INFORMATION AND COMPUTATION 119, 202–230 (1995)

Intersection and Union Types: Syntax and Semantics

FRANCO BARBANERA AND MARIANGIOLA DEZANI-CIANCAGLINI

Dipartimento di Informatica dell' Università di Torino, Corso Svizzera, 185, 10149 Turin, Italy
E-mail: barba@di.unito.it, dezani@di.unito.it

AND

UGO DE'LIGUORO

Dipartimento di Informatica dell' Università di Roma "La Sapienza", Via Salaria, 113, 00198 Rome, Italy
E-mail: deligu@dsi.uniroma1.it

Type assignment systems with *intersection* and *union* types are introduced. Although the subject reduction property with respect to β-reduction does not hold for a natural deduction-like system, we manage to overcome this problem in two, different ways. The first is to adopt a notion of parallel reduction, which is a refinement of Gross–Knuth reduction. The second is to introduce type theories to refine the system, among which is the theory called Π that induces an assignment system preserving β-reduction. This type assignment system further clarifies the relation with the intersection discipline through the decomposition, first, of a disjunctive type into a set of conjunctive types and, second, of a derivation in the new type assignment system into a set of derivations in the intersection type assignment system. For this system we propose three semantics and prove soundness and completeness theorems.

INTRODUCTION

Intersection types were introduced in (Coppo *et al.*, 1981) as a generalization of the Curry type inference system, mainly with the aim of describing the functional behaviour of all solvable λ-terms. In the intersection type discipline the usual "$\rightarrow$"-based type language for λ-calculus was extended by adding a constant ω (the universal type) and a new connective "$\wedge$" whose intended meaning is intersection of two types. This type language, together with suitable axioms and rules, allows the system of (Coppo *et al.*, 1981) to have the following properties:

(i) the set of types given to a λ-term is invariant under β-conversion;

(ii) the sets of solvable and normalizing λ-terms can be characterized very neatly by the types of their members.

Moreover, this system was proved to be sound and compete with respect to the standard semantics (Barendregt *et al.*, 1983; Hindley, 1983).

The aim of the present paper is foundational: to study the intersection type discipline enriched with a new type constructor "$\vee$" whose intended meaning is *union* of types, allowing for a sort of disjunction. A type constructor of this kind was first introduced in (MacQueen *et al.*, 1986) in connection with a proposal for interpreting types as ideals. The rules considered in that paper resemble the natural deduction rules for disjunction, hinting at an instance of the Curry–Howard "formulas-as-types" analogy.

However, as in the case of intersection type discipline, there is not a real analogy. Indeed, in the extensions of Curry's type assignment system that are discussed in this paper, λ-terms cannot be seen as encodings of (intuitionistic) proofs of (the formulas corresponding to) their types, since they do not reflect the structure of the derivations. Instead they can be seen as models of their types, and types as properties of applicative structures in which terms are interpreted. This does not obstruct having a good formulation of the basic type assignment system, which can be still presented in a Gentzen style, both as a natural deduction calculus and as a sequent calculus. This is essential when proof-theoretical properties of the system are investigated, a study not carried on in (MacQueen *et al.*, 1986).

When disjunction is considered, a fundamental property of typed languages, namely subject reduction, is not preserved. This is a serious problem as much as rules for $\vee$, especially the rule for its elimination, are quite natural. To overcome this difficulty we propose two possible solutions.

The first consists in introducing a simple kind of parallel reduction. This takes care of the fact that one can substitute for a variable, for which both types σ and τ have been proved to be safely assumed, a term of type $\sigma \vee \tau$. The term could contain a redex and the substitution does not commute with the reduction of such a redex unless one reduces in parallel the copies of the redex produced by the substitution. This implies that subject reduction holds for the Gross–Knuth reduction (a kind of Kleene rule), which is the most powerful among the parallel reductions. As immediate consequence we obtain that well-typed programs cannot encounter run-time type failures.

0890-5401/95 $6.00

The second solution will be provided by suitable type inclusions, which take the intended interpretation of type constructors into account. We will introduce the notion of type theory, as a set of type inclusions; each type theory induces a type assignment system through the subtyping rule

$$\frac{B \vdash M : \sigma}{B \vdash M : \tau} \quad \text{if } \sigma \text{ is a subtype of } \tau.$$

A statement turns out to be derivable in the basic type assignment system if and only if it is derivable in the system induced by the minimal type theory.

A natural type theory, called Ξ, is justified by the interpretation of ω as the universe, of $\rightarrow$ as function space constructor, of $\wedge$ as intersection, and of $\vee$ as union of sets. The type assignment system induced by Ξ makes it possible to derive more statements than the basic one. However, types are still not invariant under β-reduction of subjects.

To get a more powerful system we will define another type theory, called Π, which is justified by interpreting types as upward closed subsets of a Scott domain. In fact, this is the type inclusion considered in (Abramsky, 1991) where types are interpreted as *compact-open* subsets. We will prove that the type assignment system induced by Π with intersection and union types still satisfies the above properties (i) and (ii). The proof will be carried out by associating to each type a finite set of intersection types and to each deduction in the full system a set of deductions in the intersection type assignment system. Moreover, we will prove an approximation theorem, stating that the types of a λ-term are exactly the types of its approximants.

A natural way of defining the semantics of a type assignment system is to interpret the underlying untyped language in a domain D, and the types as subsets of D, in such a way that the interpretation of a λ-term belongs to the interpretation of every type which can be assigned to that λ-term in the system. The intended meanings of type constructions naturally lead to a first definition of type interpetation.

We will introduce a more refined notion of type interpretation inspired by Beth–Kripke models for intuitionistic propositional logic (Beth, 1965; Kripke, 1959). We will consider models where knowledge can expand, i.e., where there are different worlds. In principle, there are incompatible ways of extending knowledge represented as a tree of possibilities; this tree will represent a partially ordered set of possible worlds, and a branch in such a tree will determine a possible history of knowledge in that model. What we will do is to make essentially sure that whenever a λ-term M has a union type $\sigma \vee \tau$ in a given world w, then M will eventually have either σ or τ in every branch going through w (not necessarily the same one for every branch). We shall prove soundness and completeness of the type assignment system induced by Π with respect to this notion of type interpretation.

Next to this, two other notions of type interpretation will also be considered, the former inspired by the second order interpretation of the disjunction as given in logic, the latter inspired by a translation of types into sets of types without unions. The type assignment system induced by Π will be proved to be sound and complete also with respect to both these type interpretations.

As by-products of the above results we will obtain interesting properties for the type assignment system induced by any type theory (also, therefore, for the basic type assignment system). In fact, all these systems are sound with respect to the first three notions of type interpretation. Moreover, all the systems induced by type theories which are included in Π characterize the sets of solvable and normalizing λ-terms, and they are also sound with respect to the last notion of type interpretation.

Section 1 will present the basic system both in a natural deduction and in a sequent calculus style; these formalisms will be proved to be equivalent. Section 2 will introduce the notion of parallel reduction and prove that types in the basic system are invariant under such a reduction relation. Section 3 will define type theories that induce type assignment systems, whose proof theoretical properties are investigated in Section 4. Finally, Section 5 will introduce three possible semantics establishing soundness and completeness results.

(Barbanera and Dezani–Ciancaglini, 1991) contains a first version of Sections 4 and 5 of the present paper.

1. THE BASIC TYPE ASSIGNMENT SYSTEM

In this section we will define the basic system for which we will give two formulations: the former in natural deduction style and the latter in sequent calculus style. These formulations will be proved to be equivalent.

The set of types we consider is built out of an infinite set of type variables and the type constant ω, by means of the function space- ("$\rightarrow$"), union- ("$\vee$"), and intersection- ("$\wedge$") type constructors.

1.1. Definition (Types). The set $\mathbf{T}$ of *types* is inductively defined by

— $\phi_0, \phi_1, \ldots \in \mathbf{T}$ (type variables)
— $\omega \in \mathbf{T}$ (type constant)
— if $\sigma, \tau \in \mathbf{T}$, then $(\sigma \rightarrow \tau)$, $(\sigma \wedge \tau)$, $(\sigma \vee \tau) \in \mathbf{T}$.

Conventionally, we omit parentheses according to the precedence rule: "$\wedge$ and $\vee$ over $\rightarrow$".

A *typing statement* is an expression of the form $M : \sigma$, where M is a λ-term and σ a type; M is called the *subject* and σ the *predicate* of the typing statement. A *basic* typing statement is a typing statement whose subject is a variable.

A *basis* is a set of basic typing statements such that subjects are pairwise distinct. If B is a basis, then $FV(B)$ will denote the set of term variables which are subjects of basic type statements in B. $B, x:\sigma$ will denote the basis $B \cup \{x:\sigma\}$, when B is a basis such that either $x \notin FV(B)$, or $x:\sigma \in B$.

A *statement* is an expression of the form $B \vdash M:\sigma$, where B is a basis and $M:\sigma$ is a typing statement. We call B the assumptions, and $M:\sigma$ the succedent of $B \vdash M:\sigma$.

1.2. DEFINITION (Natural Deduction Formulation: $\mathfrak{N}$). The axioms and rules to derive statements in the system $\mathfrak{N}$ are

$$(Ax)\ \frac{}{B, x:\sigma \vdash x:\sigma}$$

$$(\omega)\ \frac{}{B \vdash M:\omega}$$

$$(\rightarrow E)\ \frac{B \vdash M:\sigma\rightarrow\tau \quad B \vdash N:\sigma}{B \vdash MN:\tau}$$

$$(\rightarrow I)\ \frac{B, x:\sigma \vdash M:\tau}{B \vdash \lambda x.M:\sigma\rightarrow\tau}$$

$$(\wedge I)\ \frac{B \vdash M:\sigma \quad B \vdash M:\tau}{B \vdash M:\sigma\wedge\tau}$$

$$(\wedge E)\ \frac{B \vdash M:\sigma\wedge\tau}{B \vdash M:\sigma} \qquad \frac{B \vdash M:\sigma\wedge\tau}{B \vdash M:\tau}$$

$$(\vee I)\ \frac{B \vdash M:\sigma}{B \vdash M:\sigma\vee\tau} \qquad \frac{B \vdash M:\tau}{B \vdash M:\sigma\vee\tau}$$

$$(\vee E)\ \frac{B, x:\sigma \vdash M:\rho \quad B, x:\tau \vdash M:\rho \quad B \vdash N:\sigma\vee\tau}{B \vdash M[N/x]:\rho}.$$

We write $B \vdash_{\mathfrak{N}} M:\sigma$ to express that the statement $B \vdash M:\sigma$ is derivable in $\mathfrak{N}$. We note that, in this system, weakening is an admissible rule; if $B \vdash_{\mathfrak{N}} M:\sigma$, and $B' \supseteq B$, then $B' \vdash_{\mathfrak{N}} M:\sigma$.

As stated in the Introduction, the natural deduction formulation of the basic system is actually the system of (MacQueen *et al.*, 1986), restricted to the type constructors $\rightarrow$, $\wedge$, $\vee$.

This presentation looks familiar and intuitive, as it shares with natural deduction systems the nice property of defining the "meaning" of type operators directly, through introduction-elimination rules.

However, if one is interested in the proof-theoretical properties of the system, it can be useful to reformulate it in a sequent calculus style. We therefore introduce a type assignment system below whose rules are introductions of type constructors to the left and to the right. Besides technical motivations for doing so, we think that this system deserves interest on its own.

The resulting system is not a pure sequent calculus, since we still consider bases as *sets* of premises, in order to avoid in proofs the boring treatment of structural rules. We preserve, however, the "multiplicative" character of the rules, a typical feature of Gentzen's original calculus.

A *sequent* is an expression of the form $B :- M:\sigma$. We write B, B' to mean that $B \mathbin{\dot\cup} B'$ is still a basis; i.e., if $x \in FV(B) \cap FV(B')$, then there is a unique σ such that $x:\sigma \in B$ and $x:\sigma \in B'$.

1.3. DEFINITION (Sequent Calculus Formulation: $\mathfrak{S}$). The axioms and rules to derive sequents in the system $\mathfrak{S}$ are

$$\frac{}{B, x:\sigma :- x:\sigma}\,Ax \qquad \frac{}{B :- M:\omega}\,\omega$$

$$\frac{B, x:\tau :- M:\rho \quad B' :- N:\sigma}{B, B', y:\sigma\rightarrow\tau :- M[yN/x]:\rho} \rightarrow L$$

$$\frac{B, x:\sigma :- M:\tau}{B :- \lambda x.M:\sigma\rightarrow\tau} \rightarrow R$$

$$\frac{B, x:\sigma :- M:\rho}{B, x:\sigma\wedge\tau :- M:\rho} \qquad \frac{B, x:\tau :- M:\rho}{B, x:\sigma\wedge\tau :- M:\rho} \wedge L$$

$$\frac{B :- M:\sigma \quad B' :- M:\tau}{B, B' :- M:\sigma\wedge\tau} \wedge R$$

$$\frac{B, x:\sigma :- M:\rho \quad B', x:\tau :- M:\rho}{B, B', x:\sigma\vee\tau :- M:\rho} \vee L$$

$$\frac{B :- M:\sigma}{B :- M:\sigma\vee\tau} \qquad \frac{B :- M:\tau}{B :- M:\sigma\vee\tau} \vee R$$

$$\frac{B, x:\sigma :- M:\tau \quad B' :- N:\sigma}{B, B' :- M[N/x]:\tau}\,\text{cut}$$

Recall that, by the notational convention just before this definition, the set of assumptions in the conclusion of each rule has to be actually a basis: each term variable may occur at most once. Note that there is no loss of generality in this restriction. If $B_1 :- M:\sigma$ and $B_2 :- N:\tau$, using $(\wedge L)$ we can always build B_1' and B_2' such that $B_1' :- M:\sigma$, $B_2' :- N:\tau$, and B_1', B_2' is a basis. More precisely, B_1' and B_2' will contain $x:\rho\wedge\upsilon$ whenever $x:\rho \in B_1$ and $x:\upsilon \in B_2$.

Notation.

(i) We write $B :-_{\mathfrak{S}} M:\sigma$ to express that the sequent $B :- M:\sigma$ is derivable in $\mathfrak{S}$.

(ii) If $\mathfrak{D}$ is a derivation showing that $B :-_{\mathfrak{S}} M:\sigma$, we write $\mathfrak{D} : B :-_{\mathfrak{S}} M:\sigma$.

(iii) We denote an arbitrary rule of the sequent calculus by

$$\frac{B_i :- M:\sigma_i}{B :- N:\tau}\ \text{rule},$$

assuming that the index i will have the correct value (i.e., $i=0$, $i=1$, or $i=1,2$) and that the terms and types involved will match the rules as in Definition 1.3.

The proof of the equivalence between the systems $\mathfrak{N}$ and $\mathfrak{S}$ follows a standard pattern and will be given in Theorem 1.5.

1.4. Lemma (Substitution Lemma). *If $B, x:\sigma \vdash_{\mathfrak{N}} M:\tau$ and $B \vdash_{\mathfrak{N}} N:\sigma$, then $B \vdash_{\mathfrak{N}} M[N/x]:\tau$.*

Proof. This can be proved, as usual for Curry type systems, by induction on the derivation of $B, x:\sigma \vdash M:\tau$. However, in the present setting we can shorten the proof as follows:

$$(\vee E)\,\frac{B, x:\sigma \vdash M:\tau \quad B, x:\sigma \vdash M:\tau \quad (\vee I)\,\dfrac{B \vdash N:\sigma}{B \vdash N:\sigma \vee \sigma}}{B \vdash M[N/x]:\tau}$$

∎

1.5. Theorem (Equivalence between $\mathfrak{N}$ and $\mathfrak{S}$). *For any basis B, term M, and type σ,*

$$B \vdash_{\mathfrak{N}} M:\sigma \text{ if and only if } B :\!-_{\mathfrak{S}} M:\sigma.$$

Proof. ($\Rightarrow$) By induction on the derivation of $B \vdash_{\mathfrak{N}} M:\sigma$. We only consider the interesting cases, i.e., the elimination rules.

Case ($\to E$).

$$(\to E)\,\frac{B \vdash M:\sigma \to \tau \quad B \vdash N:\sigma}{B \vdash MN:\tau}$$

becomes

$$\frac{\dfrac{\dfrac{}{x:\tau :\!- x:\tau}\,Ax \qquad \dfrac{\text{ind. hyp.}}{B :\!- N:\sigma}}{B, y:\sigma\to\tau :\!- yN:\tau}\,\to L \qquad \dfrac{\text{ind. hyp.}}{B :\!- M:\sigma\to\tau}}{B :\!- MN:\tau}\,\text{cut}$$

Case ($\wedge E$).

$$(\wedge E)\,\frac{B \vdash M:\sigma \wedge \tau}{B \vdash M:\sigma}$$

becomes

$$\frac{\dfrac{\dfrac{}{x:\sigma :\!- x:\sigma}\,Ax}{x:\sigma\wedge\tau :\!- x:\sigma}\,\wedge L \qquad \dfrac{\text{ind. hyp.}}{B :\!- M:\sigma\wedge\tau}}{B :\!- M:\sigma}\,\text{cut}$$

Case ($\vee E$).

$$(\vee E)\,\frac{B, x:\sigma \vdash M:\rho \quad B, x:\tau \vdash M:\rho \quad B \vdash N:\sigma \vee \tau}{B \vdash M[N/x]:\rho}$$

becomes

$$\frac{\dfrac{\dfrac{\text{ind. hyp.}}{B, x:\sigma :\!- M:\rho} \qquad \dfrac{\text{ind. hyp.}}{B, x:\tau :\!- M:\rho}}{B, x:\sigma\vee\tau :\!- M:\rho}\,\vee L \qquad \dfrac{\text{ind. hyp.}}{B :\!- N:\sigma\vee\tau}}{B :\!- M[N/x]:\rho}\,\text{cut}$$

($\Leftarrow$) By induction on the derivation of $B :\!-_{\mathfrak{S}} M:\sigma$. The proof is immediate in almost all cases. In case of cut, use the substitution lemma. The other nontrivial cases are $\to L$, $\wedge L$, and $\vee L$.

Case $\to L$.

$$\frac{B, x:\tau :\!- M:\rho \quad B' :\!- N:\sigma}{B, B', y:\sigma\to\tau :\!- M[yN/x]:\rho}\,\to L$$

We can freely assume that $x \notin FV(B')$. From the induction hypothesis and the admissibility of weakening we have

$$B, B', y:\sigma\to\tau, x:\tau \vdash M:\rho,$$

and

$$B, B', y:\sigma\to\tau \vdash N:\sigma.$$

From this, by (Ax) and ($\to E$), we get $B, B', y:\sigma\to\tau \vdash yN:\sigma$; hence the thesis follows by the substitution lemma.

Case $\wedge L$.

$$\frac{B, x:\sigma :\!- M:\rho}{B, x:\sigma\wedge\tau :\!- M:\rho}\,\wedge L$$

By induction $B, x:\sigma \vdash M:\rho$. So we can obtain a derivation of $B, x:\sigma\wedge\tau \vdash M:\rho$ simply by replacing in a derivation for $B, x:\sigma \vdash M:\rho$ every occurrence of the axiom

$$(Ax)\,\frac{}{B, x:\sigma \vdash x:\sigma}$$

by

$$(\wedge E)\,\frac{(Ax)\,\dfrac{}{B, x:\sigma\wedge\tau \vdash x:\sigma\wedge\tau}}{B, x:\sigma\wedge\tau \vdash x:\sigma}$$

Case $\vee L$.

$$\frac{B, x:\sigma :\!- M:\rho \quad B', x:\tau :\!- M:\rho}{B, B', x:\sigma\vee\tau :\!- M:\rho}\,\vee L$$

By induction $B, x:\sigma \vdash M:\rho$; hence, if y is fresh, $B, B', x:\sigma \vee \tau, y:\sigma \vdash M[y/x]:\rho$ by weakening (notice that deductions are independent of variables names). Similarly, $B, B', x:\sigma \vee \tau, y:\tau \vdash M[y/x]:\rho$. The thesis follows from:

$$(\vee E)\,\frac{B', x:\sigma \vee \tau, y:\sigma \vdash M[y/x]:\rho \quad B', x:\sigma \vee \tau, y:\tau \vdash M[y/x]:\rho \quad B', x:\sigma \vee \tau \vdash x:\sigma \vee \tau}{B', x:\sigma \vee \tau \vdash M:\rho} \blacksquare$$

1.6. *Remark.* Intersection does not correspond to intuitionistic conjunction in the standard formulas-as-types analogy, as already noted in (Hindley, 1984). A simple counter-example is the type $\sigma \to \tau \to \sigma \wedge \tau$ (an axiom of the Hilbert system of propositional logic), which cannot be deduced for any closed λ-term. Similarly, union does not correspond to intuitionistic disjunction; a counter-example is again given by an axiom of the Hilbert system: $(\sigma \to \tau) \to (\rho \to \tau) \to \sigma \vee \rho \to \tau$. These are all consequences of the fact that $\to$ is not the right adjoint of $\wedge$; in any case we can deduce that $\vdash \lambda xy.xy : (\sigma \to \tau) \wedge (\rho \to \tau) \to \sigma \vee \rho \to \tau$.

(Dezani *et al.*, 1992) presented a relevance logic which restored the formulas-as-types analogy with respect to the assignment system $\vdash^{\Pi}$, to be introduced in Section 3.

