* $M \in \Lambda$ *such that* $\Gamma \vdash M : A$.
2. *There exists a term* $N \in \Lambda$ *such that* $\vdash N : A_1 \to \cdots \to A_n \to A$.

PROOF. (1) $\Rightarrow$ (2) Define $N \equiv \lambda x_1 \ldots x_n . M$. Apply n times rule ($\to$I).

(2) $\Rightarrow$ (1) Take $M \equiv N x_1 \cdots x_n$ and apply n times ($\to$E) and (weakening). ∎

The main result of the present section (Theorem 19.5.30) is that type inhabitation is undecidable for $\mathcal{T} = \mathrm{CDV}$. Compare this to Statman [1979], stating that for simple types the problem is decidable in polynomial space.

By Theorem 19.2.18 and Corollary 16.2.3 we can consider only inhabitants which are in normal form. The main idea of the undecidability proof is based on the following observation. The process of solving an instance of the inhabitation problem can be seen as a certain (solitary) game of building trees. In this way, one can obtain a combinatorial representation of the computational contents of the inhabitation problem (for a restricted class of types). We call this model a "tree game". In order to win a tree game, the player may be forced to execute a computation of a particular automaton (a "typewriter automaton", TWA). Thus, the global strategy of the proof is as follows. We make the following abbreviations.

EQA	:=	Emptiness Problem for Queue Automata;
ETW	:=	Emptiness Problem for Typewriter Automata;
WTG	:=	Problem of determining whether one can Win a Tree Game;
IHP	:=	Inhabitation Problem in $\lambda_\cap^{\mathrm{CDV}}$.

If we write $P_1 \le_T P_2$, this means that problem P_1 is (Turing) reducible to P_2, and hence that undecidability of P_1 implies that of P_2. It is well known that EQA is undecidable, see e.g. Kozen [1997]. The following inequalities show that IHP is undecidable.

EQA	$\le_T$	ETW	(Lemma 19.5.25);
ETW	$\le_T$	WTG	(Proposition 19.5.29);
WTG	$\le_T$	IHP	(Corollary 19.5.23).

Basic properties

We begin with some basic observations concerning the relation $\le$.

19.5.2. LEMMA. *Let* $n > 0$.

(i) *Let* $\alpha \in \mathbb{A}_\infty$, *and none of the* $A_1, \ldots, A_n \in \mathbb{T}$ *be an intersection. Then*

$$A_1 \cap \ldots \cap A_n \le \alpha \;\Rightarrow\; \exists i. \alpha \equiv A_i.$$

(ii) *Let* $\alpha_1, \ldots, \alpha_n \in \mathbb{A}_\infty$ *and* $A \in \mathbb{T}$ *be not an intersection.* *nor* Ω. *Then*

$$\alpha_1 \cap \ldots \cap \alpha_n \leq A \;\Rightarrow\; \exists i.A \equiv \alpha_i.$$

(iii) *Let* $\alpha_1, \ldots, \alpha_n \in \mathbb{A}_\infty$ *and* $A \in \mathbb{T}$. *Then*

$$\alpha_1 \cap \ldots \cap \alpha_n \leq A \;\Rightarrow\; A = \alpha_{i_1} \cap \ldots \cap \alpha_{i_k},$$

for some $k \geq 0$ *and* $1 \leq i_1 < \ldots < i_k \leq n$.

Proof. (i), (ii) Exercise 19.6.24.

(iii) Let $A = B_1 \cap \ldots \cap B_k$ with $k > 0$ and the B_j not intersections. $B_j \neq \Omega$. Then for each j one has $\alpha_1 \cap \ldots \cap \alpha_n \leq B_j$ and can apply (ii) to show that $B_j \equiv \alpha_{i_j}$. ∎

19.5.3. Lemma. *Let* $A_1, \ldots, A_n, B \in \mathbb{T}$ *and* $\alpha_1, \ldots, \alpha_n, \beta \in \mathbb{A}_\infty$, *with* $n > 0$. *Then*

$$(A_1 \to \alpha_1) \cap \ldots \cap (A_n \to \alpha_n) \leq (B \to \beta) \;\Rightarrow\; \exists i.\beta \equiv \alpha_i \;\&\; B \leq A_i.$$

Proof. By Theorem 16.1.8 CDV is $\boldsymbol{\beta}$-sound. Hence the assumption implies that $B \leq (A_{i_1} \cap \ldots \cap A_{i_k})$ and $(\alpha_{i_1} \cap \ldots \cap \alpha_{i_k}) \leq \beta$. By Lemma 19.5.2(ii) one has $\beta \equiv \alpha_{i_p}$, for some $1 \leq p \leq k$, and the conclusion follows. ∎

19.5.4. Lemma. *If* $\Gamma \vdash \lambda x.M : A$ *then* $A \notin \mathbb{A}_\infty$.

Proof. Suppose $\Gamma \vdash \lambda x.M : \alpha$. By Lemma 16.1.1(iii) it follows that there are $n > 0$ and $B_1, \ldots, B_n, C_1, \ldots, C_n$ such that $\Gamma, x : B_i \vdash M : C_i$, for $1 \leq i \leq n$, and $(B_1 \to C_1) \cap \ldots \cap (B_n \to C_n) \leq \alpha$. This is impossible by Lemma 19.5.2(i). ∎

Game contexts

In order to prove that a general decision problem is undecidable, it is enough to identify a "sufficiently difficult" fragment of the problem and prove undecidability of that fragment. Such an approach is often useful. This is because restricting the consideration to specific instances may simplify the analysis of the problem. Of course the choice should be done in such a way that the "core" of the problem remains within the selected special case. This is the strategy we are applying for our inhabitation problem. Namely, we restrict our analysis to the following special case of relativized inhabitation.

$$\Gamma \vdash ? : \alpha,$$

where α is a type atom, and Γ is a "game context", the notion of game context being defined as follows.

19.5.5. Definition. (i) If $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{T}$ are sets of types, then

$$\begin{array}{rcl} \mathcal{X} \to \mathcal{Y} & = & \{X \to Y \mid X \in \mathcal{X},\ \mathsf{Y} \in \mathcal{Y}\}. \\ \mathcal{X} \cap \mathcal{Y} & = & \{X \cap Y \mid X \in \mathcal{X},\ \mathsf{Y} \in \mathcal{Y}\}. \\ \mathcal{X}^{\cap} & = & \{A_1 \cap \ldots \cap A_n \mid n \geq 1 \;\&\; A_1, \ldots, A_n \in \mathcal{X}\}. \end{array}$$

If $A \equiv A_1 \cap \ldots \cap A_n$, and each A_i is not an intersection, then A_i are called the *components* of A.

(ii) We consider the following sets of types.
(1) $\mathcal{A} = \mathbb{A}_\infty^\cap$.
(2) $\mathcal{B} = (\mathbb{A}_\infty \to \mathbb{A}_\infty)^\cap$.
(3) $\mathcal{C} = (\mathcal{D} \to \mathbb{A}_\infty)^\cap$.
(4) $\mathcal{D} = (\mathcal{B} \to \mathbb{A}_\infty) \cap (\mathcal{B} \to \mathbb{A}_\infty)$.
(iii) Types in $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$ are called *game types*.
(iv) A type context Γ is a *game context* iff all types in Γ are game types.

Write $A \in \mathcal{A}$ (respectively $\in \mathcal{B}$, $\in \mathcal{C}$) if $A = A' \in \mathcal{A}$ $(\mathcal{B}, \mathcal{C})$ for some A'. We show some properties of type judgements involving game types.

19.5.6. Lemma. *For a game context Γ and $A \neq \Omega$ the following hold.*
(i) $\Gamma \vdash xM : A \quad \Rightarrow \quad A \in \mathcal{A}\ \&\ \Gamma(x) = (E_1 \to \alpha_1) \cap \ldots \cap (E_n \to \alpha_n),\ n > 0,$
$\&\ A = \alpha_{i_1} \cap \ldots \cap \alpha_{i_k}\ \&\ \Gamma \vdash M : E_{i_1} \cap \ldots \cap E_{i_k},$
for some $k > 0\ \&\ 1 \leq i_1 < \ldots < i_k \leq n$.
(ii) $\Gamma \vdash xM : \alpha \quad \Rightarrow \quad \Gamma(x) = (E_1 \to \alpha_1) \cap \ldots \cap (E_n \to \alpha_n),\ n > 0,$
$\&\ \alpha = \alpha_i\ \&\ \Gamma \vdash M : E_i,$ *for some* $1 \leq i \leq n$.
(iii) $\Gamma \not\vdash xMN : A$.

Proof. (i) Suppose $\Gamma \vdash xM : A$. By Lemma 16.1.10(ii) we have for some type B that $\Gamma \vdash x : (B \to A)$ and $\Gamma \vdash M : B$. Then $\Gamma(x) \leq B \to A$, by Lemma 16.1.1(i). By Lemma 19.5.2(ii), this cannot happen if $\Gamma(x)$ is in $\mathcal{A}$. Thus $\Gamma(x)$, being a game type, is of the form $(E_1 \to \alpha_1) \cap \ldots \cap (E_n \to \alpha_n)$. Since CDV is $\boldsymbol{\beta}$-sound and

$$(E_1 \to \alpha_1) \cap \ldots \cap (E_n \to \alpha_n) = \Gamma(x) \leq (B \to A),$$

we have $B \leq (E_{i_1} \to \alpha_{i_1}) \cap \ldots \cap (E_{i_h} \to \alpha_{i_h})$ and $\alpha_{i_1} \cap \ldots \cap \alpha_{i_k} \leq A$, for some i_j such that $1 \leq i_j \leq n$. Then, by Lemma 19.5.2(iii), we conclude that $A = \alpha_{i_1} \cap \ldots \cap \alpha_{i_k}$, where $0 < k \leq h$.
(ii) By (i) and Lemma 19.5.2(ii).
(iii) By (i), using that $B \to A \neq \alpha_{i_1} \cap \ldots \cap \alpha_{i_k}$, by Lemma 19.5.2(ii). ∎

19.5.7. Lemma. *If A is a game type and $D \in \mathcal{D}$, then $A \not\leq D$.*

Proof. Suppose $A \leq D \leq (B \to \alpha)$, with $B \in \mathcal{B}$. The case $A \in \mathcal{A}$, is impossible by Lemma 19.5.2(ii). If $A \in \mathcal{B}$, then $(\alpha_1 \to \beta_1) \cap \ldots \cap (\alpha_n \to \beta_n) \leq B \to \alpha$ and hence $B \leq \alpha_i$ for some i, by Lemma 19.5.3. By Lemma 19.5.2(i) this is also impossible. If $A \in \mathcal{C}$, then $(D_1 \to \beta_1) \cap \ldots \cap (D_n \to \beta_n) \leq B \to \alpha$ and hence $B \leq D_i \in \mathcal{D}$ for some i, by Lemma 19.5.3. We have already shown that this is impossible. ∎

For game contexts the Generation Lemma 16.1.10 can be extended as follows.

19.5.8. Lemma. *Let Γ be a game context, and let M be in normal form.*
(i) *If $\Gamma \vdash M : (B_1 \to \alpha_1) \cap (B_2 \to \alpha_2) \in \mathcal{D}$, with $B_i \in \mathcal{B}$, then $M \equiv \lambda y.N$, and $\Gamma, y{:}B_i \vdash N : \alpha_i$ for $i = 1, 2$.*
(ii) *If $\Gamma \vdash M : \alpha$, with $\alpha \in \mathbb{A}_\infty$, then there are two exclusive possibilities.*
- *M is a variable z and $\Gamma(z)$ is in $\mathcal{A}$, where α is one of the components.*

- *$M \equiv xN$, where $\Gamma(x) = (E_1 \to \beta_1) \cap \cdots \cap (E_n \to \beta_n)$, and $\alpha = \beta_i$ and $\Gamma \vdash N : E_i$, for some $1 \leq i \leq n$.*

Proof. (i) Notice first that by Lemmas 19.5.6 and 19.5.2(ii) the term M cannot be an application. If it is a variable x, then $\Gamma(x) \leq (B_1 \to \alpha_1) \cap (B_2 \to \alpha_2)$, by Lemma 16.1.1(i). This contradicts Lemma 19.5.7, because $\Gamma(x)$ is a game type. It follows that $M = \lambda y.N$ and $\Gamma \vdash \lambda y.N : (B_i {\to} \alpha_i)$. Then for $i = 1, 2$ one has $\Gamma, y{:}B_i \vdash N_i : \alpha_i$, by Lemma 16.1.10(iii).

(ii) M is not an abstraction by Lemma 19.5.4. If $M \equiv x$, then $\Gamma(x) \leq \alpha$ and $\Gamma(x)$ is a game type, i.e. in $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. By Lemma 19.5.2(i) one has $\Gamma(x) \in \mathcal{A}$, with α as one of the components. If M is an application, $M \equiv xM_1 \ldots M_m$, $m > 0$, write $\Gamma(x) = (E_1 \to \beta_1) \cap \cdots \cap (E_n \to \beta_n)$, as it is a game type and by Theorem 16.1.10(ii) it cannot be in $\mathcal{A}$. Then $M \equiv xN$, by Lemma 19.5.6(iii). By (ii) of the same Lemma one has $\alpha = \beta_i$ and $\Gamma \vdash N : E_i$ for some i. ∎

Tree games

In order to show the undecidability of inhabitation for CDV we will introduce a certain class of *tree games*. 'Rounds' in a tree game are an intermediate step in our construction. The idea of a tree game is to represent, in an abstract way, the crucial combinatorial behaviour of proof search in CDV. We will first show how inhabitation problems can be represented by tree games and then how tree games can represent computations of certain machines called typewriter automata (TWA).

19.5.9. Definition. Let Σ be a finite alphabet; its elements are called *labels*.

1. A *local move* (over Σ) is a finite nonempty set B of pairs of labels.
2. A *global move* (over Σ) is a finite nonempty set C of triples of the form
$$\langle \langle X, b \rangle, \langle Y, c \rangle, d \rangle,$$
where $b, c, d \in \Sigma$ and X, Y are local moves.
3. A *tree game* (over Σ) is a triple of the form
$$G = \langle a, A, \{C_1, \ldots, C_n\} \rangle,$$
where $a \in \Sigma$, $A \subseteq \Sigma$ and $C_1, \ldots, C_n$ are global moves. We call a the *initial label* and A the set of *final labels*.

Before we explain the rules of the game, we give an interpretation of the constituents of the tree games in terms of types.

19.5.10. Definition. Let Σ be a finite subset of $\mathbb{A}_\infty$, the infinite set of type atoms, and let G be a tree game over Σ. Moves of G, and the set of final labels, can be interpreted as types of CDV as follows.

1. If $A = \{a_1, \ldots, a_n\}$, then $\tilde{A} = a_1 \cap \ldots \cap a_n$.

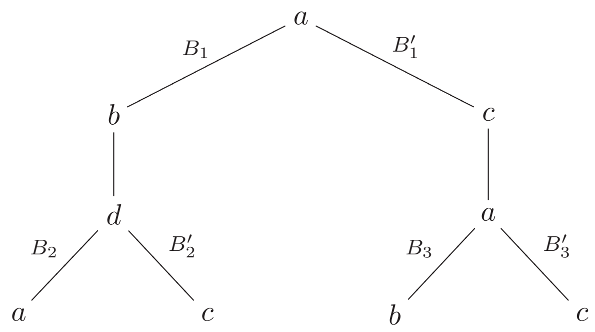

Figure 19.2: An example position

2. If $B = \{\langle a_1, b_1 \rangle, \ldots, \langle a_n, b_n \rangle\}$, then $\tilde{B} = (a_1 \to b_1) \cap \ldots \cap (a_n \to b_n)$.
3. If $C = \{\langle \langle B_1, b_1 \rangle, \langle B_1', b_1' \rangle, c_1 \rangle, \ldots, \langle \langle B_n, b_n \rangle, \langle B_n', b_n' \rangle, c_n \rangle\}$, then

$$\tilde{C} = (((\tilde{B_1} \to b_1) \cap (\tilde{B_1'} \to b_1')) \to c_1) \cap \ldots \cap (((\tilde{B_n} \to b_n) \cap (\tilde{B_n'} \to b_n')) \to c_n).$$

Notice that $\tilde{A} \in \mathcal{A}$, $\tilde{B} \in \mathcal{B}$ and $\tilde{C} \in \mathcal{C}$.

A tree game is a solitary game, i.e., there is only one player. Starting from an initial position, the player can nondeterministically choose a sequence of moves, and wins if (s)he can manage to reach a final position. Every position (configuration) of the game is a finite labelled tree, and at every step the depth of the tree is increasing.

19.5.11. Definition. Let $G = \langle a, A, \{C_1, \ldots, C_n\} \rangle$ be a tree game over Σ. A *position* T of G is a finite labelled tree, satisfying the following conditions.

- The root is labelled by the initial symbol a;
- Every node has at most two children;
- Nodes at the same level (the same distance from the root) have the same number of children (in particular all leaves are at the same level);
- All nodes are labelled by elements of Σ;
- In addition, if a node v has two children v' and v'', then the branches $\langle v, v' \rangle$ and $\langle v, v'' \rangle$ are labelled by local moves.

19.5.12. Definition. Let $G = \langle a, A, \{C_1, \ldots, C_n\} \rangle$ be a tree game.

1. The *initial position* of G is the tree with a unique node labelled a.
2. A position T is *winning* iff all labels of the leaves of T are in A.

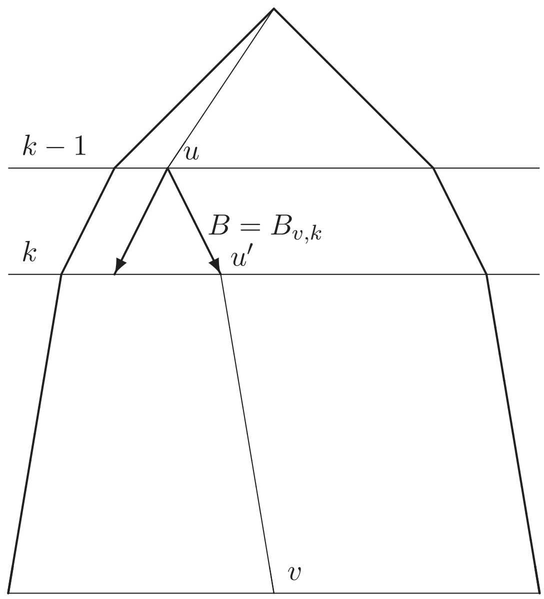

Figure 19.3: Local move X associated to node v

19.5.13. Definition. Let T be a position in a game $G = \langle a, A, \{C_1, \dots, C_n\}\rangle$, and let v be a leaf of T. Let k be such that all nodes in T at level $k-1$ have two children as shown in Figure 19.3. There is a node u at level $k-1$ which is an ancestor of v, one of the children of u, say u', is also an ancestor of v (possibly improper, i.e., it may happen that $u' = v$). Assume that B is the label of the branch $\langle u, u' \rangle$. Then we say that B is the *k-th local move associated to* v, and we write $B = B_{v,k}$.

Now we can finally describe the rules of the game.

19.5.14. Definition. (i) Let $G = \langle a, A, \{C_1, \dots, C_n\} \rangle$ and let T be a (current) position in G. There are two possibilities to obtain a next position.

(1) The player can perform a "global" step, by first selecting one of the global moves C_i and then performing the following actions for each leaf v of T.
   - Choose a triple $\langle \langle B, b \rangle, \langle B', b' \rangle, c \rangle \in C_i$ such that c is the label of v;
   - Create two children of v, say v' and v'', labelled b and b', respectively;
   - Label the branch $\langle v, v' \rangle$ by B and the branch $\langle v, v'' \rangle$ by B'.

   The step is only allowed if the resulting tree is *legal*, i.e. satisfies the conditions of definition 19.5.11.

(2) The player can also perform a "local" step. This begins with a choice of a level $k > 0$ of T such that each node at level $k-1$ has two children. Then, for each leaf v of T, the player executes the following actions.
   - Choose a pair $\langle a, b \rangle \in B_{v,k}$ such that b is the label of v;
   - Create a single child of v, labelled a.

Again the step is only allowed the appropriate actions can be performed at every leaf, otherwise the resulting tree is not a position.

(ii) If a position T' is reachable from T with help of one global step C_i, we write

$$T \Rightarrow^{C_i} T'.$$

If T' is obtained from T by a local step defined via level k, we write

$$T \Rightarrow^{k} T'.$$

If T' is reachable from T in one step (global or local), then we write $T \Rightarrow T'$.

(iii) A position T in G is called *favorable* iff there is a position T' with

$$T \Rightarrow_T^* T' \text{ and } T' \text{ is winning,}$$

where $\Rightarrow_T^*$ is the reflexive transitive closure of $\Rightarrow_T$.

(iv) The game G *can be won by the player*, notation sol(G), iff the initial position is favorable.

The following example gives an idea of the tree games and moreover it is important for our principal construction.