2. SYNTACTIC PROPERTIES OF THE BASIC SYSTEM: INVARIANCE OF TYPES UNDER PARALLEL REDUCTION OF SUBJECTS

The type assignment systems $\mathfrak{N}$ and $\mathfrak{S}$ are not invariant under β-reduction of subjects; this means that from $B \vdash_{\mathfrak{N}} M:\sigma$ and $M \twoheadrightarrow_{\beta} N$ we cannot infer that $B \vdash_{\mathfrak{N}} N:\sigma$. The problem is that, because of rule $(\vee E)$, we lose the correspondence between subterms and subdeductions; many occurrences of the same subterm correspond in fact to a unique subdeduction. For example, one can deduce the type

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

for both $\lambda xyz.x((yz)(yz)$ and $\lambda xyz.x((\lambda t.t)\,yz)((\lambda t.t)\,yz)$, but this type can be deduced neither for $\lambda xyz.x(yz)((\lambda t.t)\,yz)$; nor for $\lambda xyz.x((\lambda t.t)\,yz)(yz)$ (Pierce, 1991b). Thus in this system types are preserved neither under β-reduction, nor under β-expansion. A similar example shows that types are not preserved under η-reduction.

Invariance of typing under β-reduction is a desirable property, giving us, among other things, the feeling that, in a sense, the rules of the system are correct. In the present case, however, rule $(\vee E)$ can hardly be considered incorrect, notwithstanding the lack of invariance of typing under β-reduction, since it is sound for all the semantics that will be introduced in Section 5.

One could then argue that the notion of β-reduction itself is not the right notion of reduction for our system, as rule $(\vee E)$, by containing a substitution which can duplicate redexes, suggests. Indeed, we will prove system $\mathfrak{S}$ to be invariant under *parallel* β-reductions and then, by the equivalence of $\mathfrak{N}$ and $\mathfrak{S}$, the same property will be established for $\mathfrak{N}$. More precisely, typing is invariant under β-reductions which are performed simultaneously on all the occurrences of the same subterm, which corresponds to the same subdeduction.

Our proof is inspired by Gentzen's proof of the Haupsatz (Gentzen, 1969). Following his argument, we introduce the notions of rank and degree of cuts, and use them to show that the cut elimination procedure terminates. We do not eliminate all cuts but only those generating the β-redexes we want to reduce. Indeed, there are cuts which cannot be avoided, but these cuts never generate β-redexes.

The main line of our proof is as follows. First we define in our context the standard notion of rank and readiness of a cut. Then we define parallel reductions as the reductions which are always balanced, in the sense that all occurrences of the same redex are reduced in parallel. This will be formalized through the notion of uniform set of redexes: essentially, a set is uniform if it contains either all occurrences of a redex or none. What we really will prove is invariance of types under the complete development of an arbitrary uniform set of redexes. To achieve this, we will introduce a notion of cut degree which is 0 only when the cut does not generate a redex belonging to the current uniform set of redexes. Otherwise, the cut degree is the degree of the cut type. Our definition of degree of a type is the standard one, except for the case of an arrow type; in fact, we want the degree of an arrow type to be minimal. This is due to the fact that union and intersection types are proof-functional connectives, in the sense of (Lopez-Escobar, 1985). In particular, rules $\wedge R$ and $\vee L$ require the same subject in both premises. This implies that cuts whose cut type is an arrow type have to be eliminated all together and after any other cut whose cut type has a "logical" connective as its main operator (cf. also Remark 2.9).

Thanks to the above machinery, the proofs of the Rank Lemma and of the Cut Elimination Theorem follow essentially the standard pattern.

Usually, the cut elimination proves the (weak) normal form theorem for λ-terms having types. This is not true here, since each λ-term has at least type ω. The system obtained by dropping type ω, instead, is weakly normalizing. This could be proved along the same pattern using a different definition of cut degree.

2.1. Definition (Cut-type and Rank). Given a cut in a derivation

$$\dfrac{B, x:\sigma :- M:\tau \quad B' :- N:\sigma}{B, B' :- M[N/x]:\tau}\,\text{cut}$$

we call $B, x:\sigma :- M:\tau$ the left premise, $B' :- N:\sigma$ the right premise, and σ the *cut-type*.

The *left rank* of the cut is the largest number of consecutive sequents containing $x:\sigma$ in the assumptions, along every path leading from the left premise of the cut to a leaf of the derivation tree.

The *right rank* of the cut is the largest number of consecutive sequents along every path from the right premise of the cut to a leaf of the derivation tree, such that σ is the predicate of the succedent of each of these sequents.

The *rank* of the cut is the sum of its left and right ranks.

For example, let $\rho = ((\sigma\to\sigma)\to\sigma\to\sigma) \wedge (\sigma\to\sigma)$ and let $\mathfrak{D}$ be the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{y:\sigma\to\sigma\} :- y:\sigma\to\sigma}\,Ax \quad \dfrac{}{\{z:\sigma\to\sigma\} :- z:\sigma\to\sigma}\,Ax}{\{x:(\sigma\to\sigma)\to\sigma\to\sigma, z:\sigma\to\sigma\} :- xz:\sigma\to\sigma}\,\to L}{\{x:\rho, z:\sigma\to\sigma\} :- xz:\sigma\to\sigma}\,\wedge L}{\{x:\rho, z:\rho\} :- xz:\sigma\to\sigma}\,\wedge L \quad \dfrac{}{\{x:\rho\} :- x:\rho}\,Ax}{\{x:\rho\} :- xx:\sigma\to\sigma}\,\text{cut}$$

Then in the derivation

$$\dfrac{\mathfrak{D} \quad \dfrac{\dfrac{\dfrac{}{\{t:\sigma\} :- t:\sigma}\,Ax}{:- \lambda t.t:\sigma\to\sigma}\,\to R \quad \dfrac{\dfrac{}{\{t:\sigma\to\sigma\} :- t:\sigma\to\sigma}\,Ax}{:- \lambda t.t:(\sigma\to\sigma)\to\sigma\to\sigma}\,\to R}{:- \lambda t.t:\rho}\,\wedge R}{:- (\lambda t.t)(\lambda t.t):\sigma\to\sigma}\,\text{cut}$$

the left and right ranks of the lowest cut are 3 and 1, respectively.

2.2. Definition (Ready Cut and Ready Derivations).

(i) A *cut* is *ready* if and only if it has rank 2;

(ii) A *derivation* $\mathfrak{D}$ is *ready* if and only if each cut in $\mathfrak{D}$ is ready.

If a cut is ready, either its left and right premises are respectively the conclusions of the left and right introduction rule for the principal constructor of the cut type, or one of its premises is an axiom or an instance of rule ω. Therefore, one can easily check that the following are all the possible shapes of ready cuts:

(a)

$$\dfrac{\dfrac{B, z:\mu :- P:\tau \quad B' :- R:\upsilon}{B, B', x:\upsilon\to\mu :- P[xR/z]:\tau}\,\to L \quad \dfrac{B'', y:\upsilon :- Q:\mu}{B'' :- \lambda y.Q:\upsilon\to\mu}\,\to R}{B, B', B'' :- P[xR/z][\lambda y.Q/x]:\tau}\,\text{cut}$$

(b)

$$\dfrac{\dfrac{B, x:\mu :- P:\tau}{B, x:\mu\wedge\upsilon :- P:\tau}\,\wedge L \quad \dfrac{B' :- Q:\mu \quad B'' :- Q:\upsilon}{B', B'' :- Q:\mu\wedge\upsilon}\,\wedge R}{B, B', B'' :- P[Q/x]:\tau}\,\text{cut}$$

(c)

$$\dfrac{\dfrac{B, x:\mu :- P:\tau \quad B'', x:\upsilon :- P:\tau}{B, B'', x:\mu\vee\upsilon :- P:\tau}\,\vee L \quad \dfrac{B' :- Q:\mu}{B' :- Q:\mu\vee\upsilon}\,\vee R}{B, B', B'' :- P[Q/x]:\tau}\,\text{cut}$$

(d)

$$\dfrac{\dfrac{B_i :- M:\rho}{B, x:\sigma :- M:\rho}\,\chi L \quad \dfrac{}{B', y:\sigma :- y:\sigma}\,Ax}{B, B', y:\sigma :- M[y/x]:\rho}\,\text{cut}$$

(e)

$$\dfrac{\dfrac{}{B, x:\sigma :- x:\sigma}\,Ax \quad \dfrac{B_i :- M:\sigma_i}{B' :- M:\sigma}\,\chi R}{B, B' :- M:\sigma}\,\text{cut}$$

where χ is any type constructor, and the premises of rules χL and χR are one or two according to Definition 1.3.

(f)

$$\dfrac{\dfrac{}{B, x:\omega :- x:\omega}\,\Theta \quad \dfrac{}{B' :- M:\omega}\,\omega}{B, B' :- M:\omega}\,\text{cut}$$

where Θ is either Ax or ω.

In the following we will introduce a notion of parallel reduction. In order to do this formally, we will define the notion of uniform set of redex occurrences in a term. A nonempty set of redex occurrences in a term M is called uniform if every other occurrence of the same redex is in the set as well.

As usual, any λ-term M can be identified with its syntactical tree, where nodes are represented by the elements of the subset of $\{0, 1\}^*$, corresponding to the tree domain of M. If N is a subterm of M, and its syntactical subtree is rooted at

$\alpha \in \{0, 1\}^*$ in the tree of M, then we say that "N occurs at α in M," and denote this occurrence by $\langle \alpha, N \rangle$.

2.3. Definition (Uniform Sets of Redexes). *Let M be a λ-term, then*

(i) $\mathrm{Occ}(M) = \{\langle \alpha, N \rangle \mid N \text{ occurs at } \alpha \text{ in } M\}$;

(ii) $\mathrm{Red}(M) = \{\langle \alpha, (\lambda x.P)Q \rangle \in \mathrm{Occ}(M)\}$;

(iii) $\mathfrak{F} \subseteq \mathrm{Red}(M)$ is *uniform* if and only if

$$\langle \alpha, R \rangle \in \mathfrak{F} \text{ and } \langle \beta, R \rangle \in \mathrm{Red}(M) \text{ imply } \langle \beta, R \rangle \in \mathfrak{F}.$$

In the remainder of the paper we need the notions of residual and complete development, which are well known concepts of the λ-calculus theory. Let $M \to_\beta N$ and let R be a redex occurrence in M. Informally one can say that the *set of residuals* of R in N is the (possibly empty) set of redexes which are either instances of R or copies of it generated by the reduction. This can be visualized by underlining R and preserving this underlining during the reduction. The generalization to $\twoheadrightarrow_\beta$ is straightforward. A *complete development* of $(M, \mathfrak{F})$, where $\mathfrak{F} \subseteq \mathrm{Red}(M)$, is a reduction such that all and only residuals of redexes belonging to $\mathfrak{F}$ are reduced. For a formal definition we refer the reader to (Barendregt, 1984, Chap. 11), from which we take our notation.

2.4. Definition (Parallel Reduction). The reduction relation $\Rightarrow_p$ over λ-terms is defined by

$$M \Rightarrow_p N \text{ if and only if there exists } \mathfrak{F} \subseteq \mathrm{Red}(M) \text{ such that } \mathfrak{F} \text{ is uniform and } (M, \mathfrak{F}) \twoheadrightarrow_{\mathrm{cpl}} N,$$

where $(M, \mathfrak{F}) \twoheadrightarrow_{\mathrm{cpl}} N$ is the complete development of $(M, \mathfrak{F})$ (see Barendregt, 1984, p. 286).

In the literature the idea of parallel reduction in the framework of the λ-calculus is usually formalized by the Gross–Knuth reduction as defined in (Böhm, 1966) and (Knuth, 1970) (cf. also (Barendregt, 1984, 13.2.7)).

2.5. Definition (Gross–Knuth Reduction). $M \to_{\mathrm{gk}} N$ if and only if $(M, \mathrm{Red}(M)) \twoheadrightarrow_{\mathrm{cpl}} N$.

Actually, this is a particular case of the relation $\Rightarrow_p$, since $\mathrm{Red}(M)$ is trivially uniform.

2.6. Definition (Cuts Generating $\mathfrak{F}$-Redexes). Let $\mathfrak{D}$ be a deduction whose conclusion subject is M, let $\mathfrak{F} \subseteq \mathrm{Red}(M)$ be uniform, and finally let

$$\frac{B, x : \sigma :\!- P : \tau \quad B' :\!- \lambda y.Q : \sigma}{B, B' :\!- P[\lambda y.Q/x] : \tau}\,\mathrm{cut}$$

be a cut in $\mathfrak{D}$. We say that the above cut *generates $\mathfrak{F}$-redexes of M* if and only if P has at least one subterm of the shape xR, such that all substitution instances of $(\lambda y.Q)R$ in M belong to $\mathfrak{F}$.

2.7. *Remark.* We have to consider substitution instances $(\lambda y.Q')R'$ of $(\lambda y.Q)R$ to take into account that other cuts, under the given one in the current deduction, may substitute variables inside $(\lambda y.Q)R$. For example, let us consider the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{z:\sigma\} :\!- z:\sigma}\,Ax \quad \dfrac{}{\{t:\sigma\} :\!- t:\sigma}\,Ax}{\{t:\sigma, y:\sigma\to\sigma\} :\!- yt:\sigma}\to L \quad \dfrac{\dfrac{}{\{x:\sigma\} :\!- x:\sigma}\,Ax}{:\!- \lambda x.x : \sigma\to\sigma}\to R}{\{t:\sigma\} :\!- (\lambda x.x)t:\sigma}\,\mathrm{cut} \quad \dfrac{}{\{u:\sigma\} :\!- u:\sigma}\,Ax}{\{u:\sigma\} :\!- (\lambda x.x)u:\sigma}\,\mathrm{cut}$$

If $\mathfrak{F} = \{\langle \varepsilon, (\lambda x.x)u \rangle\}$, then the upper cut generates $\mathfrak{F}$-redexes, even if the subject of its conclusion is $(\lambda x.x)t$.

2.8. Definition (Degree).

(i) The *degree* of a *type* σ is defined by

$$d(\phi) = d(\omega) = 0$$
$$d(\sigma \to \tau) = 1$$
$$d(\sigma \wedge \tau) = d(\sigma \vee \tau) = 2 + d(\sigma) + d(\tau).$$

(ii) Let $\mathfrak{D}$ be a deduction whose conclusion subject is M; take a uniform $\mathfrak{F} \subseteq \mathrm{Red}(M)$:

(a) the *degree* of a *cut* in $\mathfrak{D}$ *relative* to $\mathfrak{F}$ is the degree of its cut type if the cut generates $\mathfrak{F}$-redexes, and 0 otherwise;

(b) the *degree* $\mathfrak{d}(\mathfrak{D}, \mathfrak{F})$ of $\mathfrak{D}$ *relative* to $\mathfrak{F}$ is the highest degree of a cut in $\mathfrak{D}$ relative to $\mathfrak{F}$.

(iii) Let $\mathfrak{D} : B :\!- M : \sigma$ and $\mathfrak{D}' : B' :\!- M : \sigma$: then $\mathfrak{D}'$ is *non-increasing* with respect to $\mathfrak{D}$ if and only if $\mathfrak{d}(\mathfrak{D}', \mathfrak{F}) \leqslant \mathfrak{d}(\mathfrak{D}, \mathfrak{F})$ for all uniform $\mathfrak{F} \subseteq \mathrm{Red}(M)$.

2.9. *Remark.* The standard definition of degree is the height of the type expression in tree form. Our degree, instead, is 1 exactly when we have an arrow type. This is relevant when the cut generates a redex in $\mathfrak{F}$. Indeed, if the contraction of a redex in $\mathfrak{F}$ produces new redexes, these will be never contracted in the development of $\mathfrak{F}$. In other words, when eliminating a cut which generates an $\mathfrak{F}$-redex and whose cut type is of the form $\sigma \to \tau$, the redexes generated by the new cuts cannot be in $\mathfrak{F}$, by definition of residual. Hence their degree relative to $\mathfrak{F}$ will be 0.

We will now prove that we can transform deductions by adding assumptions, by changing the names of (free) term variables, and by eliminating the cuts of the shapes (e) and (f), while preserving readiness and without increasing the cut degrees.

We say that a deduction $\mathfrak{D}'$ is *similar* to a deduction $\mathfrak{D}$ if and only if $\mathfrak{D}'$ can be obtained from $\mathfrak{D}$ by adding (basic) typing statements to the bases and renaming term variables. This means that $\mathfrak{D}'$ and $\mathfrak{D}$ have the same deduction tree and differ only for the bases and the names of term variables.

2.10. LEMMA.

(i) *If* $\mathfrak{D} : B :-_{\in} M : \sigma$ *and* $B' \supseteq B$, *then there exists* $\mathfrak{D}' : B' :-_{\in} M : \sigma$ *such that* $\mathfrak{D}'$ *is similar to* $\mathfrak{D}$.

(ii) *If* $\mathfrak{D} : B, x : \tau :-_{\in} M : \sigma$ *and* y *is fresh, then there exists* $\mathfrak{D}' : B, y : \tau :-_{\in} M[y/x] : \sigma$ *such that* $\mathfrak{D}'$ *is similar to* $\mathfrak{D}$.

(iii) *All occurrences of ready cuts of the shapes* (e) *and* (f) *can be eliminated while preserving readiness, and obtaining a non-increasing deduction, with respect to the original deduction, of the same sequent.*

Proof.

(i) Easy, by adding the premises of $B' - B$ to each sequent (possibly renaming some bound variables of M).

(ii) Easy, since deductions are independent of the names of (free) term variables.

(iii) Let us consider a cut of the shape (e) or (f)

$$\dfrac{\dfrac{}{B, x : \sigma :- x : \sigma}\,Ax \qquad \dfrac{B_i :- M : \sigma_i}{B' :- M : \sigma}\,\chi R}{B, B' :- M : \sigma}\,\text{cut}$$

$$\dfrac{\dfrac{}{B, x : \omega :- x : \omega}\,\Theta \qquad \dfrac{}{B' :- M : \omega}\,\omega}{B, B' :- M : \omega}\,\text{cut}$$

where Θ is either Ax or ω. In both cases (i) implies that we can replace $B' :- M : \sigma$ by $B, B' :- M : \sigma$, obtaining a deduction that satisfies (iii). ∎

The following lemma (proved in the Appendix) states that every deduction can be transformed into a ready deduction of the same sequent without increasing the complexity of the proof.

2.11. LEMMA (Rank Lemma). *Let* $\mathfrak{D} : B :-_{\in} M : \sigma$ *be any derivation; then there exists* $\mathfrak{D}' : B :-_{\in} M : \sigma$, *such that* $\mathfrak{D}'$ *is ready and non-increasing with respect to* $\mathfrak{D}$.

Now we are able to prove that all cuts generating $\mathfrak{F}$-redexes for a fixed uniform set $\mathfrak{F}$ can be eliminated, and this will give the desired result.

2.12. THEOREM (Invariance of Typing under Parallel Reduction).

(i) $B :-_{\in} M : \rho$ *and* $M \Rightarrow_{\mathrm{p}} N$ *imply that* $B :-_{\in} N : \rho$.

(ii) $B \vdash_{\mathfrak{N}} M : \rho$ *and* $M \Rightarrow_{\mathrm{p}} N$ *imply that* $B \vdash_{\mathfrak{N}} N : \rho$.

Proof. (i) If $M \Rightarrow_{\mathrm{p}} N$, then for some uniform $\mathfrak{F} \subseteq \mathrm{Red}(M)$, $(M, \mathfrak{F}) \twoheadrightarrow_{\mathrm{cpl}} N$. Moreover, by Lemmas 2.10(iii) and 2.11, we can assume that the given $\mathfrak{D} : B :-_{\in} M : \rho$ does not contain cuts of shapes (e) and (f), and that it is ready. We proceed by induction on $\mathfrak{d}(\mathfrak{D}, \mathfrak{F})$.

If $\mathfrak{d}(\mathfrak{D}, \mathfrak{F}) = 0$, and $\mathfrak{D}$ is ready, then no cut generates $\mathfrak{F}$-redexes; hence $\mathfrak{F}$-redexes may occur just inside subterms of type ω. In this case $B :- N : \rho$ is derivable by introducing by rule ω the reducta of those subterms of M.

If $\mathfrak{d}(\mathfrak{D}, \mathfrak{F}) = 1$, then all cuts that generate $\mathfrak{F}$-redexes have an arrow type as cut type. Hence, by readiness of $\mathfrak{D}$, they have the shape

$$\dfrac{\dfrac{B, z : \mu :- P : \tau \qquad B' :- R : \upsilon}{B, B', x : \upsilon \to \mu :- P[xR/z] : \tau}\to L \qquad \dfrac{B'', y : \upsilon :- Q : \mu}{B'' :- \lambda y.Q : \upsilon \to \mu}\to R}{B, B', B'' :- P[xR/z][\lambda y.Q/x] : \tau}\,\text{cut}$$

Then all these cuts can be replaced top down,

$$\dfrac{B, z : \mu :- P^* : \tau \qquad \dfrac{B'', y : \upsilon :- Q^* : \mu \qquad B' :- R^* : \upsilon}{B', B'' :- Q^*[R^*/y] : \mu}\,\text{cut}}{B, B', B'' :- P^*[Q^*[R^*/y]/z] : \tau}\,\text{cut}$$

where we write P^*, Q^*, and R^* to take into account that the original terms P, Q and R may have been modified by the elimination of innermost cuts. Moreover the subject parts in the rest of $\mathfrak{D}$ have to be reduced by $\Rightarrow_{\mathrm{p}}$ in order to match. Eventually, we get a deduction of degree 0.