19.5.15. Example. Consider the tree game $G_0 = \langle\, 1, \{c\}, \{C_1, C_2\}\,\rangle$, over the alphabet $\Sigma = \{1, 2, a, b, c\}$, where

- $C_1 = \{\langle\, \langle\, \{\langle a, a\rangle\}, 1\,\rangle, \langle\, \{\langle b, a\rangle\}, 2\,\rangle, 1\,\rangle, \langle\, \langle\, \{\langle a, b\rangle\}, 1\,\rangle, \langle\, \{\langle b, b\rangle\}, 2\,\rangle, 2\,\rangle\}$;
- $C_2 = \{\langle\, \langle\, \{\langle c, a\rangle\}, a\,\rangle, \langle\, \{\langle c, a\rangle\}, a\,\rangle, 1\,\rangle, \langle\, \langle\, \{\langle c, b\rangle\}, a\,\rangle, \langle\, \{\langle c, b\rangle\}, a\,\rangle, 2\,\rangle\}$.

Figure 19.4 demonstrates a possible winning position T of the game. Note that this position can actually be reached from the initial one in 6 steps, so that the player can win G_0. These 6 steps are as follows

$$T_0 \Rightarrow^{C_1} T_1 \Rightarrow^{C_1} T_2 \Rightarrow^{C_2} T_3 \Rightarrow^{1} T_4 \Rightarrow^{2} T_5 \Rightarrow^{3} T_6 = T,$$

where each T_i is of depth i. The reader should observe that every sequence of steps leading from T_0 to a winning position must obey a similar pattern:

$$T_0 \Rightarrow^{C_1} T_1 \Rightarrow^{C_1} \cdots \Rightarrow^{C_1} T_{n-1} \Rightarrow^{C_2} T_n \Rightarrow^{1} T_{n+1} \Rightarrow^{2} \cdots \Rightarrow^{n} T_{2n},$$

where T_{2n} is winning. Thus, the game must consist of two phases: first a number of applications of C_1, then a single application of C_2 and then a sequence of steps using only local moves. What is important in our example is that the order of local steps is fully determined. Indeed, at the position T_{n+k-1} the only action possible is "$\Rightarrow^k$". That is, one must apply the k-th local moves associated to the leaves (the moves labelling branches at depth k). This is forced by the distribution of symbols a, b at depth $n + k - 1$.

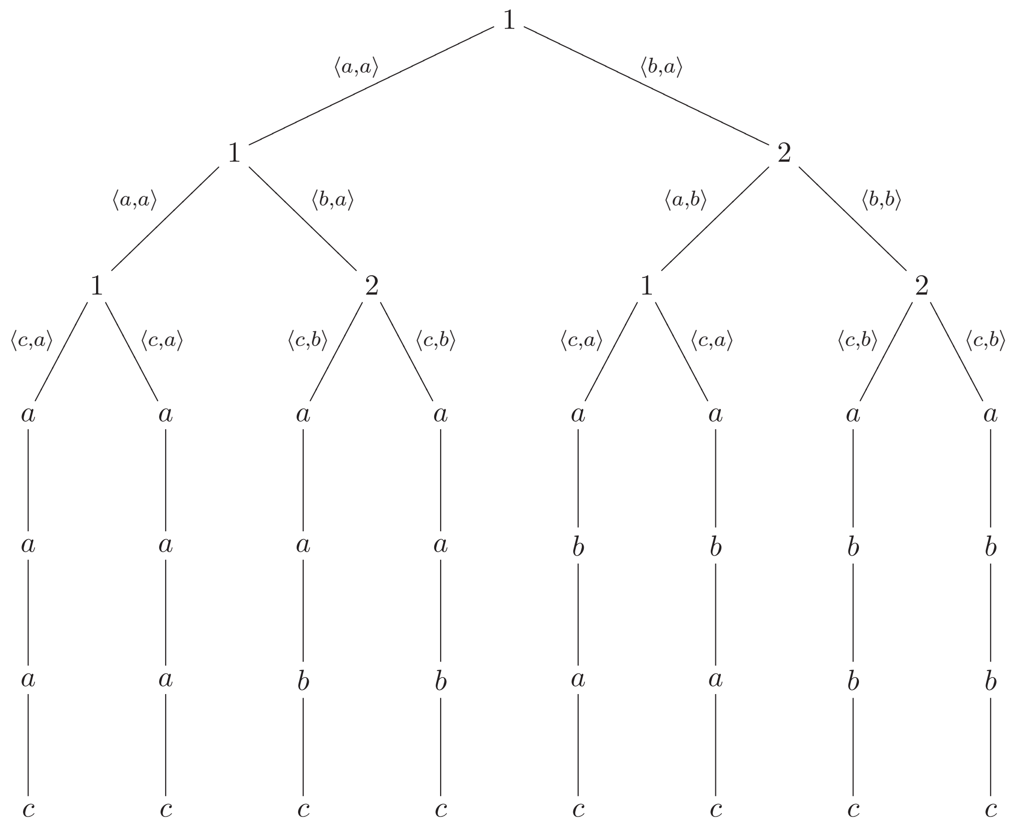

Figure 19.4: A winning position in G_0 of Example 19.5.15.

Let us emphasize a few properties of our games. First, a game is a nondeterministic process, and there are various sequences of steps possible. We can have winning sequences (reaching a winning position), and infinitely long sequences, but also "deadlocks" when no rule is applicable. Note that there are various levels of nondeterminism here: we can choose between C_i's and k's and then between various elements of the chosen set C_i (respectively $B_{v,k}$). It is an important property of the game that the actions performed at various leaves during a local step may be different, as different moves $B_{v,k}$ were "declared" before at the corresponding branches of the tree.

We now explain the relationship between tree games and term search in CDV. Since we deal with intersection types, it is not unexpected that we need sometimes to require one term to have many types, possibly within different contexts. This leads to the following definition.

19.5.16. Definition. Let $k, n > 0$. Let $\vec{A}_1, \ldots, \vec{A}_n$ be n sequences of k types:

$$\vec{A}_i = A_{i1}, \ldots, A_{ik}.$$

Let $\Gamma_i = \{x_1{:}A_{i1}, \ldots, x_k{:}A_{ik}\}$ and let $\alpha_i \in \mathbb{A}_\infty$, for $1 \leq i \leq n$. A *generalized inhabitation problem (gip)* is a finite set of pairs $P = \{\langle \Gamma_i, \alpha_i \rangle \mid i = 1, \ldots, n\}$, where each Γ_i and α_i are as above. A *solution* of P is a term M such that $\Gamma_i \vdash M : \alpha_i$ holds for each i. We say *'M solves P'*. This is equivalent to requiring that

$$\vdash \lambda\vec{x}.M : (\vec{A}_1 \to \alpha_1) \cap \ldots \cap (\vec{A}_n \to \alpha_n).$$

19.5.17. Definition. Let $G = \langle a, A, \{C_1, \ldots, C_n\} \rangle$ be a tree game and let T be a position in G.

(i) We have $\Gamma_G = \{x_0{:}\tilde{A}, x_1{:}\tilde{C}_1, \ldots, x_n{:}\tilde{C}_n\}$.

(ii) Let J be the set of all numbers k such that every node in T at level $k-1$ has two children. We associate a new variable y_k to each $k \in J$. Define for a leaf v of T the basis

$$\Gamma_v = \{y_k{:}\tilde{B}_{v,k} \mid k \in J\}.$$

(iii) Define the gip $P^G_T = \{\langle \Gamma_G \cup \Gamma_v, a_v \rangle \mid v \text{ is a leaf of } T \text{ with label } a_v\}$.

The following lemma states the exact correspondence between inhabitation and games. Let us make first one comment that perhaps may help to avoid confusion. We deal here with quite a restricted form of inhabitation problem (only game contexts), which implies that the lambda terms to be constructed have a "linear" shape (Exercise 19.6.26). Thus we have to deal with trees not because of the shape of lambda terms (as often happens with proof search algorithms) but exclusively because the nature of intersection types, and because we need to solve various inhabitation problem uniformly (i.e., to solve *gip*).

19.5.18. Lemma. *Let $G = \langle a, A, \{C_1, \ldots, C_n\} \rangle$ be a tree game.*

(i) *If T is a winning position of G, then x_0 solves P^G_T.*

(ii) *If $T_1 \Rightarrow^{C_i} T_2$ and N solves $P^G_{T_2}$, then $x_i(\lambda y_k.N)$ solves $P^G_{T_1}$, where k is the depth of T_1 and $i > 0$.*

(iii) *If* $T_1 \Rightarrow^k T_2$ *and* N *solves* $P^G_{T_2}$, *then* $y_k N$ *solves* $P^G_{T_1}$.

PROOF. (i) T is winning, hence $a_v \in A$ for each leaf v. Hence $x_0{:}\tilde{A} \vdash x_0 : a_v$ and therefore $\Gamma_G \cup \Gamma_v \vdash x_0 : a_v$.

(ii) $T_1 \Rightarrow^{C_i} T_2$, hence each leaf v of T_1 has two children v' and v'' in T_2 and the branches $\langle v, v'\rangle$ and $\langle v, v''\rangle$ are labelled by B and B' such that $\langle\langle B, a_{v'}, a_v\rangle, \langle B', a_{v''}\rangle\rangle \in T_i$. Let k =level(v). As N solves $P^G_{T_2}$ one has

$$\begin{array}{rcl} \Gamma_G \cup \Gamma_v, y_k{:}\tilde{B} & \vdash & N : a_{v'}; \\ \Gamma_G \cup \Gamma_v, y_k{:}\tilde{B}' & \vdash & N : a_{v''}. \end{array}$$

So $\Gamma_G \cup \Gamma_v \vdash \lambda y_k.N : (\tilde{B} \to a_{v'}) \cap (\tilde{B}' \to a_{v''})$. Therefore $\Gamma_G \cup \Gamma_v \vdash x_i(\lambda y_k.N) : a_v$.

(iii) $T_1 \Rightarrow^k T_2$, hence each leaf v of T_1 has a child v' in T_2 such that $\langle a_{v'}, a_v\rangle \in B_{v,k}$. Then $y_k{:}\tilde{B}_{v,k} \vdash y_k : a'_v \to a_v$ and $\Gamma_G \cup \Gamma_v \vdash N : a_{v'}$, by assumption. Therefore $\Gamma_G \cup \Gamma_v \vdash y_k N : a_v$. ∎

19.5.19. COROLLARY. *Let* G *be a tree game. For positions* T *one has*

$$T \text{ is favorable} \Rightarrow P^G_T \text{ has a solution.}$$

PROOF. By induction on the number of steps needed to reach a winning position and the lemma. ∎

For the converse we need the following result.