For $\mathfrak{d}(\mathfrak{D}, \mathfrak{F}) > 1$, let k be the number of cuts of degree $\mathfrak{d}(\mathfrak{D}, \mathfrak{F})$. Then there exists an innermost cut of this degree (i.e., in the subderivation of this cut there are only cuts of lower degree) whose cut type is either a conjunction or a disjunction. By readiness it is either of the shape

$$\dfrac{\dfrac{B, x : \mu :- P : \tau}{B, x : \mu \wedge \upsilon :- P : \tau}\wedge L \qquad \dfrac{B' :- Q : \mu \qquad B'' :- Q : \upsilon}{B', B'' :- Q : \mu \wedge \upsilon}\wedge R}{B, B', B'' :- P[Q/x] : \tau}\,\text{cut}$$

or of the shape

$$\dfrac{\dfrac{B, x : \mu :- P : \tau \qquad B'', x : \upsilon :- P : \tau}{B, B'', x : \mu \vee \upsilon :- P : \tau}\vee L \qquad \dfrac{B' :- Q : \mu}{B' :- Q : \mu \vee \upsilon}\vee R}{B, B', B'' :- P[Q/x] : \tau}\,\text{cut}$$

where we note that Q cannot be a variable since, by the fact that the cut generates redexes, it is an abstraction: consequently, the rule on the right cannot be an axiom (in fact any ready cut of shape (d) has degree 0).

In both cases, we replace it by

$$\dfrac{B, x : \mu :- P : \tau \quad B', B'' :- Q : \mu}{B, B', B'' :- P[Q/x] : \tau}\,\text{cut}$$

Since this cut has a lower degree, and we are left with $k-1$ cuts of degree $\mathfrak{d}(\mathfrak{D}, \mathfrak{F})$, we conclude that the degree of $\mathfrak{D}$ will be lowered in k steps while preserving the same subjects in all sequensts in the derivation. Once each cut of degree $\mathfrak{d}(\mathfrak{D}, \mathfrak{F})$ has been eliminated, we use the Rank Lemma to get a ready derivation, which, by what has been said above and again by the Rank Lemma, will be of lower degree.

Having a ready deduction of lower degree, the induction hypothesis applies.

(ii) Immediate from (i) and Theorem 1.5. ▮

2.13. Corollary (Invariance under Gross–Knuth Reduction).

(i) *If* $B \vdash_{\mathfrak{N}} M : \sigma$ *and* $M \rightarrow_{\mathrm{gk}} N$, *then* $B \vdash_{\mathfrak{N}} N : \sigma$;

(ii) *If* $B \vdash_{\mathfrak{N}} M : \sigma$ *and* $M \rightarrow_{\beta} N$, *then there exists* L *such that* $N \twoheadrightarrow_{\beta} L$ *and* $B \vdash_{\mathfrak{N}} L : \sigma$.

Proof. (i) Immediate from 2.12(ii).

(ii) is a consequence of (i) and of the co-finality of $\rightarrow_{\mathrm{gk}}$ with respect to $\rightarrow_{\beta}$, proved in (Barendregt, 1984, 13.2.11). ▮

3. TYPE THEORIES

In this section we consider the notion of type inclusion. This is accomplished by presenting several preorders $\leqslant$ over types and by adding the subtyping rule

$$\frac{B \vdash M : \sigma \quad \sigma \leqslant \tau}{B \vdash M : \tau}$$

to the basic type assignment system.

We will use "type theory" for any collection of inequalities between types satisfying natural closure conditions.

A *formula* is an expression of the shape $\sigma \leqslant \tau$. We say that a formula $\sigma \leqslant \tau$ is derivable in a type assignment system $\vdash^{+}$ if and only if $\{x : \sigma\} \vdash^{+} x : \tau$.

A natural requirement for a type theory, which will be included in the definition, is that it contain all formulas which are derivable in the basic system $\mathfrak{N}$.

3.1. Definition (Type Theories). *A type theory* $\mathfrak{T}$ is any set of formulas containing $\sigma \leqslant \tau$ whenever $\{x : \sigma\} \vdash_{\mathfrak{N}} x : \tau$. Let Σ be the minimal type theory.

For example, $\sigma \vee (\tau \wedge \rho) \leqslant_{\Sigma} (\sigma \vee \tau) \wedge (\sigma \vee \rho)$. In fact, we have the derivation.

$$(\vee E)\dfrac{(\wedge I)\dfrac{(\vee I)\dfrac{(Ax)\dfrac{}{\{x:\sigma,\ y:\upsilon\} \vdash x:\sigma}}{\{x:\sigma,\ y:\upsilon\} \vdash x:\sigma\vee\tau} \quad (\vee I)\dfrac{(Ax)\dfrac{}{\{x:\sigma,\ y:\upsilon\} \vdash x:\sigma}}{\{x:\sigma,\ y:\upsilon\} \vdash x:\sigma\vee\rho}}{\{x:\sigma,\ y:\upsilon\} \vdash x:(\sigma\vee\tau)\wedge(\sigma\vee\rho)} \quad (\wedge I)\dfrac{(\vee I)\dfrac{(\wedge E)\dfrac{(Ax)\dfrac{}{\{x:\tau\wedge\rho,\ y:\upsilon\} \vdash x:\tau\wedge\rho}}{\{x:\tau\wedge\rho,\ y:\upsilon\} \vdash x:\tau}}{\{x:\tau\wedge\rho,\ y:\upsilon\} \vdash x:\sigma\vee\tau} \quad (\vee I)\dfrac{(\wedge E)\dfrac{(Ax)\dfrac{}{\{x:\tau\wedge\rho,\ y:\upsilon\} \vdash x:\tau\wedge\rho}}{\{x:\tau\wedge\rho,\ y:\upsilon\} \vdash x:\rho}}{\{x:\tau\wedge\rho,\ y:\upsilon\} \vdash x:\sigma\vee\rho}}{\{x:\tau\wedge\rho,\ y:\upsilon\} \vdash x:(\sigma\vee\tau)\wedge(\sigma\vee\rho)} \quad (Ax)\dfrac{}{\{y:\upsilon\} \vdash y:\upsilon}}{\{y:\upsilon\} \vdash y:(\sigma\vee\tau)\wedge(\sigma\vee\rho)}$$

where $\upsilon \equiv \sigma \vee (\tau \wedge \rho)$. Note that the inverse inclusion, i.e., $(\sigma \vee \tau) \wedge (\sigma \vee \rho) \leqslant \sigma \vee (\tau \wedge \rho)$, does not belong to Σ.

Notation. $\sigma \leqslant_{\mathfrak{T}} \tau$ stands for $\sigma \leqslant \tau \in \mathfrak{T}$; $\sigma \sim_{\mathfrak{T}} \tau$ stands for $\sigma \leqslant_{\mathfrak{T}} \tau \leqslant_{\mathfrak{T}} \sigma$.

It is easy to verify that the following axiomatic definition of type theory is equivalent to the previous one.

3.2. Definition (Type Theories, alternative definition). A *type theory* $\mathfrak{T}$ is any set of formulas containing all instances of the axioms

(1) $\tau \leqslant \tau \wedge \tau$

(2) $\tau \vee \tau \leqslant \tau$

(3) $\sigma \wedge \tau \leqslant \sigma, \sigma \wedge \tau \leqslant \tau$

(4) $\sigma \leqslant \sigma \vee \tau, \tau \leqslant \sigma \vee \tau$

(5) $\tau \leqslant \omega$

(6) $\sigma \leqslant \sigma$

and that is closed under all instances of the rules

(7) $\sigma \leqslant \sigma', \tau \leqslant \tau' \Rightarrow \sigma \wedge \tau \leqslant \sigma' \wedge \tau'$

(8) $\sigma \leqslant \sigma', \tau \leqslant \tau' \Rightarrow \sigma \vee \tau \leqslant \sigma' \vee \tau'$

(9) $\sigma \leqslant \tau, \tau \leqslant \rho \Rightarrow \sigma \leqslant \rho$.

There are many interesting type theories extending Σ. The one presented in the next definition is suggested by the interpretation of ω as the universe, of $\rightarrow$ as the function space constructor, of $\wedge$ as the intersection, and of $\vee$ as the union operation respectively.

3.3. Definition (Type Theory Ξ). Ξ is the least type theory which includes all instances of the axioms

(10) $\sigma \wedge (\tau \vee \rho) \leqslant (\sigma \wedge \tau) \vee (\sigma \wedge \rho)$

(11) $(\sigma \rightarrow \rho) \wedge (\sigma \rightarrow \tau) \leqslant \sigma \rightarrow \rho \wedge \tau$

(12) $(\sigma \rightarrow \rho) \wedge (\tau \rightarrow \rho) \leqslant \sigma \vee \tau \rightarrow \rho$

(13) $\omega \leqslant \omega \rightarrow \omega$

and is closed under all instances of the rule

(14) $\sigma' \leqslant \sigma, \tau \leqslant \tau' \Rightarrow \sigma \rightarrow \tau \leqslant \sigma' \rightarrow \tau'$.

3.4. *Remark* (R. Meyer). The formulas of 3.2 and 3.3 not containing ω are the axiom schemes and rules of the minimal relevant logic B^{+} as defined in (Routley and Meyer, 1972).

3.5. Definition (Type Assignment System Induced by the Type Theory $\mathfrak{T}$). Let $\mathfrak{T}$ be a type theory. The axioms and rules to derive typing statements are

$$(Ax)\ \frac{}{B, x:\sigma \vdash x:\sigma}$$

$$(\omega)\ \frac{}{B \vdash M:\omega}$$

$$(\rightarrow E)\ \frac{B \vdash M:\sigma \rightarrow \tau \quad B \vdash N:\sigma}{B \vdash MN:\tau}$$

$$(\rightarrow I)\ \frac{B, x:\sigma \vdash M:\tau}{B \vdash \lambda x.M:\sigma \rightarrow \tau}$$

$$(\wedge)\ \frac{B \vdash M:\sigma \quad B \vdash M:\tau}{B \vdash M:\sigma \wedge \tau}$$

$$(\vee)\ \frac{B, x:\sigma \vdash M:\rho \quad B, x:\tau \vdash M:\rho \quad B \vdash N:\sigma \vee \tau}{B \vdash M[N/x]:\rho}$$

$$(\leqslant_{\mathfrak{T}})\ \frac{B \vdash M:\sigma \quad \sigma \leqslant_{\mathfrak{T}} \tau}{B \vdash M:\tau}$$

Notation. We write $B \vdash^{\mathfrak{T}} M:\sigma$ to express that the statement $B \vdash M:\sigma$ is derivable in the system of 3.5.

Clearly, the notions of derivability in $\vdash_{\mathfrak{N}}$ and $\vdash^{\Sigma}$ coincide.

Let Φ be the least type theory closed under clause (10) of 3.3. Then the type assignment system $\vdash^{\Xi}$ is equivalent (from the point of view of derivability) to the system obtained by adding the rule

$$(\eta)\ \frac{B \vdash \lambda x.Mx:\sigma \quad x \notin FV(M)}{B \vdash M:\sigma}$$

to the rules of the system $\vdash^{\Phi}$. This can be proved by induction on the definition of $\leqslant_{\Xi}$ in a way similar to that of Lemma 4.2 of (Barendregt *et al.*, 1983).

Note that the inference rules of 3.5 modify the basis only by erasing premises. The following admissible rules, instead, modify the basis by replacing premises:

$$(\leqslant_{\mathfrak{T}} \#)\ \frac{B, x:\sigma \vdash M:\tau \quad \rho \leqslant_{\mathfrak{T}} \sigma}{B, x:\rho \vdash M:\tau}$$

$$(\vee \#)\ \frac{B, x:\sigma \vdash M:\rho \quad B, x:\tau \vdash M:\rho}{B, x:\sigma \vee \tau \vdash M:\rho}$$

Let us prove admissibility of rule $(\leqslant_{\mathfrak{T}} \#)$. Given a derivation, each occurrence of the axiom

$$(Ax)\ \frac{}{B, x:\sigma \vdash x:\sigma}$$

can be replaced by the deduction

$$\begin{array}{l} (Ax) \\ (\leqslant_{\mathfrak{T}}) \end{array} \frac{\frac{}{B, x:\rho \vdash x:\rho} \quad \rho \leqslant_{\mathfrak{T}} \sigma}{B, x:\rho \vdash x:\sigma}$$

For rule $(\vee \#)$, simply note that this actually is rule $\vee L$ of the sequent calculus $\mathfrak{S}$. In the following proofs we freely make use of these admissible rules.

In what follows, we shall essentially study the type assignment systems induced by the type theory Π, which is an extension of Ξ. The definition of Π makes use of a predicate $\mathbf{P}:\mathbf{T} \rightarrow \{\mathbf{tt}, \mathbf{ff}\}$ which is true for a type σ if $\vee$ occurs in it only on the left-hand side of arrows. This means that σ could be written without $\vee$ occurrences (modulo the equivalence relation induced by $\leqslant_{\Pi}$). This will be clarified by the results of Section 4.

3.6. Definition (Type Theory Π).

(i) The predicate $\mathbf{P}:\mathbf{T} \rightarrow \{\mathbf{tt}, \mathbf{ff}\}$ is defined by

$$\mathbf{P}(\phi) = \mathbf{P}(\omega) = \mathbf{tt}$$

$$\mathbf{P}(\sigma \rightarrow \tau) = \mathbf{P}(\tau)$$

$$\mathbf{P}(\sigma \wedge \tau) = \mathbf{P}(\sigma)\ \&\ \mathbf{P}(\tau)$$

$$\mathbf{P}(\sigma \vee \tau) = \mathbf{ff}.$$

(ii) Π is the least type theory, extending Ξ, closed under all instances of the rule

$$(15) \quad \mathbf{P}(\sigma) \Rightarrow \sigma \rightarrow \rho \vee \tau \leqslant (\sigma \rightarrow \rho) \vee (\sigma \rightarrow \tau).$$

Examples. $\mathbf{P}(\sigma \vee \tau \rightarrow \phi) = \mathbf{P}(\phi) = \mathbf{tt}$; $\mathbf{P}(\sigma \wedge (\tau \vee \rho)) = \mathbf{P}(\sigma)\ \&\ \mathbf{P}(\tau \vee \rho) = \mathbf{ff}$.

It is worth noting that $\mathbf{P}(\sigma)$ is decidable, and therefore one can show by structural induction on types that $\leqslant_{\Pi}$ is also decidable. Decidability of $\mathbf{P}(\sigma)$ follows also from next Lemma 4.4, since its restriction to types without union is known to be decidable (Pierce, 1989).

3.7. *Remarks.*

(i) $\mathbf{P}$ is not intended as predicate over the interpretations of types, but as syntactic constraint. However, if types are interpreted as subsets of a Scott domain that can model λ-calculus, then, if $\mathbf{P}(\sigma)$ is true, the interpretation of σ has a least element; when $\mathbf{P}(\sigma)$ is false, either the interpretation of σ does not have a least element, or there exists a τ such that $\sigma \sim_{\Pi} \tau$ and $\mathbf{P}(\tau) = \mathbf{tt}$. There is an obvious relation between our $\mathbf{P}$ and the predicate C as defined in (Abramsky, 1991); i.e., $\mathbf{P}(\sigma)$ implies $C(\sigma)$ and $C(\sigma)$ implies $\exists \tau.\sigma \sim_{\Pi} \tau$ and $\mathbf{P}(\tau)$. Actually in (Abramsky, 1991) types are interepreted as compact-open subsets and condition C is "to be a coprime".

(ii) One can verify that $\mathbf{P}(\sigma)$ is true if and only if σ is a "Harrop formula" (Harrop, 1960). Hence, clause (15) can be viewed as corresponding to the "Extended Disjunction Property."

The main motivation for introducing Π is that we are seeking a type assignment system in which types are preserved under β-reduction of subjects (see Section 4).

4. SYNTACTICAL PROPERTIES OF $\vdash^{\Pi}$: CONVERSION AND APPROXIMATION THEOREMS

An interesting property of the intersection type assignment system of (Barendregt *et al.*, 1983) is the invariance of types under β-conversion and η-reduction. We will prove that this property holds for the type assignment system induced by Π as well; the proof, by means of a suitable translation of types and deductions, profits from results for the intersection type discipline.

The system with union types is not invariant under η-expansion of subjects. This is true also in the intersection type system (a trivial example is $\vdash^{\wedge} \lambda x.x : \phi \to \phi$ and $\not\vdash^{\wedge} \lambda xy.xy : \phi \to \phi$, where $\vdash^{\wedge}$ is defined in 4.1).

A further relevant property, which is strictly connected with invariance of types under β-convertibility, is established in this section: if a type σ is deducible for a λ-term M (from a given basis) then σ is deducible for an approximant of M (from the same basis). Finally, we show that the introduction of union types in the intersection type discipline preserves the characterization of solvable and normalizing λ-terms.

We will prove the conversion and the approximation theorems by defining suitable mappings from $\mathbf{T}$ to finite sets of types without union ($\mathbf{T}_{\wedge}$), and from deductions in $\vdash^{\Pi}$ to sets of deductions in the intersection type assignment system ($\vdash^{\wedge}$).

Notation. From now on, $\vdash$ stands for $\vdash^{\Pi}$, $\leqslant$ for $\leqslant_{\Pi}$, and $\sim$ for $\sim_{\Pi}$.

4.1. DEFINITION (Intersection Type Assignment System).

(i) $\mathbf{T}_{\wedge}$ is the set of types built out of type variables and the constant ω using only the type constructors $\to$ and $\wedge$.

(ii) $\leqslant^{\wedge}$ is the relation defined by restricting all clauses in 3.2 and 3.3 to types in $\mathbf{T}_{\wedge}$.

(iii) The system $\vdash^{\wedge}$ is obtained from the system $\vdash$ by allowing only types in $\mathbf{T}_{\wedge}$.

Notice that $\mathbf{T}_{\wedge}$ and $\vdash^{\wedge}$ are the set of types and the type assignment system of (Barendregt *et al.*, 1983).

4.2. THEOREM (Main Property of $\vdash^{\wedge}$). (Barendregt *et al.*, 1983). *Let $B \vdash^{\wedge} M : \sigma$ and $M =_{\beta} N$ or $M \to_{\eta} N$. Then $B \vdash^{\wedge} N : \sigma$.*

Notation. In the following $C(n, m)$ denotes the finite set of all functions $\chi : \{1, ..., n\} \to \{1, ..., m\}$. Also, $\bigwedge_{1 \leqslant i \leqslant n} \sigma_i$ and $\bigvee_{1 \leqslant i \leqslant n} \sigma_i$ stand for $\sigma_1 \wedge .. \wedge \sigma_n$ and $\sigma_1 \vee .. \vee \sigma_n$, respectively.

4.3. LEMMA (The Mapping $\mathfrak{m}$).

(i) *Let $\sigma \sim \sigma_1 \vee .. \vee \sigma_n$ and $\tau \sim \tau_1 \vee .. \vee \tau_m$. If $\mathbf{P}(\sigma_i)$ for $i = 1, ..., n$, then $\sigma \to \tau \sim \bigvee_{\chi \in C(n,m)} \bigwedge_{1 \leqslant i \leqslant n} (\sigma_i \to \tau_{\chi(i)})$.*

(ii) *Let $\sigma \in \mathbf{T}$. Then there exists a finite set $\mathfrak{m}(\sigma) = \{\sigma_1, .., \sigma_n\}$ of types in $\mathbf{T}_{\wedge}$ such that $\sigma \sim \sigma_1 \vee .. \vee \sigma_n$.*

Proof. (i) Easy from the definitions. For example, let $n = m = 2$ and $\mathbf{P}(\sigma_1)$, $\mathbf{P}(\sigma_2)$ hold; then

$$\begin{aligned}
&\sigma_1 \vee \sigma_2 \to \tau_1 \vee \tau_2 \\
&\quad \sim (\sigma_1 \to \tau_1 \vee \tau_2) \wedge (\sigma_2 \to \tau_1 \vee \tau_2) \\
&\quad \sim ((\sigma_1 \to \tau_1) \vee (\sigma_1 \to \tau_2)) \\
&\qquad \wedge ((\sigma_2 \to \tau_1) \vee (\sigma_2 \to \tau_2)) \\
&\qquad\quad \text{since } \mathbf{P}(\sigma_1) \text{ and } \mathbf{P}(\sigma_2) \\
&\quad \sim \bigvee_{\chi \in C(2,2)} \bigwedge_{1 \leqslant i \leqslant 2} (\sigma_i \to \tau_{\chi(i)}).
\end{aligned}$$

(ii) We define the set $\mathfrak{m}(\sigma)$ by induction on σ. Let $\mathfrak{m}(\tau) = \{\tau_1, .., \tau_m\}$ and $\mathfrak{m}(\rho) = \{\rho_1, .., \rho_p\}$; then

$$\begin{aligned}
&\mathfrak{m}(\phi) = \{\phi\}, \mathfrak{m}(\omega) = \{\omega\} \\
&\mathfrak{m}(\rho \to \tau) = \left\{ \bigwedge_{1 \leqslant i \leqslant p} (\rho_i \to \tau_{\chi(i)}) \mid \chi \in C(p, m) \right\} \\
&\mathfrak{m}(\rho \wedge \tau) = \{\rho_i \wedge \tau_j \mid 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant m\} \\
&\mathfrak{m}(\rho \vee \tau) = \mathfrak{m}(\rho) \cup \mathfrak{m}(\tau).
\end{aligned}$$

Now the thesis follows from elementary properties of $\leqslant$ and from (i). ∎

Note that from the above definition it follows that $\mathbf{P}(\sigma)$ implies that $\mathfrak{m}(\sigma)$ is a singleton. For example, $\mathfrak{m}(\phi_1 \vee \phi_2 \to \phi_3) = \{(\phi_1 \to \phi_3) \wedge (\phi_2 \to \phi_3)\}$.

Obviously, $\sigma' \in \mathfrak{m}(\sigma)$ implies $\sigma' \leqslant \sigma$. In the following lemma we will prove that $\mathfrak{m}$ agrees with $\leqslant$.

4.4. LEMMA (Main Property of $\mathfrak{m}$). *$\sigma \leqslant \tau$ if and only if for all $\sigma' \in \mathfrak{m}(\sigma)$ there exists $\tau' \in \mathfrak{m}(\tau)$ such that $\sigma' \leqslant^{\wedge} \tau'$.*

Proof. ($\Rightarrow$) By induction on the derivation of $\sigma \leqslant \tau$. We give only the most interesting cases. Let $\mathfrak{m}(\sigma) = \{\sigma_1, ..., \sigma_n\}$, $\mathfrak{m}(\tau) = \{\tau_1, .., \tau_m\}$, $\mathfrak{m}(\rho) = \{\rho_1, .., \rho_p\}$.

Case $(\sigma \to \rho) \wedge (\sigma \to \tau) \leqslant \sigma \to \rho \wedge \tau$. We have that

$$\begin{aligned}
&\mathfrak{m}((\sigma \to \rho) \wedge (\sigma \to \tau)) \\
&\quad = \left\{ \bigwedge_{1 \leqslant i \leqslant n} (\sigma_i \to \rho_{\chi(i)}) \wedge (\sigma_i \to \tau_{\chi'(i)}) \right. \\
&\qquad \left. \mid \chi \in C(n, p), \chi' \in C(n, m) \right\}
\end{aligned}$$

and

$$\mathfrak{m}(\sigma \to \rho \wedge \tau) = \left\{ \bigwedge_{1 \leqslant i \leqslant n} \sigma_i \to \rho_{\chi(i)} \wedge \tau_{\chi'(i)} \mid \chi \in C(n, p), \chi' \in C(n, m) \right\},$$

from which the thesis follows.

Case $\sigma \to \rho \vee \tau \leqslant (\sigma \to \rho) \vee (\sigma \to \tau)$ *with* $\mathbf{P}(\sigma) = \mathbf{tt}$. In this case $\mathfrak{m}(\sigma) = \{\sigma'\}$, for a suitable σ', $\mathfrak{m}(\sigma \to \rho \vee \tau) = \{\sigma' \to \mu \mid \mu \in \mathfrak{m}(\rho) \cup \mathfrak{m}(\tau)\}$, and $\mathfrak{m}((\sigma \to \rho) \vee (\sigma \to \tau)) = \{\sigma' \to \rho' \mid \rho' \in \mathfrak{m}(\rho)\} \cup \{\sigma' \to \tau' \mid \tau' \in \mathfrak{m}(\tau)\}$, from which the thesis follows.

Case $\mu \leqslant \sigma, \tau \leqslant \upsilon \Rightarrow \sigma \to \tau \leqslant \mu \to \upsilon$. Let $\mathfrak{m}(\mu) = \{\mu_1, .., \mu_q\}$ and $\mathfrak{m}(\upsilon) = \{\upsilon_1, .., \upsilon_r\}$. Since

$$\mathfrak{m}(\sigma \to \tau) = \left\{ \bigwedge_{1 \leqslant i \leqslant n} (\sigma_i \to \tau_{\chi(i)}) \mid \chi \in C(n, m) \right\},$$

we have $\bigwedge_{1 \leqslant i \leqslant n} (\sigma_i \to \tau_{\chi(i)}) \in \mathfrak{m}(\sigma \to \tau)$ for each χ. Then $\bigwedge_{1 \leqslant i \leqslant n} (\sigma_i \to \tau_{\chi(i)}) \leqslant^{\wedge} \sigma_i \to \tau_{\chi(i)}$, for all i $(1 \leqslant i \leqslant n)$. From the induction hypothesis we know that for all i $(1 \leqslant i \leqslant n)$, there exists $g(i)$ such that $\tau_{\chi(i)} \leqslant^{\wedge} \upsilon_{g(\chi(i))}$. Hence, for all i $(1 \leqslant i \leqslant n)$,

$$(1) \quad \sigma_i \to \tau_{\chi(i)} \leqslant^{\wedge} \sigma_i \to \upsilon_{g(\chi(i))}.$$

From the induction hypothesis we know also that for all h $(1 \leqslant h \leqslant q)$ there exists $f(h)$ such that $\mu_h \leqslant^{\wedge} \sigma_{f(h)}$. Thus, for all h $(1 \leqslant h \leqslant q)$,

$$(2) \quad \sigma_{f(h)} \to \upsilon_{g(\chi(f(h)))} \leqslant^{\wedge} \mu_h \to \upsilon_{g(\chi(f(h)))}.$$

So, from (1) and (2), we get $\sigma_{f(h)} \to \tau_{\chi(f(h))} \leqslant^{\wedge} \mu_h \to \upsilon_{g(\chi(f(h)))}$, for all h, which implies that

$$\bigwedge_{1 \leqslant i \leqslant n} (\sigma_i \to \tau_{\chi(i)}) \leqslant^{\wedge} \bigwedge_{1 \leqslant h \leqslant q} (\mu_h \to \upsilon_{g(\chi(f(h)))}),$$

where $\bigwedge_{1 \leqslant h \leqslant q} (\mu_h \to \upsilon_{g(\chi(f(h)))}) \in \mathfrak{m}(\mu \to \upsilon)$.

($\Leftarrow$). If $\mathfrak{m}(\sigma) = \{\sigma_1, ..., \sigma_n\}$ and $\mathfrak{m}(\tau) = \{\tau_1, .., \tau_m\}$, then $\sigma \sim \sigma_1 \vee .. \vee \sigma_n$ and $\tau \sim \tau_1 \vee .. \vee \tau_m$. So $\sigma \leqslant \tau$ follows easily from the fact that $\forall i$ $(1 \leqslant i \leqslant n)$ $\exists j$ $(1 \leqslant j \leqslant m)$ such that $\sigma_i \leqslant^{\wedge} \tau_j$. ∎

The previous lemma ensures that one can obtain a sound and complete subtyping algorithm for union types out of some algorithm for intersection types. A straightforward algorithm for intersection types is presented in (Pierce, 1989).

To each basis B with types in $\mathbf{T}$, $\mathfrak{m}$ naturally associates a (finite) set of bases $\mathfrak{B}(B)$ with types in $\mathbf{T}_{\wedge}$. For each basic typing statement of B that has the shape $x : \sigma$, each basis B' in $\mathfrak{B}(B)$ will contain a statement $x : \sigma'$, where σ' belongs to $\mathfrak{m}(\sigma)$.

4.5. Definition (Set of Bases $\mathfrak{B}(B)$). $\mathfrak{B}(B) = \{B' \mid FV(B) = FV(B')$ and $[x : \sigma \in B \Rightarrow \exists \sigma' \in \mathfrak{m}(\sigma) x : \sigma' \in B']\}$.

For example, let $B = \{x : \phi_0 \vee \phi_1, y : \phi_2 \vee \phi_3\}$. Then $\mathfrak{B}(B) = \{B_1, B_2, B_3, B_4\}$, where $B_1 = \{x : \phi_0, y : \phi_2\}$, $B_2 = \{x : \phi_0, y : \phi_3\}$, $B_3 = \{x : \phi_1, y : \phi_2\}$, and $B_4 = \{x : \phi_1, y : \phi_3\}$.

By rule ($\leqslant \#$), $B \vdash M : \sigma$ implies $\forall B' \in \mathfrak{B}(B)$ $B' \vdash M : \sigma$, but $\mathfrak{B}(B)$ enjoys more interesting properties.

4.6. Theorem (Main Property of $\mathfrak{B}(B)$). *$B \vdash M : \sigma$ if and only if for all $B' \in \mathfrak{B}(B)$ there exists $\sigma' \in \mathfrak{m}(\sigma)$ such that $B' \vdash^{\wedge} M : \sigma'$.*

Proof. ($\Leftarrow$) By definition of $\mathfrak{m}(\sigma)$ we have $\sigma' \leqslant \sigma$, and, by ($\leqslant$), we get $\forall B' \in \mathfrak{B}(B)$. $B' \vdash M : \sigma$. Let $B = \{x_j : \tau_j \mid 1 \leqslant j \leqslant n\}$.

Let $\mathfrak{m}(\tau_j) = \{\mu_1, ..., \mu_m\}$; by construction, there is B'' such that all the $B_i = B''$, $x_j : \mu_i$ $(1 \leqslant i \leqslant m)$ are elements of $\mathfrak{B}(B)$ (in general there are many such B''). It is clearly sufficient to prove that $B_i \vdash M : \sigma$, for $1 \leqslant i \leqslant m$, implies B'', $x_j : \tau_j \vdash M : \sigma$. This can be obtained by applying rule ($\vee \#$) m times, getting

$$B'', x_j : \mu_1 \vee .. \vee \mu_m \vdash M : \sigma$$

and, finally, rule ($\leqslant \#$), since, by definition, $\tau_j \sim \mu_1 \vee ... \vee \mu_m$.

($\Rightarrow$) By induction on the derivation of $B \vdash M : \sigma$. For rule (ω) the thesis follows trivially.

Case (Ax).

$$(Ax)\,\frac{}{B, x : \sigma \vdash x : \sigma}$$

Let $B' \in \mathfrak{B}(B, x : \sigma)$. Then $B' = B''$, $x : \sigma'$, with $\sigma' \in \mathfrak{m}(\sigma)$. By ($Ax$) we get B'', $x : \sigma' \vdash^{\wedge} x : \sigma'$.

Case ($\to I$).

$$(\to I)\,\frac{B, x : \rho \vdash N : \tau}{B \vdash \lambda x.N : \rho \to \tau}$$

By induction, $\forall B' \in \mathfrak{B}(B)$ $\forall \rho' \in \mathfrak{m}(\rho)$ $\exists \tau'_{\rho'} \in \mathfrak{m}(\tau) B'$, $x : \rho' \vdash^{\wedge} N : \tau'_{\rho'}$ ($\tau'_{\rho'}$ depends on ρ' and B').

Using ($\to I$), we get $\forall B' \in \mathfrak{B}(B)$ $\forall \rho' \in \mathfrak{m}(\rho)$ $\exists \tau'_{\rho'} \in \mathfrak{m}(\tau)$ $B' \vdash^{\wedge} \lambda x.N : \rho' \to \tau'_{\rho'}$, and then, using ($\wedge$),

$$\forall B' \in \mathfrak{B}(B)\ B' \vdash^{\wedge} \lambda x.N : \bigwedge_{\rho' \in \mathfrak{m}(\rho)} (\rho' \to \tau'_{\rho'})$$

and

$$\bigwedge_{\rho' \in \mathfrak{m}(\rho)} (\rho' \to \tau'_{\rho'}) \in \mathfrak{m}(\rho \to \tau).$$

Case $(\to E)$.

$$(\to E)\,\frac{B \vdash M : \rho \to \sigma \quad B \vdash N : \rho}{B \vdash MN : \sigma}$$

From the induction hypothesis, we have

$$(1) \quad \forall B' \in \mathfrak{B}(B)\, \exists \rho_{B'} \in \mathfrak{m}(\rho)\, B' \vdash^{\wedge} N : \rho_{B'}$$

and

$$(2) \quad \forall B' \in \mathfrak{B}(B)\, \exists \mu_{B'} \in \mathfrak{m}(\rho \to \sigma)\, B' \vdash^{\wedge} M : \mu_{B'}.$$

Let $\mathfrak{m}(\rho) = \{\rho_1, ..., \rho_m\}$ and $\mathfrak{m}(\sigma) = \{\sigma_1, ..., \sigma_n\}$; then, by definition,

$$\mu_{B'} \equiv \bigwedge_{1 \leqslant i \leqslant m} (\rho_i \to \sigma_{\chi(i)}) \text{ for some } \chi \in C(m, n).$$

Then from (2), by $(\wedge E)$, we get

$$(3) \quad \forall B' \in \mathfrak{B}(B)\, B' \vdash^{\wedge} M : \rho_{B'} \to \sigma' \text{ for some } \sigma' \in \mathfrak{m}(\sigma).$$

Hence by $(\to E)$, from (3) and (1), we get $\forall B' \in \mathfrak{B}(B)\, \exists \sigma' \in \mathfrak{m}(\sigma)\, B' \vdash^{\wedge} MN : \sigma'$.

Case $(\wedge)$.

$$(\wedge)\,\frac{B \vdash M : \tau \quad B \vdash M : \rho}{B \vdash M : \tau \wedge \rho}$$

By induction, $\forall B' \in \mathfrak{B}(B)\, \exists \tau' \in \mathfrak{m}(\tau)\, B' \vdash^{\wedge} M : \tau'$ and $\forall B' \in \mathfrak{B}(B)\, \exists \rho' \in \mathfrak{m}(\rho)\, B' \vdash^{\wedge} M : \rho'$. Therefore, using $(\wedge)$, we get $B' \vdash M : \tau' \wedge \rho'$. By definition $\tau' \wedge \rho' \in \mathfrak{m}(\tau \wedge \rho)$.

Case $(\vee)$.

$$(\vee)\,\frac{B, x : \tau \vdash M : \sigma \quad B, x : \rho \vdash M : \sigma \quad B \vdash N : \tau \vee \rho}{B \vdash M[N/x] : \sigma}$$

By induction, $\forall B' \in \mathfrak{B}(B)\, \forall \tau' \in \mathfrak{m}(\tau)\, \exists \sigma' \in \mathfrak{m}(\sigma)\, B', x : \tau' \vdash^{\wedge} M : \sigma'$ and $\forall B' \in \mathfrak{B}(B)\, \forall \rho' \in \mathfrak{m}(\rho)\, \exists \sigma'' \in \mathfrak{m}(\sigma)\, B', x : \rho' \vdash^{\wedge} M : \sigma''$. Again, from the induction hypothesis, $\forall B' \in \mathfrak{B}(B)\, \exists \mu \in \mathfrak{m}(\tau \vee \rho)\, B' \vdash^{\wedge} N : \mu$. By definition $\mu \in \mathfrak{m}(\tau) \cup \mathfrak{m}(\rho)$. For $\mu \in \mathfrak{m}(\tau)$ we obtain a proof of $B' \vdash^{\wedge} M[N/x] : \sigma'$ simply by replacing each axiom of the shape $B'', x : \mu \vdash^{\wedge} x : \mu$ by a derivation of $B'' \vdash^{\wedge} N : \mu$ in a derivation of $B', x : \mu \vdash^{\wedge} M : \sigma'$ (note that by construction $B' \subseteq B''$). The case $\mu \in \mathfrak{m}(\rho)$ can be treated in the same way.

Case $(\leqslant)$.

$$(\leqslant)\,\frac{B \vdash M : \sigma \quad \sigma \leqslant \tau}{B \vdash M : \tau}$$

From the induction hypothesis, $\forall B' \in \mathfrak{B}(B)\, \exists \sigma' \in \mathfrak{m}(\sigma)\, B' \vdash^{\wedge} M : \sigma'$. By Lemma 4.4, $\exists \tau' \in \mathfrak{m}(\tau)\, \sigma' \leqslant^{\wedge} \tau'$ and, by rule $(\leqslant^{\wedge})$, $B' \vdash^{\wedge} M : \tau'$. ∎

Thus, by Theorem 4.6, to each deduction in $\vdash$ we can associate (in a unique way) a finite set of deductions in $\vdash^{\wedge}$ for the same λ-term. The main technical points of this theorem are the elimination of rule $(\vee)$ and the substitution of rule $(\leqslant^{\wedge})$ for rule $(\leqslant)$. So it is possible to show that the most important syntactic properties of the intersection type discipline, i.e., invariance of types under β-conversion (and η-reduction) of subjects and the Approximation Theorem, hold also for the union types discipline.

4.7. Theorem (β-Conversion and η-Reduction Theorem). *Let $B \vdash M : \sigma$ and $M =_{\beta} N$ or $M \to_{\eta} N$. Then $B \vdash N : \sigma$.*

Proof.

$$\begin{aligned} B \vdash M : \sigma &\Rightarrow \forall B' \in \mathfrak{B}(B)\, \exists \sigma' \in \mathfrak{m}(\sigma)\, B' \vdash^{\wedge} M : \sigma' \\ &\qquad \text{by Theorem 4.6} \\ &\Rightarrow \forall B' \in \mathfrak{B}(B)\, \exists \sigma' \in \mathfrak{m}(\sigma)\, B' \vdash^{\wedge} N : \sigma' \\ &\qquad \text{by Theorem 4.2} \\ &\Rightarrow B \vdash N : \sigma \\ &\qquad \text{by Theorem 4.6.} \end{aligned}$$ ∎

Note that $\sigma \not\sim \omega$ implies $\sigma' \not\sim \omega$, for all $\sigma' \in \mathfrak{m}(\sigma)$, and that if ω does not occur in σ then ω does not occur in all $\sigma' \in \mathfrak{m}(\sigma)$. Therefore, by Theorem 4.6 $(\Rightarrow)$, the following characterization of solvable and normalizing terms, stated in (Barendregt *et al.*, 1983) for the intersection type discipline, is still valid.

4.8. Theorem (Characterization of Solvable and Normalizing λ-Terms).

(i) *There exist B and $\tau \not\sim \omega$ such that $B \vdash M : \tau$ if and only if M is solvable.*

(ii) *There exist B and τ without occurrences of ω such that $B \vdash M : \tau$ if and only if M has normal form.*

4.9. *Remark.* Let $\vdash^{\wedge -}$ denote the system obtained from $\vdash^{\wedge}$ by replacing rule $(\leqslant^{\wedge})$ by rule $(\wedge E)$ of Definition 1.2. Clearly, if $\Pi \supseteq \mathfrak{T}$ then

$$B \vdash^{\mathfrak{T}} M : \sigma \Rightarrow B \vdash M : \sigma$$

and

$$B' \vdash^{\wedge -} M : \sigma' \Rightarrow B' \vdash^{\mathfrak{T}} M : \sigma'$$

(where σ' and all types in B' belong to $\mathbf{T}_\wedge$). Therefore, since Theorem 4.8 holds when replacing $\vdash^{\wedge^-}$ for $\vdash$ (Coppo *et al.*, 1981), it extends to all systems $\vdash^{\mathfrak{T}}$ such that $\Pi \supseteq \mathfrak{T}$ (and therefore also to the basic system of Section 1).

To formulate the Approximation Theorem we introduce the usual notions of $\lambda-\Omega$-calculus and approximants (Wadsworth, 1976, Barendregt, 1984).

4.10. Definition ($\lambda-\Omega$-Calculus). The set of $\lambda-\Omega$-terms is obtained by adding the constant Ω to the formation rules of λ-terms. To the usual reduction rules we add the following Ω-reductions rules:

$$\Omega M \to \Omega, \quad \lambda x.\Omega \to \Omega.$$

A $\lambda-\Omega$-term A is a $\beta-\Omega$-normal form if and only if it cannot be reduced using β and Ω reduction rules.

4.11. Definition (Approximants). Let M, A be $\lambda-\Omega$-terms, A being a $\beta-\Omega$-normal form. A is an *approximant* of M ($A \sqsubseteq M$) if and only if there exists M' such that $M =_\beta M'$ and A matches M' except at occurrences of Ω in A (if any).

All the type assignment definitions above as well as those that follow in this paper can be extended to $\lambda-\Omega$-terms without modifications. This implies that terms reducible to Ω can be given only types equivalent to ω.