19.5.20. LEMMA. *Let* T_1 *be a position in a tree game* G *and let* M *be a solution in* $\boldsymbol{\beta}$*-nf of* $P^G_{T_1}$. *Then we have one of the following cases.*

1. $M \equiv x_0$ *and* T_1 *is winning.*
2. $M \equiv y_k N$ *and* N *is the solution of* $P^G_{T_2}$, *for some* T_2 *with* $T_1 \Rightarrow^k T_2$.
3. $M \equiv x_i(\lambda y_k.N)$ *and* N *is the solution of* $P^G_{T_2}$, *for some* T_2 *with* $T_1 \Rightarrow^{C_i} T_2$.

PROOF. Case $M \equiv z$. As M is a solution of $P^G_{T_1}$, one has

$$\Delta_v = \Gamma_G \cup \Gamma_v \vdash z : a_v,$$

for all leaves v of T_1. Then by Lemma 16.1.1(i) one has $\Delta_v(z) \leq a_v$. Hence by Lemma 19.5.2(i) $a_v \in A$ and $z = x_0$. Therefore T_1 is winning.

Case M is an application. Then for all leaves v of T_1

$$\Delta_v = \Gamma_G \cup \Gamma_v \vdash M : a_v.$$

By Lemma 19.5.8(ii) $M \equiv zN$ and $\Delta_v(z) \equiv (E_1 \to \beta_1) \cap \cdots \cap (E_n \to \beta_n)$, with $\Delta_v \vdash N : E_j$ and $a_v = \beta_j$, for some j. Now choose a leaf v of T_1. As Δ_v is a game context there are only two possibilities

$$E_j \in \mathbb{A}_\infty \text{ or } E_j \in \mathcal{D}.$$

Subcase $E_j \in \mathbb{A}_\infty$. Let $E_j \equiv \alpha_j$. Now $z \notin \mathrm{dom}(\Gamma_G)$, hence $z \in \mathrm{dom}(\Gamma_v)$ and $z{:}(\alpha_1 \to \beta_1) \cap \ldots \cap (\alpha_n \to \beta_n)$ being $y_k : \tilde{B}_{v,k}$, for some k. Also for each leaf w of T_1 one has $z \in \mathrm{dom}(\Gamma_w)$ and $z{:}\Gamma_w(z)$ is $y_k{:}\tilde{B}_{w,k}$. Define $T_1 \Rightarrow^k T_2$ by giving each leaf v of T_1 with label β_j a child v' with label α_j. We have $\Delta_v \vdash N : \alpha_j$ and $y_k{:}\tilde{B}_{v,k} \vdash y_k : \alpha_j \to \beta_j$. Hence $\Delta_{v'} \vdash M : a_{v'}$.

Subcase $E_j \in \mathcal{D}$. Then $\Delta_v(z) \equiv (E_1 \to \beta_1) \cap \ldots \cap (E_n \to \beta_n)$. Hence $z : \Delta_v(z)$ is $x_i{:}\tilde{C}_i$ for some i. So $z \in \mathrm{dom}(\Gamma_G)$ and therefore $\Delta_w(z) = \Delta_v(z)$ for all leaves w of T_1. Let $C_i = \{\ldots, \langle\langle B_j, \alpha_j\rangle, \langle B'_j \alpha'_j\rangle, \beta_j\rangle, \ldots\}$ and $E_j = ((\tilde{B}_j \to \alpha_j) \cap (\tilde{B}'_j \to \alpha'_j))$. Define the move $T_1 \Rightarrow^{C_i} T_2$ as follows. Give each leaf v with label β_j two children v' and v'' with labels α_j and α'_j, respectively. Label the branches with B_j and B'_j, respectively. Let $k = $depth$(T_1)$. One has $\Delta_v \vdash N : E_j$, hence by Lemma 19.5.8(i) one has $N \equiv \lambda y_k.N'$, with

$$\Delta_v, y_k{:}\tilde{B}_j \vdash N'{:}\alpha_j \;\&\; \Delta_v, y_k{:}\tilde{B}'_j \vdash N' : \alpha'_j.$$

Therefore $\Delta_{v'} \vdash N' : a_{v'}$ and $\Delta_{v''} \vdash N' : a_{v''}$.

The case that M is an abstraction is impossible, by Lemma 19.5.4. ■

19.5.21. Corollary. *Let G be a tree game. For positions T one has*

$$T \textit{ is favorable} \;\Leftrightarrow\; P^G_T \textit{ has a solution.}$$

Proof. ($\Rightarrow$) This was Corollary 19.5.19. ($\Leftarrow$) Let M be a solution of P^G_T. By Theorem 19.2.18(ii) one may assume that M is in normal form. The conclusion follows from the previous lemma by induction on the size of M. ■

19.5.22. Theorem. *Let G be a tree game with initial position $\{a\}$. Then*

$$sol(G) \;\Leftrightarrow\; P^G_{\{a\}} \textit{ has a solution.}$$

Proof. Immediate from the previous Corollary. ■

19.5.23. Corollary. *WTG $\leq_T$ IHP, i.e. winning a tree-game can be reduced to the inhabitation problem.*

Proof. By the theorem, as $P^G_{T_0} = P^G_a$ is an inhabitation problem. ■

Typewriters

In order to simplify our construction we introduce an auxiliary notion of a typewriter automaton. Informally, a typewriter automaton is just a reusable finite-state transducer. At each step, it reads a symbol, replaces it by a new one and changes the internal state. But at the end of the word, our automaton moves its reading and printing head back to the beginning of the tape and continues. This goes on until a final state is reached. That is, a typewriter automaton is a special case of a linear bounded automaton, see Kozen [1997]. A formal definition follows.

19.5.24. Definition. (i) A (deterministic) *typewriter automaton* $\mathcal{A}$ is a tuple of the form

$$\mathcal{A} = \langle\, \Sigma, Q, q_0, F, \varrho \,\rangle,$$

where Σ is a finite alphabet, Q is a finite set of states, $q_0 \in Q$ is an initial state and $F \subseteq Q$ is a set of final states. The last component is a transition function $\varrho : (Q - F) \times (\Sigma \cup \{\varepsilon\}) \to Q \times (\Sigma \cup \{\varepsilon\})$, which must satisfy the following condition: whenever $\varrho(q,a) = (p,b)$, then either $a, b \in \Sigma$ or $a = b = \varepsilon$.

(ii) A configuration (instantaneous description, ID) of $\mathcal{A}$ is represented by a triple $\langle\, w, q, v \,\rangle$, where (as usual) $wv \in \Sigma^*$ is the tape contents, $q \in Q$ is the current state, and the machine head points at the first symbol of v.

(iii) The next ID function $\overline{\varrho}$ is defined as follows:

- $\overline{\varrho}(\langle\, w, q, av \,\rangle) = \langle\, wb, p, v \,\rangle$, if $a \neq \varepsilon$ and $\varrho(q,a) = (p,b)$;
- $\overline{\varrho}(\langle\, w, q, \varepsilon \,\rangle) = \langle\, \varepsilon, p, w \,\rangle$, if $\varrho(q,\varepsilon) = (p,\varepsilon)$.

(iv) The language $L^{\mathcal{A}}$ accepted by $\mathcal{A}$ is the set of all $w \in \Sigma^*$, such that $\overline{\varrho}^k(\langle\, \varepsilon, q_0, w \,\rangle) = \langle\, u, q, v \,\rangle$, for some k and $q \in F, uv \in \Sigma^*$.

(v) ETW is the emptiness problem for typewriter automata.

Recall that EQA, the emptiness problem for queue automata, is undecidable (see Kozen [1997]). We need the following.

19.5.25. Lemma. EQA $\leq_T$ ETW.

Proof. Exercise 19.6.28. ∎

It follows that also ETW is undecidable.

Our goal is now to represent typewriters as games, in order to establish ETW$\leq$WTG. We begin with a refinement of Example 19.5.15. In what follows, triples of the form $\langle\, \langle\, B_1, b \,\rangle, \langle\, B_2, c \,\rangle, d \,\rangle$ will be represented graphically as

$$\begin{tikzcd} & d \arrow[dl, "B_1"'] \arrow[dr, "B_2"] & \\ b & & c \end{tikzcd}$$

in order to enhance readability.

19.5.26. Definition. The alphabet Σ_1 is the following Cartesian product.

$$\Sigma_1 = \{\perp, a, b\} \times \{\perp, 1, 2\} \times \{\mathrm{loc}, \mathrm{glo}\}.$$

We define a tree game $G_1 = \langle\, \langle\, \perp, 1, C \,\rangle, \Sigma_1, \{C_1, C_2, C_3\} \,\rangle$. The set of accepting labels is Σ_1, because we are interested in all possible 'rounds' (i.e. instances) of the game. The moves are defined as follows.

(i) Global move C_1 consists of the following two triples:

$$
\begin{tikzcd}
 & \langle \bot, 1, \mathrm{glo} \rangle \arrow[dl, "\{(\langle a,x,\mathrm{glo}\rangle,\langle a,x,\mathrm{loc}\rangle) \mid x\in\{\bot,1,2\}\}"'] \arrow[dr, "\{(\langle b,x,\mathrm{glo}\rangle,\langle a,x,\mathrm{loc}\rangle) \mid x\in\{\bot,1,2\}\}"] & \\
\langle \bot, 1, \mathrm{glo} \rangle & & \langle \bot, 2, \mathrm{glo} \rangle
\end{tikzcd}
$$

and

$$
\begin{tikzcd}
 & \langle \bot, 2, \mathrm{glo} \rangle \arrow[dl, "\{(\langle a,x,\mathrm{glo}\rangle,\langle b,x,\mathrm{loc}\rangle) \mid x\in\{\bot,1,2\}\}"'] \arrow[dr, "\{(\langle b,x,\mathrm{glo}\rangle,\langle b,x,\mathrm{loc}\rangle) \mid x\in\{\bot,1,2\}\}"] & \\
\langle \bot, 1, \mathrm{glo} \rangle & & \langle \bot, 2, \mathrm{glo} \rangle
\end{tikzcd}
$$

Before we define other global moves, let us point out that C_1 is very similar to rule C_1 of the game G_0 in Example 19.5.15 (observe the a and b in the first component and 1 and 2 in the second one).