4.12. Theorem (Approximation Theorem for $\vdash^\wedge$) (Ronchi della Rocca and Venneri, 1984). *$B \vdash^\wedge M:\sigma$ if and only if there exists $A \sqsubseteq M$ such that $B \vdash^\wedge A:\sigma$.*

4.13. Theorem (Approximation Theorem). *$B \vdash M:\sigma$ if and only if there exists $A \sqsubseteq M$ such that $B \vdash A:\sigma$.*

Proof.

$$\begin{aligned}
B \vdash M:\sigma &\Leftrightarrow \forall B' \in \mathfrak{B}(B)\ \exists \sigma' \in \mathfrak{m}(\sigma)\\
&\qquad B' \vdash^\wedge M:\sigma' \text{ by Theorem 4.6}\\
&\Leftrightarrow \forall B' \in \mathfrak{B}(B)\ \exists \sigma' \in \mathfrak{m}(\sigma)\ \exists A_{B',\sigma'} \sqsubseteq M\\
&\qquad B' \vdash^\wedge A_{B',\sigma'}:\sigma' \text{ by Theorem 4.12}\\
&\Leftrightarrow \exists A \sqsubseteq M\ \forall B' \in \mathfrak{B}(B)\ \exists \sigma' \in \mathfrak{m}(\sigma)\\
&\qquad B' \vdash^\wedge A:\sigma' \text{ for} \Rightarrow \text{choose}\\
&\qquad A = \textstyle\bigsqcup_{B',\sigma'} A_{B',\sigma'} \text{ (where } \bigsqcup\\
&\qquad \text{is the least upper bound)}\\
&\qquad \text{and use Theorem 4.1.2 } (\Leftarrow)\\
&\Leftrightarrow \exists A \sqsubseteq M \text{ such that } B \vdash A:\sigma\\
&\qquad \text{by Theorem 4.6.} \quad \blacksquare
\end{aligned}$$

5. SEMANTICS

Usually, when type assignment systems are modelled, λ-terms are interpreted as objects of the domain D of a λ-model and types as subsets of D, in such a way that the interpretation of a λ-term is contained in the interpretation of every type which can be assigned to that λ-term.

Let us recall (cf. (Meyer, 1982, Barendregt, 1984)) that a λ-model $\mathfrak{M} = \langle D, \cdot, \mathbf{K}, \mathbf{S}, \varepsilon\rangle$ is a set D together with a binary operation $\cdot$ and elements $\mathbf{K}, \mathbf{S}, \varepsilon \in D$ such that

$$\begin{aligned}
&(\mathbf{K}\cdot d)\cdot e = d\\
&((\mathbf{S}\cdot d)\cdot e)\cdot f = (d\cdot f)\cdot(e\cdot f)\\
&(\varepsilon\cdot d)\cdot e = d\cdot e\\
&\forall e(d_1\cdot e = d_2\cdot e) \Rightarrow \varepsilon\cdot d_1 = \varepsilon\cdot d_2\\
&\varepsilon\cdot\varepsilon = \varepsilon.
\end{aligned}$$

Given a λ-model $\mathfrak{M} = \langle D, \cdot, \mathbf{K}, \mathbf{S}, \varepsilon\rangle$ and a mapping (*environment*) ζ : Term-Variables $\to D$, the interpretation $[\![M]\!]_\zeta \in D$ of a λ-term M is inductively defined by

$$\begin{aligned}
[\![x]\!]_\zeta &= \zeta(x)\\
[\![MN]\!]_\zeta &= [\![M]\!]_\zeta \cdot [\![N]\!]_\zeta\\
[\![\lambda x.M]\!]_\zeta &= \varepsilon\cdot d \text{ where } d\cdot e = [\![M]\!]_{\zeta[e/x]}\\
&\qquad \text{for all } e \in D.
\end{aligned}$$

(Recall that the existence of d in the third clause is guaranteed by the combinatorial completeness of the applicative structure $\langle D, \cdot\rangle$, which in turn depends on the properties of $\mathbf{K}$ and $\mathbf{S}$).

Types can be interpreted as subsets of D, taking into account that ω represents the universe, $\to$ represents the function space constructor, and $\wedge$ and $\vee$ represent the intersection and union operators, respectively. Therefore, given a λ-model $\mathfrak{M} = \langle D, \cdot, \mathbf{K}, \mathbf{S}, \varepsilon\rangle$, a straightforward definition of type interpretation is a mapping $\upsilon : T \to 2^D$ such that:

(i) $\upsilon(\omega) = D$

(ii) $\{\varepsilon\cdot d \mid d \in D \text{ and } \forall e \in \upsilon(\sigma) d\cdot e \in \upsilon(\tau)\} \subseteq \upsilon(\sigma\to\tau) \subseteq \{d \in D \mid \forall e \in \upsilon(\sigma)\ d\cdot e \in \upsilon(\tau)\}$

(iii) $\upsilon(\sigma\wedge\tau) = \upsilon(\sigma)\cap\upsilon(\tau)$

(iv) $\upsilon(\sigma\vee\tau) = \upsilon(\sigma)\cup\upsilon(\tau)$.

Moreover, we say that a type interpretation υ agrees with a type theory $\mathfrak{T}$ if and only if

(v) $\sigma \leqslant_{\mathfrak{T}} \tau \Rightarrow \upsilon(\sigma) \subseteq \upsilon(\tau)$.

It is easy to verify that clause (v) is always satisfied for Σ. The soundness of the type assignment system induced by any type theory $\mathfrak{T}$ with respect to this notion of type interpretation can be proved by induction on deductions.

In what follows we will introduce three other notions of type interpretation:

— the Beth type interpretations;

— the second order type interpretations;

— the type interpretations based on intersection types.

For all type theories $\mathfrak{T}$ we will prove the soundness of the type assignment systems $\vdash^{\mathfrak{T}}$ with respect to Beth and second order type interpretations. $\vdash$ will turn out to be sound with respect to the type interpretations based on intersection types; this implies the soundness of all systems $\vdash^{\mathfrak{T}}$ such that $\mathfrak{T} \subseteq \Pi$.

Moreover, we will show the completeness of $\vdash$ with respect to these three type interpretations.

Beth Type Intepretations

Our aim is to have a completeness result without requiring any saturation condition. We will proceed as follows. As first step we will define a suitable notion of Beth model relative to a given λ-model and introduce a notion of type interpretation inspired by Beth forcing (Beth, 1965).

The Beth model relative to a given λ-model will be a finitely branching tree of possible worlds. The facts known in the possible worlds will be of the shape:

"a given element of the domain (of the λ-model) belongs to a given type".

A type σ in a world w will be interpreted naturally as the set of elements of the domain which will eventually belong in the future to σ, for every possible world that will extend w. As usual, the notion of future will be formalized through that of "bar."

Note that the interpretation of types obtained in this way will be monotonic, in the sense that it increases along each branch of the tree (all this is proved in 5.7).

The above properties of type interpretation are sufficient to show the soundness of the type assignment systems induced by any type theory $\mathfrak{T}$ (Theorem 5.8).

The second step consists in proving the completeness of the type assignment system induced by Π. To achieve this we will build (for a fixed basis) a Beth model on the term model of β-equality. The possible worlds of this model will be essentially statements provable in the type assignment system: the knowledge is expanded by adding premises to the basis. The Completeness Theorem will then follow quite naturally.

5.1. Definition (Beth Model). Given a λ-model $\mathfrak{M}$ whose domain is D, a *Beth model* $\mathfrak{B}$ for $\mathfrak{M}$ is a triple

$$\mathfrak{B} = \langle \mathfrak{P}, \sqsubseteq, \{A_w\}_{w \in \mathfrak{P}} \rangle,$$

where

— $\mathfrak{P}$ is a nonempty set of elements, called *possible worls*;

— $(\mathfrak{P}, \sqsubseteq)$ is a partial ordering which can be represented as a finitely branching tree;

— for each $w \in \mathfrak{P}$, A_w is a finite set of pairs $\langle d, \phi \rangle$, where $d \in D$ and ϕ is a type variable (the *atomic facts* known at world w);

— if $w \sqsubseteq w'$ (w' extends w), then $A_w \subseteq A_{w'}$.

Read "$\langle d, \phi \rangle \in A_w$" as "at world w, ϕ is known to contain d".

Looking at the representation of $(\mathfrak{P}, \sqsubseteq)$ as a finitely branching tree, a bar $\mathfrak{b}_w$ for a world w is any collection of worlds (each extending w) such that every maximal branch passing through w intersects (i.e., contains one of the elements of) $\mathfrak{b}_w$. The one above is the standard definition of bar. For our aims it sufficies to consider the following more restrictive and simpler definition.

5.2. Definition (Bar). Looking at the representation of $(\mathfrak{P}, \sqsubseteq)$ as a finitely branching tree, take the (infinite) subtree whose root is w and cut this subtree at some given level. The so obtained set of leaves is a *bar* of w.

The following properties of bars follow easily by definition.

5.3. Proposition (Properties of Bars).

(i) *$\{w\}$ is a bar for w.*

(ii) *Let $v \sqsubseteq w$ and $\mathfrak{b}_v$ be a bar for v. Then either*

— *there is $v' \in \mathfrak{b}_v$ such that $v' \sqsubseteq w$ or*

— *the set of all worlds in $\mathfrak{b}_v$ greater than w is a bar for w.*

(iii) *If $\mathfrak{b}_w$ is a bar for w and, for every $w' \in \mathfrak{b}_w$, $\mathfrak{b}_{w'}$ is a bar for w', then $\bigcup_{w' \in \mathfrak{b}_w} \mathfrak{b}_{w'}$ is a bar for w as well.*

The notion of Beth model naturally suggests a notion of type interpretation.

5.4. Definition (Beth Type Interpretation). Every Beth model $\mathfrak{B}$ for $\mathfrak{M} = \langle D, \cdot, \mathbf{K}, \mathbf{S}, \varepsilon \rangle$ defines a mapping $\upsilon : T \to \mathfrak{P} \to 2^D$ as follows:

(i) $\upsilon(\omega, w) = D$

(ii) $\upsilon(\phi, w) = \{d \in D \mid \exists \mathfrak{b}_w \, \forall w' \in \mathfrak{b}_w \langle d, \phi \rangle \in A_{w'}\}$

(iii) $\upsilon(\sigma \to \tau, w) = \{d \in D \mid \forall w' \sqsupseteq w \; \forall e \in \upsilon(\sigma, w') \; d \cdot e \in \upsilon(\tau, w')\}$

(iv) $\upsilon(\sigma \wedge \tau, w) = \upsilon(\sigma, w) \cap \upsilon(\tau, w)$

(v) $\upsilon(\sigma \vee \tau, w) = \{d \in D \mid \exists \mathfrak{b}_w \;\; \forall w' \in \mathfrak{b}_w \;\; d \in \upsilon(\sigma, w') \cup \upsilon(\tau, w')\}$.

We say that υ is a *Beth type interpretation that agrees with* $\mathfrak{T}$ if and only if:

(vi) $\sigma \leqslant_{\mathfrak{T}} \tau$ implies $\upsilon(\sigma, w) \subseteq \upsilon(\tau, w)$ for all $w \in \mathfrak{P}$.

5.5. *Remark.* Applications of intuitionistic model theory to the semantics of λ-calculus have been proposed in the literature for several purposes. An example is the treatment of systems allowing empty interpretation of types.

Indeed the simply typed λ-calculus turns out to be complete for *Kripke λ-models* which may have empty types (Mitchel and Moggi, 1991). Even if our purposes are very different, technically speaking the present approach could seem similar. However, besides the distinction between Beth and Kripke models in logic, our construction is based on a unique model of untyped λ-calculus, where the interpretation of types increases along each branch of the tree. Mitchell and Moggi, instead, consider families of structures each assocated to a possible world, such that each world supplies a model fo simply typed λ-calculus.

The notion of semantic satisfability for our system is defined as usual.

5.6. Definition (Satisfability). Let $\mathfrak{W}, \zeta$ be as above, and υ be a type interpretation that agrees with $\mathfrak{T}$.

(i) $\mathfrak{W}, \zeta, w \models^{\mathfrak{T}} M : \sigma \Leftrightarrow [\![M]\!]_\zeta \in \upsilon(\sigma, w)$.

(ii) $\mathfrak{W}, \zeta, w \models^{\mathfrak{T}} B \Leftrightarrow \forall x : \sigma \in B\ \mathfrak{W}, \zeta, w \models^{\mathfrak{T}} x : \sigma$.

(iii) $B \models^{\mathfrak{T}} M : \sigma \Leftrightarrow \forall \mathfrak{W}, \zeta, w :$
$[\mathfrak{W}, \zeta, w \models^{\mathfrak{T}} B \Rightarrow \mathfrak{W}, \zeta, w \models^{\mathfrak{T}} M : \sigma]$.

Before giving the soundness proof of all type assignment systems we need some lemmas.

5.7. Lemma (Properties of Beth Type Interpretations).

(i) *Let $v \sqsubseteq w$. Then, for all σ, $\upsilon(\sigma, v) \subseteq \upsilon(\sigma, w)$.*

(ii) *Let $\mathfrak{b}_w$ be a bar for w. If $d \in \upsilon(\sigma, w')$ for all $w' \in \mathfrak{b}_w$, then $d \in \upsilon(\sigma, w)$.*

Proof. (i) By induction on σ.

Case $\sigma \equiv \phi$. Let $d \in \upsilon(\phi, v)$; then, by definition, $\exists \mathfrak{b}_v\ \forall v' \in \mathfrak{b}_v \langle d, \phi\rangle \in A_{v'}$. We consider two cases according to 5.3(ii). If there is a $v' \in \mathfrak{b}_v$ such that $v' \sqsubseteq w$, then, by definition, $\langle d, \phi\rangle \in A_w$, and (since $\{w\}$ is a bar for w) $d \in \upsilon(\phi, w)$. Otherwise, let $\mathfrak{b}_w$ be the set of all worlds in $\mathfrak{b}_v$ greater than w; since $\neg\exists v' \in \mathfrak{b}_v\ v' \sqsubseteq w$, and every maximal path through w also passes through v, $\mathfrak{b}_w$ is a bar for w. We have then that $\forall w' \in \mathfrak{b}_w \langle d, \phi\rangle \in A_{w'}$ (since $\mathfrak{b}_w$ is a subset of $\mathfrak{b}_v$) and thus by definition $d \in \upsilon(\phi, w)$.

Case $\sigma \equiv \rho \to \tau$.

$$\upsilon(\sigma, v) = \{d \in D \mid \forall v' \sqsupseteq v \forall e \in \upsilon(\sigma, v')\ d \cdot e \in \upsilon(\tau, v')\}$$
$$\subseteq \{d \in D \mid \forall w' \sqsupseteq w\ \forall e \in \upsilon(\sigma, w')\ d \cdot e \in \upsilon(\tau, w')\}$$
$$\text{since } w \sqsupseteq v$$
$$= \upsilon(\tau, w).$$

Case $\sigma \equiv \rho \wedge \tau$. Straightforward from the induction hypothesis.

Case $\sigma \equiv \rho \vee \tau$. Let $d \in \upsilon(\rho \vee \tau, v)$; then, by definition, $\exists \mathfrak{b}_v\ \forall v' \in \mathfrak{b}_v\ d \in \upsilon(\rho, v')$ or $d \in \upsilon(\tau, v')$. We consider two cases according to 5.3(ii). If there is a $v' \in \mathfrak{b}_v$ such that $v' \sqsubseteq w$, then (by induction) $d \in \upsilon(\rho, w)$ or $d \in \upsilon(\tau, w)$. Otherwise, let $\mathfrak{b}_w$ be the set of all worlds in $\mathfrak{b}_v$ greater than w; as above $\mathfrak{b}_w$ is a bar for w. We have then that $\forall w' \in \mathfrak{b}_w\ d \in \upsilon(\rho, w')$ or $d \in \upsilon(\tau, w')$, and then, by definition, $d \in \upsilon(\rho \vee \tau, w)$.

(ii) By induction on σ.

Case $\sigma \equiv \phi$. We have $\forall w' \in \mathfrak{b}_w\ d \in \upsilon(\phi, w')$ and by definition of υ we get

$$\forall w' \in \mathfrak{b}_w\ \exists \mathfrak{b}_{w'}\ \forall v \in \mathfrak{b}_{w'} \langle d, \phi\rangle \in A_v.$$

Let us take $\mathfrak{b}'_w = \bigcup_{w' \in \mathfrak{b}_w} \mathfrak{b}_{w'}$. This $\mathfrak{b}'_w$ is a bar for w by 5.3(iii) and, moreover, $\forall v \in \mathfrak{b}'_w \langle d, \phi\rangle \in A_v$. Then by definition $d \in \upsilon(\phi, w)$.

Case $\sigma \equiv \rho \to \tau$. By hypothesis $\forall w' \in \mathfrak{b}_w\ d \in \upsilon(\rho \to \tau, w')$. By definition this implies that

$$\forall w' \in \mathfrak{b}_w\ \forall v' \sqsupseteq w'\ [e \in \upsilon(\rho, v') \Rightarrow d \cdot e \in \upsilon(\tau, v')].$$

We want to show that $d \in \upsilon(\rho \to \tau, v')$, i.e., $\forall v \sqsupseteq v'$ $[e \in \upsilon(\rho, v) \Rightarrow d \cdot e \in \upsilon(\tau, v)]$. We consider two cases according to 5.3(ii). Given any $v \sqsupseteq w$, if there is $w' \in \mathfrak{b}_w$ such that $w' \sqsubseteq v$, then $d \in \upsilon(\rho \to \tau, v)$ by (i). Otherwise, let $\mathfrak{b}_v$ be the set of all worlds in $\mathfrak{b}_w$ greater than v; by definition, $\mathfrak{b}_v$ is a bar for v. If $e \in \upsilon(\rho, v)$, then $e \in \upsilon(\rho, v'')$ for all $v'' \in \mathfrak{b}_v$ by (i), so that $d \cdot e \in \upsilon(\tau, v'')$ by hypothesis (since $v'' \sqsupseteq v'' \in \mathfrak{b}_v$). Thus $\forall v'' \in \mathfrak{b}_v\ d \cdot e \in \upsilon(\tau, v'')$, from which $d \cdot e \in \upsilon(\tau, v)$ follows from the induction hypothesis.

Case $\sigma \equiv \rho \wedge \tau$.

$$\begin{aligned}
&\forall w' \in \mathfrak{b}_w\ d \in \upsilon(\rho \wedge \tau, w')\\
&\quad\Rightarrow \forall w' \in \mathfrak{b}_w\ d \in \upsilon(\rho, w') \text{ and } d \in \upsilon(\tau, w')\\
&\qquad\text{by definition}\\
&\quad\Rightarrow d \in \upsilon(\rho, w) \text{ and } d \in \upsilon(\tau, w)\\
&\qquad\text{by induction}\\
&\quad\Rightarrow d \in \upsilon(\rho \wedge \tau, w)\\
&\qquad\text{by definition.}
\end{aligned}$$

Case $\sigma \equiv \rho \vee \tau$. $\forall w' \in \mathfrak{b}_w\ d \in \upsilon(\rho \vee \tau, w')$ implies that $\forall w' \in \mathfrak{b}_w\ \exists \mathfrak{b}_{w'}\ \forall w'' \in \mathfrak{b}_{w'}\ d \in \upsilon(\rho, w'')$ or $d \in \upsilon(\tau, w'')$ by definition. We have that $\bigcup_{w' \in \mathfrak{b}_w} \mathfrak{b}_{w'}$ is still a bar for w and

$$\forall w'' \in \bigcup_{w' \in \mathfrak{b}_w} \mathfrak{b}_{w'}[d \in \upsilon(\rho, w'') \text{ or } d \in \upsilon(\tau, w'')];$$

i.e., by definition, $d \in \upsilon(\rho \vee \tau, w)$. ∎

5.8. THEOREM (Semantic Soundness). *Let* $\mathfrak{T}$ *be any type theory.*

(i) *If* $B, x:\tau \models^{\mathfrak{T}} M:\sigma$ *and* $B \models^{\mathfrak{T}} N:\tau$, *then* $B \models^{\mathfrak{T}} M[N/x]:\sigma$.

(ii) *If* $B \vdash^{\mathfrak{T}} M:\sigma$, *then* $B \models^{\mathfrak{T}} M:\sigma$.

Proof. (i) Choose any $\mathfrak{W}, \zeta, w$ such that $\mathfrak{W}, \zeta, w \models^{\mathfrak{T}} B$. Let $d = [\![N]\!]_\zeta$ and $\zeta' = \zeta[d/x]$. By hypothesis, $d \in \upsilon(\tau, w)$ so that $\mathfrak{W}, \zeta', w \models^{\mathfrak{T}} B, x:\tau$. Therefore, by hypothesis, we get $[\![M]\!]_{\zeta'} \in \upsilon(\sigma, w)$, which yields the thesis, since $[\![M]\!]_{\zeta'} = [\![M]\!]_{\zeta[d/x]} = [\![M[N/x]]\!]_\zeta$.

(iii) By induction on deductions. If the last applied rule is $(\leqslant_{\mathfrak{T}})$, the soundness follows from 5.4(vi). The most interesting case is when the last applied rule is

$$(\vee)\frac{B, x:\sigma \vdash^{\mathfrak{T}} M:\rho \quad B, x:\tau \vdash^{\mathfrak{T}} M:\rho \quad B \vdash^{\mathfrak{T}} N:\sigma\vee\tau}{B \vdash^{\mathfrak{T}} M[N/x]:\rho}$$

From the induction hypothesis we have that

$$\mathfrak{W}, \zeta, w \models^{\mathfrak{T}} B \text{ and } [\![x]\!]_\zeta \in \upsilon(\sigma, w) \Rightarrow [\![M]\!]_\zeta \in \upsilon(\rho, w)$$

$$\mathfrak{W}, \zeta, w \models^{\mathfrak{T}} B \text{ and } [\![x]\!]_\zeta \in \upsilon(\tau, w) \Rightarrow [\![M]\!]_\zeta \in \upsilon(\rho, w)$$

and

$$\begin{aligned}&\mathfrak{W}, \zeta, w \models^{\mathfrak{T}} B\\ &\quad\Rightarrow [\![N]\!]_\zeta \in \upsilon(\sigma\vee\tau, w)\\ &\quad\Rightarrow \exists \mathfrak{b}_w\, \forall w' \in \mathfrak{b}_w [\,[\![N]\!]_\zeta \in \upsilon(\sigma, w') \text{ or } [\![N]\!]_\zeta \in \upsilon(\tau, w')].\end{aligned}$$

Let $w' \in \mathfrak{b}_w$ and $[\![N]\!]_\zeta \in \upsilon(\sigma, w')$. By (i) $[\![M[N/x]]\!]_\zeta \in \upsilon(\rho, w')$. Similarly if $[\![N]\!]_\zeta \in \upsilon(\tau, w')$. Therefore, we conclude that

$$\mathfrak{W}, \zeta, w \models B \Rightarrow \exists \mathfrak{b}_w\, \forall w' \in \mathfrak{b}_w [\![M[N/x]]\!]_\zeta \in \upsilon(\rho, w'),$$

which implies that $\mathfrak{W}, \zeta, w \models B \Rightarrow [\![M[N/x]]\!]_\zeta \in \upsilon(\rho, w)$ by 5.7(ii). ∎

To prove completeness we use the Term Model of β-equality.