(ii) Global move C_2 consists again of two triples:

$$
\begin{tikzcd}
 & \langle \bot, 1, \mathrm{glo} \rangle \arrow[dl, "\{(\langle a,x,\mathrm{glo},\rangle,\langle a,x,\mathrm{loc}\rangle) \mid x\in\{\bot,1,2\}\}"'] \arrow[dr, "\{(\langle b,x,\mathrm{glo},\rangle,\langle a,x,\mathrm{loc}\rangle) \mid x\in\{\bot,1,2\}\}"] & \\
\langle a, 1, \mathrm{loc} \rangle & & \langle a, 2, \mathrm{loc} \rangle
\end{tikzcd}
$$

and

$$
\begin{tikzcd}
 & \langle \bot, 2, \mathrm{glo} \rangle \arrow[dl, "\{(\langle a,x,\mathrm{glo},\rangle,\langle b,x,\mathrm{loc}\rangle) \mid x\in\{\bot,1,2\}\}"'] \arrow[dr, "\{(\langle b,x,\mathrm{glo},\rangle,\langle b,x,\mathrm{loc}\rangle) \mid x\in\{\bot,1,2\}\}"] & \\
\langle a, 1, \mathrm{loc} \rangle & & \langle a, 2, \mathrm{loc} \rangle
\end{tikzcd}
$$

(iii) Global move C_3 consists of the triples (where $z \in \{a,b\}$, i.e., $z \neq \perp$)

$$\begin{tikzcd}
& \langle\, z, 1, \mathrm{glo}\,\rangle \arrow[dl, "{\{(\langle a,x,\mathrm{glo}\rangle,\langle a,x,\mathrm{loc}\rangle) \mid x\in\{\perp,1,2\}\}}"'] \arrow[dr, "{\{(\langle b,x,\mathrm{glo}\rangle,\langle a,x,\mathrm{loc}\rangle) \mid x\in\{\perp,1,2\}\}}"] & \\
\langle\, z, 1, \mathrm{loc}\,\rangle & & \langle\, z, 2, \mathrm{loc}\,\rangle
\end{tikzcd}$$

and

$$\begin{tikzcd}
& \langle\, z, 2, \mathrm{glo}\,\rangle \arrow[dl, "{\{(\langle a,x,\mathrm{glo}\rangle,\langle b,x,\mathrm{loc}\rangle) \mid x\in\{\perp,1,2\}\}}"'] \arrow[dr, "{\{(\langle b,x,\mathrm{glo}\rangle,\langle b,x,\mathrm{loc}\rangle) \mid x\in\{\perp,1,2\}\}}"] & \\
\langle\, z, 1, \mathrm{loc}\,\rangle & & \langle\, z, 2, \mathrm{loc}\,\rangle
\end{tikzcd}$$

19.5.27. Lemma. *Every round of G_1 must have the following sequence of moves.*

$$\begin{array}{lllllllllll}
 & C_1; & & C_1; & & C_1; & \ldots & & C_1; & & (m\times\, C_1) \\
C_2; & 1; & C_3; & 2; & C_3; & 3; & \ldots & C_3; & m; & C_3; & \\
m+1; & C_3; & m+3; & C_3; & m+5; & C_3; & \ldots & & & &
\end{array}$$

That is, the game starts with m times a C_1 move (possibly $m = 0$ or $m = \infty$). After that, from the $m+1$-st position, the global and local moves alternate and the local move declared at every alternating step C_3 is executed exactly $2m+1$ steps later.

Proof. Exercise 19.6.29. ■

19.5.28. Definition. Let G_1 be as above and let $\mathcal{A} = \langle\, \Sigma^{\mathcal{A}}, Q, q_0, F^{\mathcal{A}}, \varrho\,\rangle$ be a typewriter automaton. We define a game $G^{\mathcal{A}}$ as follows.

(i) The alphabet of $G^{\mathcal{A}}$ is the cartesian product $\Sigma_1 \times Q \times (\Sigma^{\mathcal{A}} \cup \{\perp, \varepsilon\})$.

(ii) For each local move B of G_1 and each $\beta \in \Sigma^{\mathcal{A}} \cup \{\varepsilon\}$, we define a local move $B_\beta = \{(\langle\, a, q, \beta\,\rangle, \langle\, b, q, \perp\,\rangle) \mid q \in Q \text{ and } \langle\, a, b\,\rangle \in B\}$.

(iii) If $\Delta = \langle\, \langle\, B, b\,\rangle, \langle\, B', b'\,\rangle, c\,\rangle \in C_1 \cup C_2$, then we define Δ_β as the triple

$$\begin{tikzcd}
& \langle c, q_0, \perp\rangle \arrow[dl, "B_\beta"'] \arrow[dr, "B'_\beta"] & \\
\langle b, q_0, \perp\rangle & & \langle b', q_0, \perp\rangle
\end{tikzcd}$$

(iv) For each triple $\Delta = \langle \langle B, b \rangle, \langle B', b' \rangle, c \rangle \in C_3$ we define a set $C^{\mathcal{A}}_{\Delta}$ consisting of all triples of the form

$$\begin{tikzcd} & \langle c, q, \alpha \rangle \arrow[dl, "B_\beta"'] \arrow[dr, "B'_\beta"] & \\ \langle b, p, \bot \rangle & & \langle b', p, \bot \rangle \end{tikzcd}$$

where $\varrho(q, \alpha) = (p, \beta)$.

(v) Define
$$\begin{aligned} C^{\mathcal{A}}_1(\beta) &= \{\Delta_\beta \mid \Delta \in C_1\}, \qquad \text{for } \beta \in \Sigma^{\mathcal{A}}; \\ C^{\mathcal{A}}_2 &= \{\Delta_\varepsilon \mid \Delta \in C_2\}; \\ C^{\mathcal{A}}_3 &= \bigcup\{C^{\mathcal{A}}_\Delta \mid \Delta \in C_3\}. \end{aligned}$$

(vi) The initial symbol of $G^{\mathcal{A}}$ is $a = \langle a_1, q_0, \bot \rangle$, where $a_1 = \langle \bot, 1, G \rangle$, the initial symbol of G_1.

(vii) The set of final symbols is $A = \Sigma_1 \times (\Sigma^{\mathcal{A}} \cup \{\bot, \varepsilon\}) \times F^{\mathcal{A}}$;

(viii) Finally, we take $G^{\mathcal{A}} = \langle a, A, \{C^{\mathcal{A}}_1(\beta) \mid \beta \in \Sigma^{\mathcal{A}}\} \cup \{C^{\mathcal{A}}_2, C^{\mathcal{A}}_3\} \rangle$

19.5.29. Proposition. *Let $\mathcal{A}$ be a typewriter that accepts the language $L_{\mathcal{A}}$. Then*
$$L_{\mathcal{A}} \neq \emptyset \;\Leftrightarrow\; G^{\mathcal{A}} \textit{ is solvable.}$$
Hence $ETW \leq_T WTG$.

Proof. Our game $G^{\mathcal{A}}$ behaves as a "cartesian product" of G_1 and $\mathcal{A}$. Informally speaking, there is no communication between the first component and the other two. In particular we have the following.

1. Lemma 19.5.27 remains true with G_1 replaced by $G^{\mathcal{A}}$ and C_1, C_2, C_3 replaced respectively by $C^{\mathcal{A}}_1(\beta)$, $C^{\mathcal{A}}_2$ and $C^{\mathcal{A}}_3$. That is, a legitimate round of $G^{\mathcal{A}}$ must look as follows.

$$\begin{array}{llllllll} & & C^{\mathcal{A}}_1(\beta_1); & & C^{\mathcal{A}}_1(\beta_2); & \ldots & & C^{\mathcal{A}}_1(\beta_m); \\ & C^{\mathcal{A}}_2; & 1; & C^{\mathcal{A}}_3; & 2; & \ldots & C^{\mathcal{A}}_3; & m; \\ C^{\mathcal{A}}_3; & m+1; & C^{\mathcal{A}}_3; & m+3; & C^{\mathcal{A}}_3; & \ldots & & \end{array}$$

2. If a position T of $G^{\mathcal{A}}$ can be reached from the initial position, then the second and third component of labels are always the same for all nodes at every fixed level of T.

Consider a round of the game $G^{\mathcal{A}}$ as in 1 above. Note that this sequence is fully determined by the choice of m and $\beta_1, \ldots, \beta_m$. Also observe that β_i, for $i = 1, \ldots, m$ are the third components of the labels of all leaves of T_{m+2i}. Let w denote the word $\beta_1\beta_2 \ldots \beta_m$ and $O^{\mathcal{A}}(w)$ the 'opening' $C^{\mathcal{A}}_1(\beta_1), \ldots, C^{\mathcal{A}}_1(\beta_m)$ in the game $G^{\mathcal{A}}$.

Claim. Typewriter $\mathcal{A}$ accepts w iff $O^{\mathcal{A}}(w)$ leads to a winning position in $G^{\mathcal{A}}$.

We shall now prove the claim. Let β_j^k denote the symbol contained in the j-th cell of the tape of $\mathcal{A}$, after the machine has completed $k-1$ full phases (the tape has been fully scanned $k-1$ times). That is, β_j^k is the symbol to be read during the k-th phase. Of course β_j^1 is β_j. For uniformity write $\beta_{m+1}^k = \varepsilon$. Note that the $q_0, \ldots, q_m$ are renamed as $q_1^1, \ldots, q_{m+1}^1$ and the $\beta_1, \ldots, \beta_m, \epsilon$ as $\beta_1^1, \ldots, \beta_{m+1}^1$. Further, let q_j^k be the internal state of the machine, just before it reads the j-th cell for the k-th time (i.e., after $k-1$ full phases). The reader will easily show that for all k and all $j = 1, \ldots, m+1$ the following holds.

(i) The third component of labels of all leaves of $T_{(2k-1)(m+1)+2j-1}$ is β_j^k,

(ii) The second component of labels of all leaves of $T_{(2k-1)(m+1)+2j-2}$ is q_j^k.

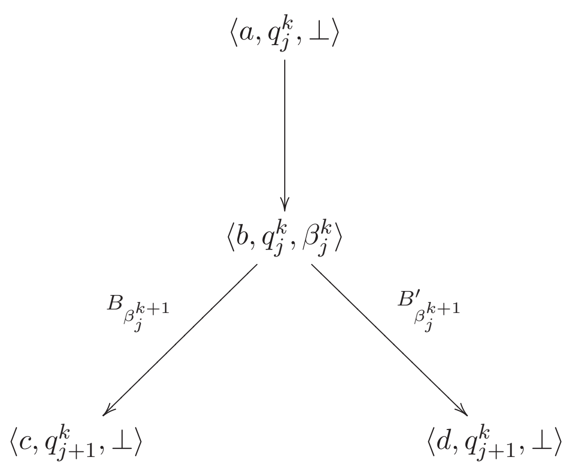

Figure 19.5: Simulation of a single machine step

Figure 19.5 illustrates the induction hypothesis by showing labels of a node at depth $(2k-1)(m+1)+2j-2$ together with her daughter and grandchildren. The claim follows when (ii) is applied to the final states.