5.9. DEFINITION (The Term Model).

(i) The *Term Model* $\mathfrak{TM} = \langle T, \cdot, [\![\ ]\!]\rangle$ is the λ-calculus model defined as follows:

— $T = \{[M] \mid M \text{ is a } \lambda\text{-term}\}$, where $[M] = \{N \mid N =_\beta M\}$

— application is defined by $[M]\cdot[N] = [MN]$

— the interpretation of a λ-term M in an environment ζ is given by

$$[\![M]\!]_\zeta = [M[N_1/x_1, ..., N_n/x_n]],$$

where x_i $(1 \leqslant i \leqslant n)$ are the free variables in M and $\zeta(x_i) = [N_i]$.

(ii) In the Term Model we define the environment ζ_0 by $\zeta_0(x) = [x]$ for all x.

A straightforward induction shows that in the Term Model $[\![M]\!]_{\zeta_0} = [M]$.

To prove the completeness of our Beth type interpretation we will build a suitable Beth model for the term model relative to a given basis B_0.

First we need some technical definitions.

To each basis B we associate the (finite) set of bases $\mathfrak{B}(B)$ of 4.5. Also, let us choose an enumeration $\{\sigma_n\}_{n\in\omega}$ of all types of $\mathbf{T}$ that satisfies the following condition:

$$\forall\tau\in\mathbf{T}\ \forall m\in\mathbb{N}\ \exists n\in\mathbb{N}\ \tau \text{ occurs exactly } m \text{ times in } \{\sigma_0, ..., \sigma_n\}.$$

The possible worlds of our Beth model will represent sets of derivable statements of our type assignment system. We build a finitely branching tree $\mathfrak{Tr}$ (whose nodes are labelled with bases) starting from a given basis B_0. In $\mathfrak{Tr}$ the basis associated to a father is less powerful (from the point of view of the deducible statements) than the bases associated to each of its sons. More precisely, if B is the label of a node whose son is labelled with B', then $x:\sigma \in B$ implies $x:\sigma' \in B'$ for some $\sigma' \leqslant \sigma$ (in general the converse does not hold). Therefore, if $B \vdash M:\sigma$, then $B' \vdash M:\sigma$.

To each node of $\mathfrak{Tr}$ with label B we associate a possible world w_B representing the set of typing statements deducible from B. The possible worlds inherit the order relation from the corresponding bases; i.e., $w_B \sqsubseteq w_{B'}$ if and only if B is an ancestor of B'.

For each possible world w_B the set of atomic facts is the subset of statements of w_B whose predicates are type variables.

5.10. DEFINITION (Beth Model for the Term Model).

(i) The tree $\mathfrak{Tr}$ generated by a basis B_0 is inductively defined by:

— the root (at level 0) is labelled by B_0

— if B is the label of a node at (even) level $2n$, then it has two sons: B and $B, z:\sigma_n$ (where z is fresh and σ_n is the nth type in the fixed enumeration)

— if B is the label of a node at odd level then its sons are all the bases belonging to $\mathfrak{B}(B)$.

(ii) $w_B = \{\langle [M], \sigma\rangle \mid B \vdash M:\sigma\}$.

(iii) The partial order $(\mathfrak{TP}, \sqsubseteq)$ is defined by:

$$\mathfrak{TP} = \{w_B \mid B \text{ labels a node of } \mathfrak{Tr}\}$$

$w_B \sqsubseteq w_{B'}$ if and only if B is (the label of) an ancestor of (the node labelled) B'.

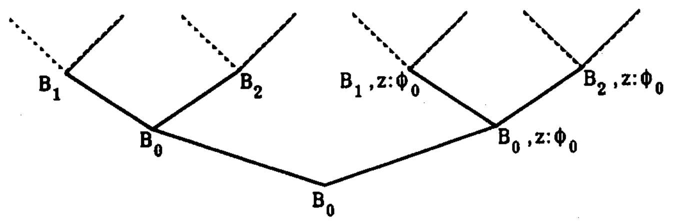

FIG. 1. The tree generated by B_0.

(iv) $A_{wB} = \{\langle [M], \phi \rangle \mid \langle [M], \phi \rangle \in w_B$ and ϕ is a type variable$\}$.

(v) The Beth model $\mathfrak{TW}$ is defined by:

$$\mathfrak{TW} = \langle \mathfrak{TP}, \sqsubseteq \{A_{wB}\}_{wB \in \mathfrak{TP}} \rangle.$$

For example, Fig. 1 shows the initial part of the tree $\mathfrak{Tr}$ generated by the basis $B_0 = \{x : \phi_0 \to \phi_1 \vee \phi_2, y : \phi_0\}$, assuming $\sigma_0 \equiv \phi_0$.

The following are some typical facts belonging to the possible worlds of the corresponding Beth model,

$$\langle [xy], \phi_1 \vee \phi_2 \rangle \in w_{B_0}$$
$$\langle [xz], \phi_1 \vee \phi_2 \rangle \in w_{B_0, z:\phi_0}$$
$$\langle [xy], \phi_1 \rangle \in w_{B_1}$$
$$\langle [xy], \phi_2 \rangle \in w_{B_2}$$
$$\langle [xz], \phi_1 \rangle \in w_{B_1, z:\phi_0}$$
$$\langle [xz], \phi_2 \rangle \in w_{B_2, z:\phi_0}$$

where $B_1 = \{x : \phi_0 \to \phi_1, y : \phi_0\}$ and $B_2 = \{x : \phi_0 \to \phi_2, y : \phi_0\}$.

5.11. LEMMA (Properties of $\mathfrak{TW}$).

(i) *If* $w_B \in \mathfrak{TP}$ *and* $B \vdash M : \sigma \vee \tau$, *then there exists* $\mathfrak{b}_{wB}$ *such that, for all* $w_{B'} \in \mathfrak{b}_{wB}$, *either* $B' \vdash M : \sigma$ *or* $B' \vdash M : \tau$.

(ii) *If* $w_B \sqsubseteq w_{B'}$, *then, for all* $x : \sigma$ *in* B, *there exists* $x : \sigma' \in B'$ *such that* $\sigma' \leqslant \sigma$.

(iii) $\upsilon_T(\sigma, w_B) = \{[M] \mid B \vdash M : \sigma\}$ *is the Beth type interpretation defined by* $\mathfrak{TW}$.

Proof. (i) Let $\mathfrak{b}_{wB}$ be the set $\{w_{B'} \mid B' \in \mathfrak{B}(B)\}$, which is a bar by definition of $\mathfrak{TW}$. By Theorem 4.6, we have that $\forall B' \in \mathfrak{B}(B)\ \exists \rho \in \mathrm{m}(\sigma) \cup \mathrm{m}(\tau)\ B' \vdash^{\wedge} M : \rho$; hence, by ($\leqslant$), we get

$$\forall B' \in \mathfrak{B}(B)\ B' \vdash M : \sigma \text{ or } B' \vdash M : \tau.$$

(ii) Immediate by construction.

(iii) Clause (vi) of Definition 5.4 is satisfied by rule ($\leqslant$).

We prove that $\upsilon_T(\sigma, w_B) = \{[M] \mid B \vdash M : \sigma\}$ satisfies the other clauses of 5.4 by induction on σ.

For ω it follows from rule (ω).

For ϕ it is true, since, by definition, $\{w_B\}$ is a bar for w_B.

In the arrow case,

$$\begin{aligned} &\upsilon_T(\sigma \to \tau, w_B) \\ &\quad = \{[M] \mid B \vdash M : \sigma \to \tau\} \\ &\quad \subseteq \{[M] \mid \forall w_{B'} \sqsupseteq w_B\ \forall N[B' \vdash N : \sigma \Rightarrow B' \vdash MN : \tau]\} \\ &\quad = \{[M] \mid \forall w_{B'} \sqsupseteq w_B\ \forall [N] \in \upsilon_T(\sigma, w_{B'})[MN] \in \upsilon_T(\tau, w_{B'})\}. \end{aligned}$$

To prove the inverse inclusion note that, if $w_{B'} \sqsupseteq w_B$, then

$$\forall N[B' \vdash N : \sigma \Rightarrow B' \vdash MN : \tau].$$

Let $\mathfrak{B}(B) = \{B_i \mid i \in I\}$ and $B'_i = B_i, z : \sigma$, where z does not occur in M. It is easy to verify that $w_{B'_i} \sqsupseteq w_B$, for all $i \in I$. Therefore:

$$\begin{aligned} &\forall N[B'_i \vdash N : \sigma \Rightarrow B'_i \vdash MN : \tau] \\ &\quad \Rightarrow B'_i \vdash^{\wedge} Mz : \tau \\ &\quad \Rightarrow B_i \vdash^{\wedge} \lambda z.Mz : \sigma \to \tau \quad \text{by } (\to I) \\ &\quad \Rightarrow B_i \vdash^{\wedge} M : \sigma \to \tau \quad \text{by Theorem 4.2.} \end{aligned}$$

Now from $\forall i \in I.\ B_i \vdash^{\wedge} M : \sigma \to \tau$ we have $B \vdash M : \sigma \to \tau$ by 4.6.

Finally, in cases of intersection and union,

$$\begin{aligned} &\upsilon_T(\sigma \wedge \tau, w_B) \\ &\quad = \{[M] \mid B \vdash M : \sigma \wedge \tau\} \\ &\quad = \{[M] \mid B \vdash M : \sigma\} \cap \{[M] \mid B \vdash M : \tau\} \\ &\quad = \upsilon_T(\sigma, w_B) \cap \upsilon_T(\tau, w_B). \end{aligned}$$

$$\begin{aligned} &\upsilon_T(\sigma \vee \tau, w_B) \\ &\quad = \{[M] \mid B \vdash M : \sigma \vee \tau\} \\ &\quad = \{[M] \mid \exists \mathfrak{b}_{wB}\ \forall w_{B'} \in \mathfrak{b}_{wB}[B' \vdash M : \sigma \text{ or } B' \vdash M : \tau]\} \\ &\qquad \text{by (i)} \\ &\quad = \{[M] \mid \exists \mathfrak{b}_{wB}\ \forall w_{B'} \in \mathfrak{b}_{wB}[M] \in \upsilon_T(\sigma, w_{B'}) \cup \upsilon_T(\tau, w_{B'})\}. \ \blacksquare \end{aligned}$$

5.12. THEOREM (Completeness). *If* $B \models M : \sigma$, *then* $B \vdash M : \sigma$.

Proof. We build $\mathfrak{Tr}$ starting from $B_0 = B$. $\mathfrak{TW}, \zeta_0, w_B$ satisfy B by definition, since $x : \sigma \in B$ implies $[x] \in \upsilon_T(\sigma, w_B)$. Then

$$\begin{aligned} B \models M : \sigma &\Rightarrow [\forall \mathfrak{W}, \zeta, w \models B \Rightarrow [\![M]\!]_{\zeta} \in \upsilon_T(\sigma, w_B)] \\ &\qquad \text{by definition} \\ &\Rightarrow [\![M]\!]_{\zeta_0} \in \upsilon_T(\sigma, w_B) \\ &\qquad \text{since } \mathfrak{TW}, \zeta_0, w_B \models B \end{aligned}$$

$\Rightarrow [M] \in \upsilon_T(\sigma, w_B)$

by definition of ζ_0

$\Rightarrow B \vdash M : \sigma$

by definition of υ_T. ∎

Second Order Type Interpretations

Another notion of type interpretation is inspired by rule ($\vee$) and by the definition of disjunction in minimal second order logic (Prawitz, 1965). This interpretation, suggested by (Berardi, 1990), avoids the introduction of possible worlds.

5.13. DEFINITION (Second Order Type Interpretation). If D is the domain of a λ-model, a mapping $\upsilon_s : \mathbf{T} \to 2^D$ is a *second order type interpretation which agrees with the type theory* $\mathfrak{T}$ if and only if:

(i) $\upsilon_s(\omega) = D$

(ii) $\upsilon_s(\sigma \to \tau) = \{d \in D \mid \forall e \in \upsilon_s(\sigma)\ d \cdot e \in \upsilon_s(\tau)\}$

(iii) $\upsilon_s(\sigma \wedge \tau) = \upsilon_s(\sigma) \cap \upsilon_s(\tau)$

(iv) $\upsilon_s(\sigma \vee \tau) = \{d \in D \mid \forall \rho\ \forall e \in \upsilon_s((\sigma \to \rho) \wedge (\tau \to \rho))\ e \cdot d \in \upsilon_s(\rho)\}$

(v) $\sigma \leqslant_{\mathfrak{T}} \tau$ implies $\upsilon_s(\sigma) \subseteq \upsilon_s(\tau)$.

This definition is not inductive in nature because of clause (iv); it must be understood as a list of conditions that any second order interpretation has to satisfy.

5.14. *Remarks.* (i) The interpretation of $\wedge$ by an operator suggested by the definition of disjunction in minimal second order logic, i.e.,

$$\upsilon_s(\sigma \wedge \tau) = \{d \in D \mid \forall \rho\ \forall e \in \upsilon_s(\sigma \to \tau \to \rho)\ e \cdot d \in \upsilon_s(\rho)\},$$

looks meaningless. As remarked by Benjamin Pierce (Pierce, 1991b), a sound second order type interpretation of $\wedge$ sould instead be

$$\upsilon_s(\sigma \wedge \tau) = \{d \in D \mid \forall \rho\ \forall e \in \upsilon_s((\sigma \to \rho) \vee (\tau \to \rho))\ e \cdot d \in \upsilon_s(\rho)\}.$$

(ii) It is always true that $\upsilon_s(\sigma) \cup \upsilon_s(\tau) \subseteq \upsilon_s(\sigma \vee \tau)$, since $d \in \upsilon_s(\sigma) \cup \upsilon_s(\tau)$ and $e \in \upsilon_s((\sigma \to \rho) \wedge (\tau \to \rho))$ imply $e \cdot d \in \upsilon_s(\rho)$. When types are interpreted as *ideals* the inverse inclusion holds as well (Coppo, 1991). This can be proved by contradiction. Indeed, if $\upsilon_s(\sigma \vee \tau) \nsubseteq \upsilon_s(\sigma) \cup \upsilon_s(\tau)$, then there exists a *finite* element d_0 such that $d_0 \in \upsilon_s(\sigma \vee \tau)$ and $d_0 \notin \upsilon_s(\sigma) \cup \upsilon_s(\tau)$. Let e_0 be a finite element such that $e_0 \notin \upsilon_s(\rho)$ for a type ρ (a possible choice is $e_0 = d_0$ and $\rho \equiv \sigma$ or $\rho \equiv \tau$), and let f_0 be the *step function*

$$f_0(d) = e_0 \text{ if } d_0 \sqsubseteq d, \perp \text{ otherwise.}$$

Then $f_0 \in \upsilon_s((\sigma \to \rho) \wedge (\tau \to \rho))$, since $\forall d \in \upsilon_s(\sigma) \cup \upsilon_s(\tau)\ f_0(d) = \perp$. On the other hand $f_0(d_0) = e_0 \notin \upsilon_s(\rho)$, which gives a contradiction.

5.15. DEFINITION (Second Order Semantic Satisfability). Let $\mathfrak{M}$, ζ be as above and let υ_s be a second order type interpretation which agrees with $\mathfrak{T}$.

(i) $\mathfrak{M}, \zeta, \upsilon_s \models_s^{\mathfrak{T}} M : \sigma \Leftrightarrow [\![M]\!]_\zeta \in \upsilon_s(\sigma)$.

(ii) $\mathfrak{M}, \zeta, \upsilon_s \models_s^{\mathfrak{T}} B \Leftrightarrow \forall x : \sigma \in B\ \mathfrak{M}, \zeta, \upsilon_s \models_s^{\mathfrak{T}} x : \sigma$.

(iii) $B \models_s^{\mathfrak{T}} M : \sigma \Leftrightarrow \forall \mathfrak{M}, \zeta, \upsilon_s : [\mathfrak{M}, \zeta, \upsilon_s \models_s^{\mathfrak{T}} B \Rightarrow \mathfrak{M}, \zeta, \upsilon_s \models_s^{\mathfrak{T}} M : \sigma]$.

5.16. THEOREM (Semantic Soundness). *If* $B \vdash^{\mathfrak{T}} M : \sigma$, *then* $B \models_s^{\mathfrak{T}} M : \sigma$.

Proof. By induction on the derivation of $B \vdash M : \sigma$. The only interesting case is when the last applied rule is

$$(\vee)\ \frac{B, x : \sigma \vdash M : \rho \quad B, x : \tau \vdash M : \rho \quad B \vdash N : \sigma \vee \tau}{B \vdash M[N/x] : \rho}$$

From the induction hypothesis we know that

(1) $\mathfrak{M}, \zeta, \upsilon_s \models_s^{\mathfrak{T}} B$ and $[\![x]\!]_\zeta \in \upsilon_s(\sigma) \Rightarrow [\![M]\!]_\zeta \in \upsilon_s(\rho)$

(2) $\mathfrak{M}, \zeta, \upsilon_s \models_s^{\mathfrak{T}} B$ and $[\![x]\!]_\zeta \in \upsilon_s(\tau) \Rightarrow [\![M]\!]_\zeta \in \upsilon_s(\rho)$

(3) $\mathfrak{M}, \zeta, \upsilon_s \models_s^{\mathfrak{T}} B \Rightarrow [\![N]\!]_\zeta \in \upsilon_s(\sigma \vee \tau)$.

From (1) and (2) $[\![\lambda x.M]\!]_\zeta \in \upsilon_s((\sigma \to \rho) \wedge (\tau \to \rho))$ follows, which implies that $[\![(\lambda x.M)N]\!]_\zeta \in \upsilon_s(\rho)$ by (3) and 5.13(iv). So we are done, since $[\![(\lambda x.M)N]\!]_\zeta \equiv [\![M[N/x]]\!]_\zeta$. ∎

For the second order type interpretation the completeness proof mimics essentially the method of (Hindley, 1982). Therefore, we use the Term Model and the environment ζ_0 as defined in 5.9.

Let us define a particular type interpretation. We relax the definition of basis to include also infinite basis; B^+ as defined below will be the only infinite basis we will deal with. As proved in (Hindley, 1983), given a λ-term M and a basis B, there is a basis B^+ such that $B \subseteq B^+$ and B^+ contains infinitely many basic statements $x_{\sigma,i} : \sigma$ for all types σ and natural numbers i, where the variables $x_{\sigma,i}$ are all distinct and occur neither in B nor in M. Consequently, $B \vdash M : \sigma$ if and only if $B^+ \vdash M : \sigma$.

5.17. DEFINITION (υ^+). Let B be a basis. $\upsilon_B^+ : \mathbf{T} \to 2^D$ is the map such that, for all $\sigma \in \mathbf{T}$,

$$\upsilon_B^+(\sigma) = \{[M] \mid B^+ \vdash M : \sigma\}.$$

In the following the parameter B will be not written explicitly since it will always be clear from the context.

To proceed in the completeness proof, we have to show that υ^+ is actually a second order type interpretation.

5.18. Lemma (Properties of υ^+).

(i) $\sigma \leqslant \tau$ *implies* $\upsilon^+(\sigma) \subseteq \upsilon^+(\tau)$;

(ii) $\upsilon^+(\omega) = \{[M] \mid M \textit{ is a } \lambda\textit{-term}\}$;

(iii) $\upsilon^+(\sigma \to \tau) = \{[M] \mid \forall [N] \in \upsilon^+(\sigma)[MN] \in \upsilon^+(\tau)\}$;

(iv) $\upsilon^+(\sigma \wedge \tau) = \upsilon^+(\sigma) \cap \upsilon^+(\tau)$;

(v) $\upsilon^+(\sigma \vee \tau) = \{d \in D \mid \forall \rho \, \forall e \in \upsilon^+((\sigma \to \rho) \wedge (\tau \to \rho))\ e \cdot d \in \upsilon^+(\rho)\}$.

Proof.

(i) $\sigma \leqslant \tau \Rightarrow [B^+ \vdash M : \sigma \Rightarrow B^+ \vdash M : \tau]$ by rule $(\leqslant)$

$$\Rightarrow \upsilon^+(\sigma) \subseteq \upsilon^+(\tau).$$

(ii) $\upsilon^+(\omega) = \{[M] \mid B^+ \vdash M : \omega\} = \{[M] \mid M$ is a λ-term$\}$.

(iii) $\upsilon^+(\sigma \to \tau)$

$$\begin{aligned} &= \{[M] \mid B^+ \vdash M : \sigma \to \tau\} \\ &\subseteq \{[M] \mid \forall N [B^+ \vdash N : \sigma \Rightarrow B^+ \vdash MN : \tau]\} \\ &= \{[M] \mid \forall [N] \in \upsilon^+(\sigma)\, [MN] \in \upsilon^+(\tau)\}. \end{aligned}$$

To prove the inverse inclusion, let z be a variable that does not occur in M and B, and such that $z : \sigma \in B^+$. Then

$$\begin{aligned} &\forall N [B^+ \vdash N : \sigma \Rightarrow B^+ \vdash MN : \tau] \\ &\quad \Rightarrow B^+ \vdash Mz : \tau \\ &\quad \Rightarrow B^+ \vdash \lambda z.Mz : \sigma \to \tau \quad \text{by } (\to I) \\ &\quad \Rightarrow B^+ \vdash M : \sigma \to \tau \quad \text{by Theorem 4.7.} \end{aligned}$$

(iv) $\upsilon^+(\sigma \wedge \tau) = \{[M] \mid B^+ \vdash M : \sigma \wedge \tau\}$

$$\begin{aligned} &= \{[M] \mid B^+ \vdash M : \sigma\} \\ &\quad \cap \{[M] \mid B^+ \vdash M : M : \tau\} \\ &= \upsilon^+(\sigma) \cap \upsilon^+(\tau). \end{aligned}$$

(v) Note first that, by (i),

$$\begin{aligned} &\upsilon^+((\sigma \to \rho) \wedge (\tau \to \rho)) \\ &\quad = \upsilon^+(\sigma \vee \tau \to \rho) \text{ since } (\sigma \to \rho) \wedge (\tau \to \rho) \sim \sigma \vee \tau \to \rho. \end{aligned}$$

$$\begin{aligned} &\upsilon^+(\sigma \vee \tau) \\ &\quad = \{[M] \mid B^+ \vdash M : \sigma \vee \tau\} \\ &\quad \subseteq \{[M] \mid \forall \rho\, \forall N [B^+ \vdash N : \\ &\qquad\qquad \sigma \vee \tau \to \rho \Rightarrow B^+ \vdash NM : \rho]\} \\ &\quad = \{[M] \mid \forall \rho\, \forall [N] \in \upsilon^+(\sigma \vee \tau \to \rho)\, [NM] \in \upsilon^+(\rho)\}. \end{aligned}$$

$$\begin{aligned} &\{[M] \mid \forall \rho\, \forall [N] \in \upsilon^+(\sigma \vee \tau \to \rho)\, [NM] \in \upsilon^+(\rho)\} \\ &\quad = \{[M] \mid \forall \rho\, \forall N (B^+ \vdash N : \\ &\qquad\qquad \sigma \vee \tau \to \rho \Rightarrow B^+ \vdash NM : \rho)\} \\ &\quad \subseteq \{[M] \mid B^+ \vdash (\lambda x.x)M : \sigma \vee \tau)\} \\ &\qquad\qquad \text{choosing } \rho \equiv \sigma \vee \tau \text{ and } N \equiv \lambda x.x \\ &\quad \subseteq \{[M] \mid B^+ \vdash M : \sigma \vee \tau\} \\ &\qquad\qquad \text{by Theorem 4.7} \\ &\quad = \upsilon^+(\sigma \vee \tau). \quad \blacksquare \end{aligned}$$

5.19. Theorem (Completeness). *If* $B \models_s M : \sigma$, *then* $B \vdash M : \sigma$.

Proof. $\mathfrak{TM}$, υ^+, ζ_0 satisfy B by definition; in fact, $x : \sigma \in B$ implies $[x] \in \upsilon^+(\sigma)$. Then

$$\begin{aligned} B \models_s M : \sigma &\Rightarrow [\forall \mathfrak{M}, \upsilon_s, \zeta \models B \Rightarrow [\![M]\!]_\zeta \in \upsilon_s(\sigma)] \\ &\qquad \text{by definition} \\ &\Rightarrow [\![M]\!]_{\zeta_0} \in \upsilon^+(\sigma) \\ &\qquad \text{since } \mathfrak{TM}, \upsilon^+, \zeta_0 \models B \\ &\Rightarrow [M] \in \upsilon^+(\sigma) \\ &\qquad \text{by definition of } \zeta_0 \text{ in 5.9(ii)} \\ &\Rightarrow B^+ \vdash M : \sigma \\ &\qquad \text{by definition of } \upsilon^+ \text{ in 5.17} \\ &\Rightarrow B \vdash M : \sigma. \quad \blacksquare \end{aligned}$$

Type Interpretations Based on Intersection Types

Using the properties of the mapping $\mathfrak{m}$, it is possible to define a type interpretation directly connected to the usual one for intersection types. Since $\mathfrak{m}$ is defined only in the case of the type theory Π, the soundness will hold only for the systems which are induced by type theories included in Π.

5.20. Definition (Type Interpretations Based on Intersection Types). If D is the domain of a λ-model, a mapping $\upsilon_\wedge : \mathbf{T}_\wedge \to 2^D$ is a *type pre-interpretation* if and only if:

(i) $\upsilon_\wedge(\omega) = D$

(ii) $\upsilon_\wedge(\sigma \to \tau) = \{d \in D \mid \forall e \in \upsilon_\wedge(\sigma)\ d \cdot e \in \upsilon_\wedge(\tau)\}$

(iii) $\upsilon_\wedge(\sigma \wedge \tau) = \upsilon_\wedge(\sigma) \cap \upsilon_\wedge(\tau)$.

Each type pre-interpretation $\upsilon_\wedge$ induces a mapping $\upsilon_\vee : \mathbf{T} \to 2^D$ (*type interpretation based on intersection types*), defined by

$$\upsilon_\vee(\sigma) = \bigcup_{\sigma' \in \mathfrak{m}(\sigma)} \upsilon_\wedge(\sigma').$$

Obviously, each type pre-interpretation is a type interpretation for $\mathbf{T}_\wedge$. Let the induced notion of semantic satisfability $\models^\vee$ be defined as usual (it sufficies to rephrase Definition 5.15).

Let $\models^\wedge$ denote semantic satisfability for intersection types. Soundness and completeness of this system (i.e., $B \models^\wedge M : \sigma \Leftrightarrow B \vdash^\wedge M : \sigma$) have been proved, using different techniques, in (Barendregt *et al.*, 1983, Hindley, 1982).

5.21. THEOREM (Soundness and Completeness). *$B \models^\vee M : \sigma$ if and only if $B \vdash M : \sigma$.*

Proof. The soundness of $\vdash$ follows easily from the soundness of $\vdash^\wedge$:

$$\begin{aligned}
& B \vdash M : \sigma \text{ and } \mathfrak{M}, \zeta, \upsilon_\vee \models^\vee B \\
& \quad \Rightarrow B \vdash M : \sigma \text{ and } \exists B' \in \mathfrak{B}(B)\ \mathfrak{M}, \zeta, \upsilon_\wedge \models^\wedge B' \\
& \qquad \text{by definition} \\
& \quad \Rightarrow \exists B' \in \mathfrak{B}(B)\ \exists \sigma' \in \mathrm{m}(\sigma)\ B' \vdash^\wedge M : \sigma' \text{ and} \\
& \qquad \mathfrak{M}, \zeta, \upsilon_\wedge \models^\wedge B' \text{ by 4.6} \\
& \quad \Rightarrow \exists \sigma' \in \mathrm{m}(\sigma)\ [\![M]\!]_\zeta \in \upsilon_\wedge(\sigma') \\
& \qquad \text{since } \vdash^\wedge \text{ is sound} \\
& \quad \Rightarrow [\![M]\!]_\zeta \in \upsilon_\vee(\sigma) \\
& \qquad \text{by definition.}
\end{aligned}$$

For the completeness proof, let $\mathfrak{M}_0$ be any model that can be used to prove the completeness of $\vdash^\wedge$ (for instance the Term Model of β-equality as defined in 5.9; see Hindley, 1982).

Let B' be an arbitrary basis belonging to $\mathfrak{B}(B)$, and let ζ_0 and $\upsilon^0_\wedge$ be respectively an environment and a type pre-interpretation (relative to $\mathfrak{M}_0$) which can be used in a completeness proof involving B' and M, i.e., such that

— $\mathfrak{M}_0, \zeta_0, \upsilon^0_\wedge \models^\wedge B'$

— $[\![M]\!]_{\zeta_0} \in \upsilon^0_\wedge(\sigma) \Rightarrow B' \vdash^\wedge M : \sigma$ for all $\sigma \in \mathbf{T}_\wedge$

(a complete definition of $\zeta_0, \upsilon^0_\wedge$ can be found in Hindley, 1982, but we do not need details here). If $\upsilon^0_\vee$ is the type interpretation induced by $\upsilon^0_\wedge$ then, by definition, $\mathfrak{M}_0, \zeta_0, \upsilon^0_\vee \models^\vee B$. Now

$$\begin{aligned}
B \models^\vee M : \sigma & \Rightarrow \forall \mathfrak{M}, \zeta, \upsilon_\vee : [\mathfrak{M}, \zeta, \upsilon_\vee \models^\vee B \\
& \qquad \Rightarrow \mathfrak{M}, \zeta, \upsilon_\vee \models^\vee M : \sigma] \quad \text{by definition} \\
& \Rightarrow [\![M]\!]_{\zeta_0} \in \upsilon^0_\vee(\sigma) \\
& \qquad \text{since } \mathfrak{M}_0, \zeta_0, \upsilon^0_\vee \models^\vee B \\
& \Rightarrow \exists \sigma' \in \mathrm{m}(\sigma)\ [\![M]\!]_{\zeta_0} \in \upsilon^0_\wedge(\sigma') \\
& \qquad \text{by 5.20} \\
& \Rightarrow \exists \sigma' \in \mathrm{m}(\sigma)\ B' \vdash^\wedge M : \sigma' \\
& \qquad \text{by hypothesis.}
\end{aligned}$$

Since B' is arbitrary, the same argument applies to each element of $\mathfrak{B}(B)$, so that

$$\begin{aligned}
B \models^\vee M : \sigma & \Rightarrow \forall B' \in \mathfrak{B}(B)\ \exists \sigma' \in \mathrm{m}(\sigma)\ B' \vdash^\wedge M : \sigma' \\
& \Rightarrow B \vdash M : \sigma \quad \text{by Theorem 4.6.} \quad \blacksquare
\end{aligned}$$

The soundness part of the above theorem easily extends to all systems $\vdash^{\mathfrak{T}}$ such that $\Pi \supseteq \mathfrak{T}$ (and therefore also to the basic system of Section 1).

6. RELATED WORK AND DIRECTIONS FOR FURTHER RESEARCH

Intersection types have been used by Reynolds (1988; 1989) to get expressive power for his programming language FORSYTHE. FORSYTHE is a typed calculus, also including records, lists, etc. However, if we restrict ourselves to pure λ-terms, we have a natural correspondence between the type assigment system of (Barendregt *et al.*, 1983) and the typed calculus of (Reynolds, 1988), as shown in (Altenkirch, 1983).

Pierce (1990; 1991a) investigated an extension of the type system of FORSYTHE in which union types are present. Pierce worked in a typed environment, but the rules of typed λ-term formation dealing with union types, developed independently, very much resemble those used in our system. He gives very interesting motivations and examples of the use of union types in actual programming.

The following is an example by Pierce, showing that some real expressive power is gained when union types are allowed. The type for the IF operator is such that the term $\mathbf{n} =$ IF b THEN 1 ELSE -1 has the type $\mathbf{n}$: Neg-num $\vee$ Pos-num. Now, if we wish to test whether $\mathbf{n}$ is zero, using the function

$$\begin{aligned}
& \text{Is-Zero} : (\text{Neg-num} \to \text{False}) \\
& \qquad \wedge (\text{Zero} \to \text{True}) \wedge (\text{Pos-num} \to \text{False})
\end{aligned}$$

we get

$$(\text{Is-Zero } \mathbf{n}) : \text{False}.$$

Without union types the best information we can obtain out of the type of (Is-Zero $\mathbf{n}$) is just that it is a boolean. Therefore we can conclude with Pierce that union types allow a restricted form of abstract interpretation to be performed during typechecking. This idea has been developed in (Jensen, 1992; Coppo and Ferrari, 1993).

A PER model validating the typing and equality rules is given in (Pierce, 1990). (Pierce, 1991a) added second order polymorphic types and showed that in his system many data types of common use can be encoded without any type or term constant.

In (Pierce, 1991a) other current researches involving union types were described.

Abramsky (Abramsky, 1991) proposed a *categorical* paradigm which generalizes Stone duality for unifying domain thery, logics of programs, and, to some extent, concurrency thery. There are technical connections between that approach and ours that are worth discussing briefly.

Let us add to our set **T** of types a minimum type. This set of types quotiented by the equality induced by the type theory Π is a domain pre-locale, taken with approximable mappings defined as in (Abramsky, 1991). The mapping $\mathfrak{m}$ introduced in Section 3 corresponds to the existence of normal forms in the sense of (Abramsky, 1991).

In (Abramsky, 1991) other type constructors (among which product, coalesced sum, lifting, a Plotkin power-domain, and a fixedpoint operator) were considered. The λ-calculus syntax is consequently enriched with corresponding term constructors. However, if we restrict ourselves to the classical syntax of terms and to types constructors build up from arrow, intersection, and union, the semantics of domains prelocales given in 4.2 of (Abramsky, 1991) is a particular case of our type interpretation based on intersection types (i.e., when the evaluation function maps intersection types on filters). Therefore, the semantic soundness and completeness results (Theorem 4.3.1 of Abramsky, 1991) could be derived immediately from our analogous results. Indeed, the completeness of types built using arrow and intersection only (Theorem 4.3.4 of Abramsky, 1991) is essentially the result of (Barendregt *et al.*, 1983).

Special cases of the kind of relationship outlined above were already described non-categorically by various authors in the literature. The origin of these ideas can be traced back to the first axiomatic presentation of domain theory given by Dana Scott, which used the notions of neighborhood systems (Scott, 1981) and information systems (Scott, 1982). The relation between information systems and intersection type disciplines was already pointed out in (Coppo *et al.*, 1984; Coppo *et al.*, 1983).

Really, the topological structures described by domain prelocales are particular cases of those described by information systems, which instead properly include those described by intersection and union types, as stated in (Alessi, 1994).

A consequence of the Approximation Theorem is that the class of λ-terms having a type different from ω does not increase by the addition of the union type constructor to intersection types. Some results of (Barbanera and Dezani–Ciancaglini, 1991) however indicate that we obtain more meaningful types. In particular, allowing also a limited form of polymorphism, "infinite unions" and "infinite intersections," we succeed in typing in a uniform way some interesting classes of λ-terms, such as the λ-terms representing numbers known as Berarducci numerals (Berarducci, 1983) and, in general, the λ-terms of *-algebras (*-algebras are λ-representations of free algebras in Böhm, 1985).

A type assignment system with infinite intersections of types was independently introduced and investigated by Leivant in (Leivant, 1990). Leivant's aim is wider. He wishes to investigate infinite intersections as a framework in which several typings, such as stratified quantificational polymorphism, can be interpreted and related. His discipline naturally stratifies into subdisciplines, allowing type formation only up to certain levels. Leivant characterizes the class of functions definable in these subdisciplines.

In a further paper (Cardone *et al.*, 1994) the present type assignment is extended by universal and existential type quantifiers and type recursion. Also in this generalization types are preserved under parallel reduction strategies. The two proofs are similar (even if the presence of other type constructors requires a more involved definition of type degree) but neither of the two immediately implies the other.

In the intersection type language we can build λ-models (filter models) in which the interpretation of a λ-term coincides with the set of all types that can be assigned to it (Barendregt *et al.*, 1983). Filter models turn out to be a very rich class containing, in particular, each inverse-limit space, and have been widely used to study properties of $D_\infty - \lambda$-models (Coppo *et al.*, 1984).

From the results of the present paper it is not clear how to derive filter models in the case of intersection and union types without introducing saturation conditions. Take for example the basis $B = \{x : \mu, y : \mu\}$, where $\mu \equiv (\sigma \wedge (\sigma \to \tau)) \vee (\rho \wedge (\rho \to \tau))$. Then $B \vdash_{\mathfrak{R}} xx : \tau$, but $B \not\vdash_{\mathfrak{R}} xy : \tau$. Since the sets of types that can be derived for x and y coincide, one would have, for a suitable filter d and all environments ζ which satisfy B, that $[\![x]\!]_\zeta = [\![y]\!]_\zeta = d$. But this leads to a ontradiction when we interpret xx and xy, since both $\tau \in d \cdot d$ and $\tau \notin d \cdot d$ should hold!

If one considers extensions of the λ-calculus by non-deterministic choice and parallel operators, a substantial use of union types results into a logical characterization of these operators. Indeed intersection and union types seem suitable to give the semantics of this enriched λ-calculus both in a classical (Dezani–Ciancaglini *et al.*, 1993), and in a lazy (Dezani–Ciancaglini *et al.*, 1994) operational perspective. The works of Boudol (1990; 1994) and Ong (1992; 1993) have been a major source of inspiration. Union types fit in the behaviour of non-determinism under a total correctness operational congruence, and the filter construction is recovered constraining the elimination rule ($\vee E$) to a suitable set of values. Both call-by-name and call-by-value abstractions are allowed; to distinguish between them disjunction is crucial. Intersection and union types dually reflect the angelic parallelism and the demonic non-determinism as stated in the operational semantics. Results of full abstraction are also proved, showing that the operational and denotational semantics completely agree.

Finally, let us mention (Verschuur, 1993), which shows an application of intersection and union types to linguistics.

APPENDIX: PROOF OF THE RANK LEMMA

RANK LEMMA. *Let* $\mathfrak{D} : B :-_{\mathfrak{F}} M : \sigma$ *be any derivation; then there exists* $\mathfrak{D}' : B :-_{\mathfrak{F}} M : \sigma$, *such that* $\mathfrak{D}'$ *is ready and non-increasing with respect to* $\mathfrak{D}$.

Proof. It suffices to show that, if $\mathfrak{D} : B :- M : \sigma$ is a derivation ending with one occurence of cut of rank $r > 2$, in which all other cuts are ready, then $\mathfrak{D}$ can be transformed into a derivation $\mathfrak{D}' : B :- M : \sigma$ containing some occurrences of the original cut with lower rank and possibly some new cuts of rank 2, while preserving the readiness of all other cuts.

The non-increasing property of $\mathfrak{D}'$ with respect to $\mathfrak{D}$ is proved by routine calculations. Note that these calculations are parametric with respect to $\mathfrak{F}$, hence it is actually an arbitrary uniform set of redex occurrences. In fact, in all the following transformations the degree of the cuts remains unchanged and the degree of the newly generated cuts is less than or equal to that of the old ones. Hence, $\mathfrak{D}'$ is non-increasing with respect to $\mathfrak{D}$.

By Lemma 2.10(iii) we can assume that $\mathfrak{D}$ does not contain ready cuts of the shape (e) or (f).

We will split the proof into two parts. In the first, we will lower the right rank of the considered cut, which is supposed to be bigger than 1; in the second, we will assume the right rank to be 1 and we will lower the left rank, which is supposed to be bigger than 1. Since all transformations leave, respectively, the left rank in part 1 and the right rank in part 2 unchanged, the rank of the cuts in the derivation will eventually be 2.

Part 1: right rank > 1. We distinguish various cases according to the shape of the last rule above the right premise of the cut. Note that Ax and ω cannot occur because of the right rank being bigger than 1. For the same motivation, cases of rules introducing type constructors to the right are impossible.

Case $\to L$.

$$\dfrac{B, x : \sigma :- M : \rho \qquad \dfrac{B', u : \tau :- N : \sigma \qquad B' :- P : \mu}{B', B'', v : \mu \to \tau :- N[vP/u] : \sigma}\to L}{B, B', B''\ v : \mu \to \tau :- M[N[vP/u]/x] : \rho}\,\mathrm{cut}$$

We can freely assume that $u \notin FV(B, B'')$ and $u \notin FV(M)$; hence we transform the above derivation into

$$\dfrac{\dfrac{B, x : \sigma :- M : \rho \qquad B', u : \tau :- N : \sigma}{B, B', u : \tau :- M[N/x] : \rho}\,\mathrm{cut} \qquad B'' :- P : \mu}{B, B', B'', v : \mu \to \tau :- M[N/x][vP/u] : \rho}\to L.$$

Note that the new cut in the figure above is legal since $u \notin FV(B, B'')$ and B, B', B'' is a basis. Moreover $u \notin FV(M)$ implies that $M[N[vP/u]/x] \equiv M[N/x][vP/u]$.

Clearly the new cut has the same left rank as the original one, while the right rank has been decreased by 1.