It follows immediately from 1 and the claim that $L_{\mathcal{A}} \neq \emptyset$ iff there is a strategy to win $G^{\mathcal{A}}$. Hence the emptiness problem for typewriter automata can be reduced to the problem of winning tree games. ■

19.5.30. Theorem. *The inhabitation problem for $\lambda_\cap^{\mathrm{CDV}}$ is undecidable.*

Proof. By Corollary 19.5.23, Lemma 19.5.25 and Proposition 19.5.29. ■

Remarks

The proof of undecidability of the inhabitation problem presented in this section is a modified version of the original proof in Urzyczyn [1999].

The following notion of rank has been given for intersection types by Leivant [1983]. We denote it by *i-rank* in order to distinguish it from the notion defined in Definition **??**.

$$\begin{array}{rcll}
\text{i-rank}(A) & = & 0, & \text{for simple types } A; \\
\text{i-rank}(A \cap B) & = & \max(1, \text{i-rank}(A), \text{i-rank}(B)); & \\
\text{i-rank}(A \to B) & = & \max(1 + \text{i-rank}(A), \text{i-rank}(B)), & \text{if } \cap \text{ occurs in } A \to B.
\end{array}$$

It should be observed that all types in game contexts are of i-rank at most 3. Thus, the relativized inhabitation problem is undecidable for contexts of i-rank 3, and the inhabitation problem (with empty contexts) is undecidable for types of i-rank 4. It is an open problem whether inhabitation is decidable for i-rank 3. But it is decidable for i-rank 2 (Exercise 19.6.30). Some other decidable cases are discussed in Kurata and Takahashi [1995].

From the point of view of the the *formulae as types* principle, (the "Curry-Howard isomorphism"), the inhabitation problem should correspond to a provability problem for a certain logic. It is however not at all obvious what should be the logic corresponding to intersection types. We deal here with a "proof-functional" rather than "truth-functional" connective $\cap$, which is called sometimes "strong conjunction": a proof of $A \cap B$ must *be* a proof of both A and B, rather than merely *contain* two separate proofs of A and B. See Lopez-Escobar [n.d.], Mints [1989], and Alessi and Barbanera [1991] for the discussion of strong conjunction logics.

Various authors defined Church-style calculi for intersection types, which is another way leading to understand the logic of intersection types. We refer the reader to Venneri [1994], Dezani-Ciancaglini et al. [1997], Wells et al. [1997], Ronchi Della Rocca [2002], Liquori and Ronchi Della Rocca [2005], and Pimentel et al. [2005] for this approach.

19.6. Exercises

19.6.1. Show by means of examples that the type theories Plotkin and Engeler are not adequate.

19.6.2. Show that $\Sigma(\{\omega\}, \mathrm{CDV})$ is the smallest complete type theory (w.r.t. the order of Figure 15.3).

19.6.3.
1. The terms that are in PHN have the form $\lambda x_1 \ldots x_n.yM_1 \ldots M_k$ where $y \notin \{x_1, \ldots x_k\}$. Are these all of them? Is this enough to characterize them?
2. The terms that are in PN have the form $\lambda x_1 \ldots x_n.yM_1 \ldots M_k$ where $y \notin \{x_1, \ldots x_k\}$ and $M_1 \ldots M_k \in \mathsf{N}$. Are these all of them? Is this enough to characterize them?
3. Conclude that $\mathsf{PN} \subset \mathsf{PHN}$.

19.6.4. Show that PHN is HN-stable and PN is N-stable.

19.6.5. Construct a normal form X such that $x(\lambda y.a(y(\lambda z.b(yz)))[x := X]$ does not have a normal form.

19.6.6. Show that in the system induced by the type theory $\Sigma(\mathbb{A}^{\mathrm{CDZ}}, \mathrm{Scott} \cup \{(\omega\varphi), (\top \to \varphi)\})$, where

$$(\top \to \varphi) \quad \top \to \varphi \sim \varphi,$$

the terms typable with type φ in the context all whose predicates are ω are precisely the head normalizing ones.

19.6.7. Show that in the system induced by the type theory AO, the terms typable with type $\top \to \top$ for a suitable context are precisely the lazy normalizing ones, i.e. the terms which reduce either to an abstraction or to a (λ-free) head normal form.

19.6.8. Show that in the system induced by the type theory EHR the terms typable with type ν in the context all whose predicates are ν are precisely the terms which reduce either to an abstraction or to a variable using the call-by-value β-reduction rule.

19.6.9. Show that in the system induced by the type theory Park the terms typable with type ω in the empty context are precisely the closed terms.

19.6.10. Let TM be the term model of β-equality and $[M]$ the equivalence class of M under β-equality. Let a term M be persistently head normalizing iff $M\vec{N}$ has a head normal form for all terms $\vec{N}$ (see Definition 19.2.3). Prove that the type interpretation domain $\langle TM, \cdot, [\![\ ]\!], TM\rangle$, and the type environment $\xi_{\mathrm{Scott}}(\omega) = \{[M] \mid M \text{ is persistently head normalizing}\}$ agree with the type theory Scott.

19.6.11. Let TM and $[M]$ be as in Exercise 19.6.10. Prove that the type interpretation domain $\langle TM, \cdot, [\![\ ]\!], TM\rangle$, and the type environment $\xi_{\mathrm{Park}}(\omega) = \{[M] \mid M \text{ reduces to a closed term}\}$ agree with the type theory Park.

19.6.12. Let TM and $[M]$ be as in Exercise 19.6.10. Let a term M be persistently normalizing iff $M\vec{N}$ has a normal form for all normalizing terms $\vec{N}$ (see Definition 19.2.3). Prove that the type interpretation domain $\langle TM, \cdot, [\![\ ]\!], TM\rangle$, and the type environment $\xi_{CDZ}(\omega) = \{[M] \mid M \text{ is persistently normalizing}\}$, $\xi_{CDZ}(\varphi) = \{[M] \mid M \text{ is normalizing}\}$ agree with CDZ.

19.6.13. Show that for all $\mathcal{T}$, x, $A, B \in \mathbb{T}^{\mathcal{T}}$, $\psi \in \mathbb{A}^{\mathcal{T}}$, $A', B' \in \mathbb{T}^{Plo}$:

1. $\models_s^{\mathcal{T}} x : \top \to \top$;
2. $\not\models_i^{\mathcal{T}} x : \top \to \top$;
3. $\models_s^{\mathcal{T}} x : \nu$;
4. $\not\models_i^{\mathcal{T}} x : \nu$;
5. $x{:}\psi\cap^{\top}(A \to B) \models_F^{\mathcal{T}} \lambda y.xy : \psi$;
6. $x{:}\varphi\cap^{\top}(A' \to B') \not\models_i^{Plo} \lambda y.xy : \varphi$.

19.6.14. Show that if $\mathcal{S}$ is a natural type structure which is adequate for the F-semantics, then for all $n \geq 0$ and for all $\psi \in \mathbb{A}^{\mathcal{S}}$ we can find $I \neq \emptyset, A_i^{(1)}, \dots, A_i^{(n)}, B_i$ such that

$$\psi\cap^{\top}(\top^n \to \top) =_{\mathcal{S}} \bigcap_{i \in I}(A_i^{(1)} \to \dots \to A_i^{(n)} \to B_i).$$

19.6.15. Show that if $\mathcal{S}$ is a natural type structure which is adequate for the F-semantics, then the following rule (Dezani-Ciancaglini and Margaria [1986])

$$\text{(Hindley rule)} \quad \frac{\Gamma \vdash_{\cap^{\top}}^{\mathcal{S}} M : \psi\cap^{\top}(\top^n \to \top) \quad x_i \notin \mathrm{FV}(M)\ (1 \leq i \leq n)}{\Gamma \vdash_{\cap^{\top}}^{\mathcal{S}} \lambda x_1 \dots x_n.Mx_1 \dots x_n : \psi}$$

is admissible in $\lambda_\cap^{\mathcal{S}}$ for all $\psi \in \mathbb{A}^{\mathcal{S}}$. [Hint. Use the result of Exercise 19.6.14.]

19.6.16. Show that all type interpretation domains and all type environments agree with AO and with EHR.

19.6.17. Show that all quasi λ-models and all type environments preserve $\leq_{\mathrm{BCD}}$.

19.6.18. Show Proposition 19.4.11.

19.6.19. Show Theorem 19.4.13 using the mapping $\mathsf{m} : \mathsf{P}(\omega) \rightarrow \mathbb{T}^{\mathcal{T}(\omega)}$ defined as

$$\begin{array}{lcl} \mathsf{m}(\emptyset) & = & \top, \\ \mathsf{m}(\{i\}) & = & i, \\ \mathsf{m}(u \cup \{i\}) & = & \mathsf{m}(u) \cap^{\top} i. \end{array}$$

where $u \in \mathsf{P}_{\mathit{fin}}(\omega)$, $i \in \omega$.

19.6.20. Show Theorem 19.4.16 using the mapping $\mathsf{m} : \mathsf{P}_{\mathit{fin}}(\mathsf{Pm}) \rightarrow \Pi^{\mathsf{Pm}}$ defined as

$$\begin{array}{lcl} \mathsf{m}(\emptyset) & = & \top, \\ \mathsf{m}(\{\omega\}) & = & \omega, \\ \mathsf{m}(u \cup \{a\}) & = & \mathsf{m}(u) \cap^{\top} \mathsf{m}(\{a\}), \\ \mathsf{m}(\{\langle u, v \rangle\}) & = & \mathsf{m}(u) \rightarrow \mathsf{m}(v). \end{array}$$

where $u, v \in \mathsf{P}_{\mathit{fin}}(\mathsf{Pm})$, $a \in \mathsf{Pm}$.

19.6.21. Show Theorem 19.4.19 using the mapping $\mathsf{m} : \mathsf{P}_{\mathit{fin}}(\mathsf{Em}) \rightarrow \mathbb{T}^{\mathsf{Em}}$ defined as

$$\begin{array}{lcl} \mathsf{m}(\emptyset) & = & \top, \\ \mathsf{m}(\{\mathsf{a}\}) & = & \mathsf{a}, \\ \mathsf{m}(u \cup \{e\}) & = & \mathsf{m}(u) \cap^{\top} \mathsf{m}(\{e\}) \\ \mathsf{m}(\{\langle u, e \rangle\}) & = & \mathsf{m}(u) \rightarrow \mathsf{m}(\{e\}). \end{array}$$

where $u \in \mathsf{P}_{\mathit{fin}}(\mathsf{Em})$, $e \in \mathsf{Em}$, $\mathsf{a} \in \mathbb{A}_\infty$.

19.6.22. A term $(\lambda x.M)N$ is a $\boldsymbol{\beta}\mathsf{N}$-*redex* if $x \notin FV(M)$ or [N is either a variable or a closed SN (strongly normalizing) term] (Honsell and Lenisa [1999]). We denote by $\twoheadrightarrow_{\boldsymbol{\beta}\mathsf{N}}$ the induced reduction. Show that if Γ assigns types to all free variables in N, i.e. $x \in \mathrm{dom}(\Gamma)$ for all $x \in \mathrm{FV}(N)$, then

$$\Gamma \vdash_\cap^{\mathrm{HL}} M : A \;\&\; N \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{N}} M \;\Rightarrow\; \Gamma \vdash_\cap^{\mathrm{HL}} N : A.$$

[Hint: use Theorem 19.2.18(iii).]

19.6.23. Prove that the set $\{M \mid \exists \Gamma, A \,.\, \Gamma \vdash_{\cap^\top}^{\mathcal{T}} M : A\}$ is recursive. [Hint. This is obvious.]

19.6.24. Prove Lemma 19.5.2(i) and (ii). [Hint. Similar to the proof of Lemma 15.1.14.]

19.6.25. Consider the type assignment systems $\mathtt{K}$ and $\mathtt{K}_\top$ as defined in Exercises 15.5.2 and **??**.

(i) Prove an analogue of Lemma 19.5.8.

(ii) Prove that if Γ is a game context then $\Gamma \vdash^{\mathtt{K}} M : \alpha$ and $\Gamma \vdash^{\mathtt{K}_\top} M : \alpha$ are equivalent to $\Gamma \vdash M : \alpha$, for all type variables α. Conclude that type inhabitation remains undecidable without $(\leq)$.

(iii) Prove that the type $\delta \cap (\alpha \to \beta) \cap (\alpha \to \gamma) \to \delta \cap (\alpha \to \beta \cap \gamma)$ is inhabited in $\lambda_{\cap\top}^{\mathrm{BCD}}$ but is not inhabited in $\mathtt{K}_\top$.

19.6.26. Let Γ be a game context and $\alpha \in \mathbb{A}_\infty$. Prove that if $\Gamma \vdash M : \alpha$ then every node in the Böhm tree of M (as defined in Barendregt [1984]) has at most one branch.

19.6.27. Complete the proofs of Proposition 19.5.22 and Lemma 19.5.23.

19.6.28. Prove Lemma 19.5.25. [Hint. Encode a queue automaton (also called a Post machine, i.e. a deterministic finite automaton with a queue) into a typewriter, thus reducing the halting problem for queue automata to the emptiness problem for typewriters. One possible way of doing it is as follows. Represent a queue, say "011100010", as a string of the form

$$\$\$\cdots\$\text{<}011100010\text{>}\sharp\cdots\sharp\sharp,$$

with a certain number of the \$'s and $\sharp$'s. The initial empty queue is just "<>$\sharp\cdots\sharp\sharp$". Now an *insert* instruction means: *replace ">" with a digit and replace the first "$\sharp$" with ">"*, and similarly for a *remove*. The number of \$'s increases after each *remove*, while the suffix of $\sharp$'s shrinks after each *insert*, so that the queue "moves to the right". If the number of the initial suffix of $\sharp$'s is sufficiently large, then a typewriter automaton can verify the queue computation.]

19.6.29. Prove Lemma 19.5.27. [Hint. Compare G_1 to the game G_0 of Example 19.5.15. Observe that in each sequence of positions

$$T_{(2k+1)n}, \ldots, T_{(2k+3)n},$$

the odd steps behave as an initial phase of G_0, while the even steps behave as a final phase of G_0. Writing

$$\begin{array}{rcl}
\perp_i & = & \langle \perp, i, G \rangle; \\
A & = & \langle a, 1, \{G, L\} \rangle \langle a, 2, \{G, L\} \rangle; \\
B & = & \langle b, 1, \{G, L\} \rangle \langle b, 2, \{G, L\} \rangle
\end{array}$$

we have the following (the canope of a tree is the collection of its leaves) for the case of two initial C_1 steps.

position #	via move	canope of position
0		$\perp_1$
1	C_1	$\perp_1\perp_2$
2	C_1	$(\perp_1\perp_2)^2$
3	C_2	A^2A^2
4	1	A^2B^2
5	C_3	A^4B^4
6	2	$(A^2B^2)^2$
7	C_3	$(A^4B^4)^2$
8	3	$(A^2B^2)^4$
9	C_3	$(A^4B^4)^4$
10	5	$(A^2B^2)^8$
11	C_3	$(A^4B^4)^8$
12	7	$(A^2B^2)^{16}$
...	...	...
$4+2k$		$(A^2B^2)^{2^k}$

If one starts with m moves of C_1 $(0<m<\infty)$, then the canope of position $m+2+2k$ will be $(A^{2^{m-1}}B^{2^{m-1}})^{2^k}$. Note that $m=0$ or $m=\infty$ yield possible plays of the game.]

19.6.30. Prove that in $\lambda_\cap^{\mathrm{CDV}}$ the inhabitation problem for types of i-rank at most 2 is decidable. Here rank is defined as after Theorem 19.5.30. Note that a type of the shape $B\to C$ is of i-rank at most 2 iff it $B\to C\equiv A_1\to \ldots\to A_n$ and all A_i have i-rank at most 1.**Comment:** I cannot find a simple solution, we need to check the literature and ask Pawel.

19.6.31. [D. Kuśmierek]

(i) Let $\iota=(\alpha\to\beta)\cap(\beta\to\alpha)$ and define for $k=0,\ldots,n$ the type

$$A_k=\alpha\to\iota^k\to(\alpha\to\beta)\to(\beta\to\alpha)^{n-k}\to\beta.$$

Prove that the shortest inhabitant of $A_1,\ldots,A_n$ is of length exponential in n. Can you modify the example so that the shortest inhabitant is of double exponential length?

(ii)* How long (in the worst case) is the shortest inhabitant of a given type of rank 2, if it exists?

Comment: the actual further reading chapter is a waste of time and paper!

Bibliography

Abramsky, Samson [1991]. Domain theory in logical form, *Ann. Pure Appl. Logic* **51**(1-2), pp. 1–77. Second Annual IEEE Symposium on Logic in Computer Science (Ithaca, NY, 1987).

Abramsky, Samson and C.-H. Luke Ong [1993]. Full abstraction in the lazy lambda calculus, *Inform. and Comput.* **105**(2), pp. 159–267.

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

Alessi, Fabio [1993]. The p model, Internal Report, Udine University.

Alessi, Fabio and Franco Barbanera [1991]. Strong conjunction and intersection types, *in:* A. Tarlecki (ed.), *Proc. MFCS 1991*, LNCS 520, pp. 64–73.

Alessi, Fabio, Franco Barbanera and Mariangiola Dezani-Ciancaglini [2003]. Intersection types and computational rules, *in:* Ruy de Queiroz, Elaine Pimentel and Lucilia Figueiredo (eds.), *WoLLIC'03*, ENTCS 84, Elsevier.

Alessi, Fabio, Franco Barbanera and Mariangiola Dezani-Ciancaglini [2004]. Tailoring filter models, *in:* Stefano Berardi, Mario Coppo and Ferruccio Damiani (eds.), *Types '03*, Lecture Notes in Computer Science 3085, Springer-Verlag, pp. 17–33.

Alessi, Fabio, Franco Barbanera and Mariangiola Dezani-Ciancaglini [2005]. Intersection types and lambda models, *Theoret. Comput. Sci.* to appear.

Alessi, Fabio, Franco Barbanera and Mariangiola Dezani-Ciancaglini [2006]. Intersection Types and Lambda Models, *Theoretical Compututer Science.* to appear.

Alessi, Fabio, Mariangiola Dezani-Ciancaglini and Furio Honsell [2004]. Inverse limit models as filter models, *in:* Delia Kesner, Femke van Raamsdonk and J. B. Wells (eds.), *HOR '04*, RWTH Aachen, Aachen, pp. 3–25. ISSN 09353232, AIB-2004-03.

Alessi, Fabio, Mariangiola Dezani and Furio Honsell [2001]. Filter models and easy terms, *in:* Simona Ronchi and Antonio Restivo (eds.), *In ICTCS'01*, LNCS 2202, Springer, pp. 17–37.

Amadio, Roberto M. and Pierre-Louis Curien [1998]. *Domains and lambda-calculi*, Cambridge University Press, Cambridge.

Baeten, J. and B. Boerboom [1979]. ω can be anything it shouldn't be, *Indag.Math.* **41**, pp. 111–120.

van Bakel, Steffen [1992]. Complete restrictions of the intersection type discipline, *Theoret. Comput. Sci.* **102**(1), pp. 135–163.

van Bakel, Steffen, Franco Barbanera, Mariangiola Dezani-Ciancaglini and Fer-Jan de Vries [2002]. Intersection types for lambda-trees, *Theoret. Comput. Sci.* **272**(1–2), pp. 3–40.

Barbanera, Franco, Mariangiola Dezani-Ciancaglini and Ugo de'Liguoro [1995]. Intersection and union types: syntax and semantics, *Inform. and Comput.* **119**(2), pp. 202–230.

Barendregt, Henk and Sylvia Ghilezan [n.d.]. Lambda terms for natural deduction, sequent calculus and cut-elimination, *Journal of Functional Porgramming.* To appear.

Barendregt, Henk, Mario Coppo and Mariangiola Dezani-Ciancaglini [1983]. A filter lambda model and the completeness of type assignment, *J. Symbolic Logic* **48**(4), pp. 931–940 (1984).

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

Berline, Chantal [2000]. From computation to foundations via functions and application : the lambda-calculus and its webbed models, *Theoret. Comput. Sci.* **249**, pp. 81–161.

Böhm, Corrado and Mariangiola Dezani-Ciancaglini [1975]. λ-terms as total or partial functions on normal forms, pp. 96–121. Lecture Notes in Comput. Sci., Vol. 37.

Coppo, Mario and Mariangiola Dezani-Ciancaglini [1980]. An extension of the basic functionality theory for the λ-calculus, *Notre Dame J. Formal Logic* **21**(4), pp. 685–693.

Coppo, Mario, Mariangiola Dezani-Ciancaglini and Giuseppe Longo [1983]. Applicative information systems, *CAAP '83 (L'Aquila, 1983)*, Springer, Berlin, pp. 35–64.

Coppo, Mario, Mariangiola Dezani-Ciancaglini and Patrick Sallé [1979]. Functional characterization of some semantic equalities inside lambda-calculus., *in:* Hermann A. Maurer (ed.), *ICALP'79*, pp. 133–146.

Coppo, Mario, Mariangiola Dezani-Ciancaglini and B. Venneri [1980]. Principal type schemes and λ-calculus semantics, *To H. B. Curry: essays on combinatory logic, lambda calculus and formalism*, Academic Press, London, pp. 535–560.

Coppo, Mario, Mariangiola Dezani-Ciancaglini and B. Venneri [1981]. Functional characters of solvable terms, *Z. Math. Logik Grundlag. Math.* **27**(1), pp. 45–58.

Coppo, Mario, Mariangiola Dezani-Ciancaglini and Maddalena Zacchi [1987]. Type theories, normal forms, and D_∞-lambda-models, *Inform. and Comput.* **72**(2), pp. 85–116.

Coppo, Mario, Mariangiola Dezani-Ciancaglini, Furio Honsell and Giuseppe Longo [1984]. Extended type structures and filter lambda models, *Logic colloquium '82 (Florence, 1982)*, North-Holland, Amsterdam, pp. 241–262.

Curry, H.B. and R. Feys [1958]. *Combinatory Logic*, Studies in Logic and the Foundations of Mathematics I, North-Holland, Amsterdam.

Dezani-Ciancaglini, M., S. Ghilezan and B. Venneri [1997]. The "relevance" of intersection and union types, *Notre Dame Journal of Formal Logic* **38**(2), pp. 246–269.

Dezani-Ciancaglini, Maraingiola, Furio Honsell and Fabio Alessi [2003]. A complete characterization of complete intersection-type preorders, *ACM TOCL* **4**(1), pp. 120–147.

Dezani-Ciancaglini, Mariangiola and Ines Margaria [1986]. A characterization of F-complete type assignments, *Theoret. Comput. Sci.* **45**(2), pp. 121–157.

Dezani-Ciancaglini, Mariangiola, Furio Honsell and Yoko Motohama [2001]. Approximation theorems for intersection type systems, *Journal of Logic and Computation* **11**(3), pp. 395–417.

Dezani-Ciancaglini, Mariangiola, Furio Honsell and Yoko Motohama [2005]. Compositional characterization of λ-terms using intersection types, *Theoret. Comput. Sci.* **340**(3), pp. 459–495. An extended abstract appeared in *MFCS'00*, volume 1893 of *LNCS*, pages 304–313. Springer-Verlag, 2000.

Di Gianantonio, Pietro and Furio Honsell [1993]. An abstract notion of application, *Typed lambda calculi and applications (Utrecht, 1993)*, Springer, Berlin, pp. 124–138.

Egidi, Lavinia, Furio Honsell and Simona Ronchi Della Rocca [1992]. Operational, denotational and logical descriptions: a case study, *Fund. Inform.* **16**(2), pp. 149–169. Mathematical foundations of computer science '91 (Kazimierz Dolny, 1991).

Engeler, Erwin [1981]. Algebras and combinators, *Algebra Universalis* **13**(3), pp. 389–392.

Ghilezan, Silvia [1996]. Strong normalization and typability with intersection types, *Notre Dame J. Formal Logic* **37**(1), pp. 44–52.

Girard, Jean-Yves, Yves G. A. Lafont and Paul Taylor [1989]. *Proofs and Types*, Cambridge Tracts in Theoretical Computer Science 7, Cambridge University Press.

Hindley, J.Roger [1983]. The completeness theorem for typing λ-terms, *Theoret. Comput. Sci.* **22**, pp. 1–17.

Hindley, R. and G. Longo [1980]. Lambda-calculus models and extensionality, *Z. Math. Logik Grundlag. Math.* **26**(4), pp. 289–310.

Honsell, Furio and Marina Lenisa [1993]. Some results on the full abstraction problem for restricted lambda calculi, *Mathematical foundations of computer science 1993 (Gdańsk, 1993)*, Springer, Berlin, pp. 84–104.

Honsell, Furio and Marina Lenisa [1999]. Semantical analysis of perpetual strategies in λ-calculus, *Theoret. Comput. Sci.* **212**(1-2), pp. 183–209. Gentzen (Rome, 1996).

Honsell, Furio and Simona Ronchi Della Rocca [1992]. An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus, *J. Comput. System Sci.* **45**(1), pp. 49–75.

Hyland, Martin [1975/76]. A syntactic characterization of the equality in some models for the lambda calculus, *J. London Math. Soc. (2)* **12**(3), pp. 361–370.

Jacopini, Giuseppe [1975]. A condition for identifying two elements of whatever model of combinatory logic, *λ-calculus and computer science theory (Proc. Sympos., Rome, 1975)*, Springer, Berlin, pp. 213–219. Lecture Notes in Comput. Sci., Vol. 37.

Johnstone, Peter T. [1986]. *Stone spaces*, Cambridge University Press, Cambridge. Reprint of the 1982 edition.

Kozen, Dexter [1997]. *Automata and Computability*, Undergraduate Texts in Computer Science, Springer-Verlag, New York. first edition.

Krivine, Jean-Louis [1990]. *Lambda-calcul Types et modèles*, Masson, Paris.

Kurata, T. and M. Takahashi [1995]. Decidable properties of intersection type systems, *in:* G. Plotkin M. Dezani-Ciancaglini (ed.), *Proc. Typed Lambda Calculi and Applications '95*, LNCS, Springer, pp. 297–311.

Leivant, D. [1983]. Polymorphic type inference, *Proc. of 10-th ACM Symposium on Principles of Programming Languages*, Austin, Texas, pp. 88–98.

Leivant, Daniel [1986]. Typing and computational properties of lambda expressions, *Theoret. Comput. Sci.* **44**(1), pp. 51–68.

Liquori, Luigi and Simona Ronchi Della Rocca [2005]. Towards an intersection typed system a la Church, *in:* Mario Coppo and Ferruccio Damiani (eds.), *Workshop on Intersection Types and Related Systems*, ENTCS 136, Elsevier.

Longo, G. [1987]. On Church's formal theory of functions and functionals, *Technical Report 9*, Department of Computing Science, Pisa University.

Longo, Giuseppe [1983]. Set-theoretical models of λ-calculus: theories, expansions, isomorphisms, *Ann. Pure Appl. Logic* **24**(2), pp. 153–188.

Lopez-Escobar, E.G.K. [n.d.]. Proof functional connectives, *Proc. Methods in Mathematical Logic*, number 1130 in *LNiM*, Berlin, pp. 208–221.

Meyer, Albert R. [1982]. What is a model of the lambda calculus?, *Inform. and Control* **52**(1), pp. 87–122.

Mints, G. E. [1989]. The completeness of provable realizability, *Notre Dame J. Formal Logic* **30**, pp. 420–441.

Mitchell, John [1988]. Polymorphic type inference and containment, *Inform. and Comput.* **76**, pp. 211–249.

Mitchell, John [1996]. *Foundation for Programmimg Languages*, MIT Press.

Mitschke, Gerd [1976]. λ-kalkül, δ-konversion und axiomatische rekursionstheorie, *Technical Report Preprint 274*, Technische Hochschule, Darmstadt.

Park, D. [1976]. The Y-combinator in Scott's λ-calculus models (revised version), Theory of Computation Report 13, Department of Computer Science, University of Warick.

Pimentel, Elaine, Simona Ronchi Della Rocca and Luca Roversi [2005]. Intersection types: a proof-theoretical approach, To appear in Proceedings of STRUCTURES AND DEDUCTION - ICALP Workshop Lisbon July 16-17.

Plotkin, G. D. [1975]. Call-by-name, call-by-value and the λ-calculus, *Theoret. Comput. Sci.* **1**(2), pp. 125–159.

Plotkin, Gordon D. [1993]. Set-theoretical and other elementary models of the λ-calculus, *Theoret. Comput. Sci.* **121**(1-2), pp. 351–409. A collection of contributions in honour of Corrado Böhm on the occasion of his 70th birthday.

Pottinger, G. [1981]. The church-rosser theorem for the typed λ-calculus with surjective pairing, *Notre Dame J. Formal Logic* **22**(3), pp. 264–268.

Pottinger, Garrel [1980]. A type assignment for the strongly normalizable λ-terms, *To H. B. Curry: essays on combinatory logic, lambda calculus and formalism*, Academic Press, London, pp. 561–577.

Pravato, A., S. Ronchi and L. Roversi [1999]. The call-by-value lambda calculus: a semantic investigation, *Mathematical structures in computer science* **9**(5), pp. 617–650.

Ronchi Della Rocca, Simona [1988]. Lecture notes on semantics and types. Internal Report, Torino University.

Ronchi Della Rocca, Simona [2002]. Intersection typed lambda-calculus, *ITRS 2002*, ENTCS 70.1, Elsevier.

Ronchi Della Rocca, Simona and Luca Paolini [2004]. *THE PARAMETRIC LAMBDA CALCULUS A Metamodel for Computation*, Texts in Theoretical Computer Science. An EATCS Series XIII, Springer-Verlag.

Scott, Dana [1972]. Continuous lattices, *Toposes, algebraic geometry and logic (Conf., Dalhousie Univ., Halifax, N. S., 1971)*, Springer, Berlin, pp. 97–136. Lecture Notes in Math., Vol. 274.

Scott, D.S. [1975]. Open problem, *Lambda Calculus and Computer Science Theory*, Lecture Notes in Computer Science 37, Springer, Berlin, p. 369.

Severi, Paula [1996]. *Normalisation in lambda calculus and its relation to type inference*, Dissertation, Eindhoven University of Technology.

Statman, Richard [1979]. The typed λ-calculus is not elementary recursive, *Theoret. Comput. Sci.* **9**(1), pp. 73–81.

Urzyczyn, Pawel [1999]. The emptiness problem for intersection types, *J. Symbolic Logic* **64**(3), pp. 1195–1215.

Venneri, B. [1994]. Intersection types as logical formulae, *J. Logic Computat.* **4**(2), pp. 109–124.

Venneri, Betti [1996]. Private communication, Florence University.

Wadsworth, Christopher P. [1976]. The relation between computational and denotational properties for Scott's D_∞-models of the lambda-calculus, *SIAM J. Comput.* **5**(3), pp. 488–521. Semantics and correctness of programs.

Wells, J.B., A. Dimock, R. Muller and F. Turbak [1997]. A typed intermediate language for flow-directed compilation, *Proc. 7th International Joint Conference on the Theory and Practice of Software Development, Lille, France*, number 1214 in *LNCS*, pp. 757–771.

Zylberajch, Cécile [1991]. *Syntaxe et Semantique de la Facilité en Lambda-Calcul.*, Dissertation, Université de Paris VII.