Cases $\wedge L$, $\vee L$. We treat only the former in detail, the latter being similar:

$$\dfrac{B, x : \sigma :- M : \rho \qquad \dfrac{B', y : \tau :- N : \sigma}{B', y, \tau \wedge \mu :- N : \sigma}\wedge L}{B, B', y : \tau \wedge \mu :- M[N/x] : \rho}\,\mathrm{cut}$$

If $y \notin FV(B)$, we transform the above derivation into

$$\dfrac{\dfrac{B, x : \sigma :- M : \rho \qquad B', y : \tau :- N : \sigma}{B, B', y : \tau :- M[N/x] : \rho}\,\mathrm{cut}}{B, B', y : \tau \wedge \mu :- M[N/x] : \rho}\wedge L$$

Otherwise, $B = B''$, $y : \tau \wedge \mu$ for some B'', since B, B', $y : \tau \wedge \mu$ is a basis. Let w be a fresh variable; define $N' \equiv N[w/y]$, so that, by 2.10(ii), there exists a derivation of B', $w : \tau :- N' : \sigma$, similar to the given subderivation of B', $y : \tau :- N : \sigma$. Hence, we transform the derivation into

$$\dfrac{\dfrac{\dfrac{B'', y : \tau \wedge \mu, x : \sigma :- M : \rho \qquad B', w : \tau :- N' : \sigma}{B', B'', y : \tau \wedge \mu, w : \tau :- M[N'/x] : \rho}\,\mathrm{cut}}{B', B'', y : \tau \wedge \mu, w : \tau \wedge \mu :- M[N'/x] : \rho}\wedge L \qquad \dfrac{}{y : \tau \wedge \mu :- y : \tau \wedge \mu}\,Ax}{B', B'', y : \tau \wedge \mu :- M[N'/x][y/w] : \rho}\,\mathrm{cut}$$

where $M[N'/x][y/w] \equiv M[N/x]$ since $w \notin FV(M)$. We note that the lower cut has rank 2 and degree 0, while the upper cut has lower rank than the original one and the same degree.

Case cut. In this case, by the hypothesis, the upper cut is ready. We must distinguish subcases according to the shapes (a)–(d) of ready cuts.

We can assume that $x \notin FV(B, B', B'')$ and $z, y, u \notin FV(B''')$.

(a)
$$\dfrac{B''', x:\sigma :- M:\rho \qquad \dfrac{\dfrac{B, z:\xi :- N:\sigma \qquad B' :- R:\upsilon}{B, B', u:\upsilon \to \xi :- N[uR/z]:\sigma}\to L \qquad \dfrac{B'', y:\upsilon :- Q:\xi}{B'' :- \lambda y.Q:\upsilon\to\xi}\to R}{B, B', B'' :- N[uR/z][\lambda y.Q/u]:\sigma}\,\text{cut}}{B, B', B'', B''' :- M[N[uR/z][\lambda y.Q/u]/x]:\rho}\,\text{cut}$$

can be replaced by

$$\dfrac{\dfrac{\dfrac{B''', x:\sigma :- M:\rho \qquad B, z:\xi :- N:\sigma}{B, B''', z:\xi :- M[N/x]:\rho}\,\text{cut} \qquad B' :- R:\upsilon}{B, B', B''', u:\upsilon\to\xi :- M[N/x][uR/z]:\rho}\to L \qquad \dfrac{B'', y:\upsilon :- Q:\xi}{B'' :- \lambda y.Q:\upsilon\to\xi}}{B, B', B'', B''' :- M[N/x][uR/z][\lambda y.Q/u]:\rho}\,\text{cut}$$

Note that $M[N[uR/z][\lambda y.Q/u]/x] \equiv M[N/x][uR/z][\lambda y.Q/u]$ since we can assume that $x \notin FV(N)$ and $u, z \notin FV(M)$.

(b and c) These cases are similar; we will treat the first one:

$$\dfrac{B''', x:\sigma :- M:\rho \qquad \dfrac{\dfrac{B, y:\mu :- N:\sigma}{B, y:\mu\wedge\upsilon :- N:\sigma}\wedge L \qquad \dfrac{B' :- P:\mu \qquad B'' :- P:\upsilon}{B', B'' :- P:\mu\wedge\upsilon}\wedge R}{B, B', B'' :- N[P/y]:\sigma}\,\text{cut}}{B, B', B'', B''' :- M[N[P/y]/x]:\rho}\,\text{cut}$$

can be replaced by:

$$\dfrac{\dfrac{\dfrac{B''', x:\sigma :- M:\rho \qquad B, y:\mu :- N:\sigma}{B, B''', y:\mu :- M[N/x]:\rho}\,\text{cut}}{B, B''', y:\mu\wedge\upsilon :- M[N/x]:\rho}\wedge L \qquad \dfrac{B' :- P:\mu \qquad B'' :- P:\upsilon}{B', B'' :- P:\mu\wedge\upsilon}\wedge R}{B, B', B'', B''' :- M[N/x][P/y]:\rho}\,\text{cut}$$

Note that $M[N[P/y]/x] \equiv M[N/x][P/y]$ since we can assume $y \notin FV(M)$.

(d) We are given a derivation of the shape

$$\dfrac{B'', x:\sigma :- M:\rho \qquad \dfrac{\dfrac{B_i :- N:\sigma}{B, y:\tau :- N:\sigma}\chi L \qquad \dfrac{}{B', z:\tau :- z:\tau}Ax}{B, B', z:\tau :- N[z/y]:\sigma}\,\text{cut}}{B, B', B'', z:\tau :- M[N[z/y]/x]:\rho}\,\text{cut}$$

where $z \notin FV(B)$ or of the shape

$$\dfrac{B'', x:\sigma :- M:\rho \qquad \dfrac{\dfrac{B_i :- N:\sigma}{B, z:\tau, y:\tau :- N:\sigma}\chi L \qquad \dfrac{}{B', z:\tau :- z:\tau}Ax}{B, B', z:\tau :- N[z/y]:\sigma}\,\text{cut}}{B, B', B'', z:\tau :- M[N[z/y]/x]:\rho}\,\text{cut}$$

where $z:\tau\in B_i$ $(i=1,2)$. In both cases we can assume that $y\notin FV(B'')$ and $y\notin FV(M)$. We transform the two derivations above into

$$\dfrac{\dfrac{B'',x:\sigma:-M:\rho \qquad B_i[z/y],B':-N[z/y]:\sigma}{B'',B_i[z/y],B':-M[N[z/y]/x]:\rho}\,\text{cut}}{B,B',B'',z:\tau:-M[N[z/y]/x]:\rho}\,\chi L$$

and into

$$\dfrac{\dfrac{\dfrac{B'',x:\sigma:-M:\rho \qquad B_i:-N':\sigma}{B'',B_i:-M[N/x]:\rho}\,\text{cut}}{B'',B,z:\tau,y:\tau:-M[N/x]:\rho}\,\chi L \qquad \dfrac{}{B',z:\tau:-z:\tau}\,Ax}{B,B',B'',z:\tau:-M[N/x][z/y]:\rho}\,\text{cut}$$

respectively.

Here and in what follows we assume there to be one or two cuts according to the value of the index i.

Note that there is a decuction of $B_i[z/y],B':-N[z/y]:\sigma$ similar to the given deduction of $B_i:-N:\sigma$ by Lemma 2.10(i) and (ii). Moreover, by the assumption $y\notin FV(M)$, $M[N[z/y]/x]\equiv M[N/x][z/y]$.

Part 2: right rank $=1$, left rank >1. In this case the deduction will have the shape

$$\dfrac{\dfrac{B_i:-N_i:\tau_i}{B:-M:\rho}\,\text{rule} \qquad B':-N:\sigma}{B,B':-M[N/x]:\rho}\,\text{cut}$$

Since the left rank is positive, the occurrence of σ in the left premise of the cut has at least one father. If $i=2$ we shall assume that the statement $x:\sigma$ occurs both in B_1 and in B_2, the case in which it occurs just once being similar but simpler.

We distinguish various cases according to the last rule above the left premise of the cut. Note that cases Ax and ω are impossible because of the left rank bigger than 1.

Case $\rightarrow L$. We are given a derivation either of the shape

$$\dfrac{\dfrac{B,x:\sigma,u:\tau:-M:\rho \qquad B',x:\sigma:-P:\xi}{B,B',v:\xi\rightarrow\tau,x:\sigma:-M[vP/u]:\rho}\rightarrow L \qquad B'':-N:\sigma}{B,B',B'',v:\xi\rightarrow\tau:-M[vP/u][N/x]:\rho}\,\text{cut}$$

or of the shape

$$\dfrac{\dfrac{B,x:\sigma\rightarrow\tau,u:\tau:-M:\rho \qquad B',x:\sigma\rightarrow\tau:-P:\sigma}{B,B',x:\sigma\rightarrow\tau:-M[xP/u]:\rho}\rightarrow L \qquad B'':-N:\sigma\rightarrow\tau}{B,B',B'':-M[xP/u][N/x]:\rho}\,\text{cut}$$

In both cases we can assume that $u\notin FV(B',B'')$ and $u\notin FV(N)$.

We transform the two derivations above into

$$\dfrac{\dfrac{B,u:\tau,x:\sigma:-M:\rho \qquad B'':-N:\sigma}{B,B'',u:\tau:-M[N/x]:\rho}\,\text{cut} \qquad \dfrac{B',x:\sigma:-P:\xi \qquad B'':-N:\sigma}{B',B'':-P[N/x]:\xi}\,\text{cut}}{B,B',B'',v:\xi\rightarrow\tau:-M[N/x][vP[N/x]/u]:\rho}\rightarrow L$$

and into

$$\dfrac{\dfrac{\dfrac{B,u:\tau,x:\sigma\rightarrow\tau:-M:\rho \qquad B'':-N:\sigma\rightarrow\tau}{B,B'',u:\tau:-M[N/x]:\rho}\,\text{cut} \qquad \dfrac{B',x:\sigma\rightarrow\tau:-P:\sigma \qquad B'':-N:\sigma\rightarrow\tau}{B',B'':-P[N/x]:\sigma}\,\text{cut}}{B,B',B'',z:\sigma\rightarrow\tau:-M[N/x][zP[N/x]/u]:\rho}\rightarrow L \qquad B'':-N:\sigma\rightarrow\tau}{B,B',B'':-M[N/x][zP[N/x]/u][N/z]:\rho}\,\text{cut}$$

respectively, where z is a fresh variable.

Note that, since $u \notin FV(N)$ and z is fresh, $M[N/x][vP[N/x]/u] \equiv M[vP/u][N/x]$ and $M[N/x][zP[N/x]/u][N/z] \equiv M[xP/u][N/x]$.

Case $\to R$.

$$\dfrac{\dfrac{B, x:\sigma, y:\tau :- M:\rho}{B, x:\sigma :- \lambda y.M : \tau \to \rho}\to R \qquad B' :- N:\sigma}{B, B' :- (\lambda y.M)[N/x] : \tau \to \rho}\text{cut}$$

Clearly, we can assume that $y \notin FV(N)$ and $y \notin FV(B')$. The derivation is transformed into

$$\dfrac{\dfrac{B, y:\tau, x:\sigma :- M:\rho \qquad B' :- N:\sigma}{B, B', y:\tau :- M[N/x] : \rho}\text{cut}}{B, B' :- \lambda y.M[N/x] : \tau \to \rho}\to R$$

and, by the fact that $y \notin FV(N)$, $(\lambda y.M)[N/x] \equiv \lambda y.M[N/x]$.

Cases $\wedge L$, $\vee L$. Again we treat just the first case, the second one being similar:

$$\dfrac{\dfrac{B, y:\tau, x:\sigma :- M:\rho}{B, y:\mu \wedge \tau, x:\sigma :- M:\rho}\wedge L \qquad B' :- N:\sigma}{B, B', y:\mu \wedge \tau :- M[N/x] : \rho}\text{cut}$$

If $y:\mu \wedge \tau \notin B'$, this derivation is transformed into

$$\dfrac{\dfrac{B, y:\tau, x:\sigma :- M:\rho \qquad B' :- N:\sigma}{B, B', y:\tau :- M[N/x] : \rho}\text{cut}}{B, B', y:\mu \wedge \tau :- M[N/x] : \rho}\wedge L$$

Otherwise, we use the same renaming technique of the cases $\wedge L$ (and $\vee L$) in the first part of this proof.

Cases $\wedge R$, $\vee R$.

$$\dfrac{\dfrac{B, x:\sigma :- M:\rho \qquad B', x:\sigma :- M:\tau}{B, B', x:\sigma :- M : \rho \wedge \tau}\wedge R \qquad B'' :- N:\sigma}{B, B', B'' :- M[N/x] : \rho \wedge \tau}\text{cut}$$

is transformed into

$$\dfrac{\dfrac{B, x:\sigma :- M:\rho \qquad B'' :- N:\sigma}{B, B'' :- M[N/x] : \rho}\text{cut} \qquad \dfrac{B', x:\sigma :- M:\tau \qquad B'' :- N:\sigma}{B', B'' :- M[N/x] : \tau}\text{cut}}{B, B', B'' :- M[N/x] : \rho \wedge \tau}\wedge R.$$

Case $\vee R$ is treated similarly.

Case cut. In this case by hypothesis the upper cut is ready. We must distinguish subcases according to the shapes (a)–(d) of ready cuts. There is no loss of generality in assuming that $y \notin FV(N)$ and $y \notin FV(B''')$.

(a)
$$\dfrac{\dfrac{\dfrac{B, x:\sigma, z:\xi :- M:\rho \qquad B', x:\sigma :- R:\upsilon}{B, B', x:\sigma, y:\upsilon \to \xi :- M[yR/z] : \rho}\to L \qquad \dfrac{B'', x:\sigma, u:\upsilon :- Q:\xi}{B'', x:\sigma :- \lambda u.Q : \upsilon \to \xi}\to R}{B, B', B'', x:\sigma :- M[yR/z][\lambda u.Q/y] : \rho}\text{cut} \qquad B''' :- N:\sigma}{B, B', B'', B''' :- M[yR/z][\lambda u.Q/y][N/x] : \rho}\text{cut}$$

is replaced by:

$$\dfrac{\dfrac{\dfrac{B, x:\sigma, z:\xi :- M:\rho \qquad B''' :- N:\sigma}{B, B''', z:\xi :- M[N/x]:\rho}\,\text{cut} \qquad \dfrac{B', x:\sigma :- R:\upsilon \qquad B''' :- N:\sigma}{B', B''' :- R[N/x]:\upsilon}\,\text{cut}}{B, B', B'', y:\upsilon\to\xi :- M[N/x][yR[N/x]/z]:\rho}\,{\to}L \qquad \dfrac{\dfrac{B'', u:\upsilon, x:\sigma :- Q:\xi \qquad B''' :- N:\sigma}{B'', B''', u:\upsilon :- Q[N/x]:\xi}\,\text{cut}}{B'', B''' :- \lambda u.Q[N/x]:\upsilon\to\xi}\,{\to}R}{B, B', B'', B''' :- M[N/x][yR[N/x]/z][\lambda u.Q[N/x]/y]:\rho}\,\text{cut}$$

Note that $M[yR/z][\lambda u.Q/y][N/x] \equiv M[N/x][yR[N/x]/z][\lambda u.Q[N/x]/y]$.

(b and c) In case (b) we have

$$\dfrac{\dfrac{\dfrac{B, x:\sigma, y:\mu :- M:\rho}{B, x:\sigma, y:\mu\wedge\upsilon :- M:\rho}\,\wedge L \qquad \dfrac{B', x:\sigma :- P:\mu \qquad B'', x:\sigma :- P:\upsilon}{B', B'', x:\sigma :- P:\mu\wedge\upsilon}\,\wedge R}{B, B', B'', x:\sigma :- M[P/y]:\rho}\,\text{cut} \qquad B''' :- N:\sigma}{B, B', B'', B''' :- M[P/y][N/x]:\rho}\,\text{cut}$$

which is replaced by

$$\dfrac{\dfrac{\dfrac{B, x:\sigma, y:\mu :- M:\rho \qquad B''' :- N:\sigma}{B, B''', y:\mu :- M[N/x]:\rho}\,\text{cut}}{B, B''', y:\mu\wedge\upsilon :- M[N/x]:\rho}\,\wedge L \qquad \dfrac{\dfrac{B', x:\sigma :- P:\mu \qquad B''' :- N:\sigma}{B', B''' :- P[N/x] :- \mu}\,\text{cut} \qquad \dfrac{B'', x:\sigma :- P:\upsilon \qquad B''' :- N:\sigma}{B'', B''' :- P[N/x]:\upsilon}\,\text{cut}}{B', B'', B''' :- P[N/x]:\mu\wedge\upsilon}\,\wedge R}{B, B', B'', B''' :- M[N/x][P[N/x]/y]:\rho}\,\text{cut}$$

Note that $M[P/y][N/x] \equiv M[N/x][P[N/x]/y]$. Case (c) is similar.

Finally in case (d) we are given a derivation either of the shape

$$\dfrac{\dfrac{\dfrac{B_i, x:\sigma :- M:\rho}{B, y:\tau, x:\sigma :- M:\rho}\,\chi L \qquad \dfrac{}{B', z:\tau :- z:\tau}\,Ax}{B, B', z:\tau, x:\sigma :- M[z/y]:\rho}\,\text{cut} \qquad B'' :- N:\sigma}{B, B', B'', z:\tau :- M[z/y][N/x]:\rho}\,\text{cut}$$

or of the shape

$$\dfrac{\dfrac{\dfrac{B_i, x:\sigma :- M:\rho}{B, y:\sigma, x:\sigma :- M:\rho}\,\chi L \qquad \dfrac{}{B', x:\sigma :- x:\sigma}\,Ax}{B, B', x:\sigma :- M[x/y]:\rho}\,\text{cut} \qquad B'' :- N:\sigma}{B, B', B'' :- M[x/y][N/x]:\rho}\,\text{cut}$$

We transform the two derivations above into

$$\dfrac{\dfrac{B_i[z/y], B', x:\sigma :- M[z/y]:\rho \qquad B'' :- N:\sigma}{B_i[z/y], B', B'' :- M[z/y][N/x]:\rho}\,\text{cut}}{B, B', B'', z:\tau :- M[z/y][N/x]:\rho}\,\chi L$$

and into

$$\dfrac{\dfrac{\dfrac{B_i, B', x:\sigma :- M:\rho \qquad B'' :- N:\sigma}{B_i, B', B'' :- M[N/x]:\rho}\,\text{cut}}{B, B', B'', y:\sigma :- M[N/x]:\rho}\,\chi L \qquad B'' :- N:\sigma}{B, B', B'' :- M[N/x][N/y]:\rho}\,\text{cut}$$

respectively. Note that in the second case the newly generated cut is ready, since by hypothesis the right rank of the current cut was 1. ∎

ACKNOWLEDGMENTS

We are very grateful to Steffen van Bakel, Stefano Berardi, Felice Cardone, Corrado Böhm, Mario Coppo, Roger Hindley, Furio Honsell, Robert Meyer, Benjamin Pierce, Simona Ronchi della Rocca and Betti Venneri for helpful discussions about the subject of this paper. Moreover, we thank the referees for pointing out some inaccuracies on a previous version of this paper and for suggesting improvements to the text. The first author thanks Paola Alfieri for her support.

Received December 1, 1992; final manuscript received February 9, 1994

REFERENCES

Abramsky, S. (1991), Domain theory in logical form, *Ann. Pure Appl. Logic* **51**, 1–77.

Alessi, F. (1994), Elementary type structures, *in* "CAAP '94," pp. 37–51, Lecture Notes in Computer Science, Vol. 787, Springer-Verlag, Berlin.

Altenkirch T. (1983), "Ein konstruktiver Ansatz zur konjunktiven Typisierung des λ-Kalkulus," Diplomarbeit, Technische Universität Berlin.

Barbanera, F., and Dezani–Ciancaglini, M. (1991), Intersection and union types, *in* "TACS '91," pp. 651–674, Lecture Notes in Computer Science, Vol. 526, Springer-Verlag, Berlin.

Barendregt, H. P. (1984), "The Lambda Calculus: Its Syntax and Semantics," Studies in Logic and the Foundations of Mathematics, Vol. 103, North-Holland, Amsterdam.

Barendregt, H. P., Coppo, M., and Dezani-Ciancaglini, M. (1983), A filter lambda model and the completeness of type assignment, *J. Symbolic Logic* **48**, 931–940.

Berardi, S. (1990), personal communication.

Berarducci, A. (1983), "Programmazione funzionale e rappresentabilità in alcuni sistemi di logica combinatoria," Tesi di Laurea, Università di Roma.

Beth, E. W. (1965), "The Foundation of Mathematics," 2nd ed., North-Holland, Amsterdam.

Böhm, C. (1966), The CUCH as a formal and description language, *in* "Formal Language Description for Computer Programming," pp. 179–197, North-Holland, Amsterdam.

Böhm, C. (1985), Functional programming and combinatory algebras, *in* "MFCS '88," pp. 14–26, Lecture Notes in Computer Science, Vol. 324, Springer-Verlag, Berlin.

Boudol, G. (1990), "A Lambda-Calculus for Parallel Functions," Rapport de Recherche INRIA 1231.

Boudol, G. (1994), A lambda-calculus for (strict) parallel functions, *Inform. and Comput.* **108**, 51–127.

Cardone, F., Dezani–Ciancaglini, M., and de'Liguoro, U. (1994), Com bining type disciplines, *Ann. Pure Appl. Logic* **66**, 197–230.

Coppo, M. (1991), personal communication.

Coppo, M., Dezani–Ciancaglini, M., Honsell, F., and Longo, G. (1984), Extended type structures and filter lambda models, *in* "Logic Colloquium 82," pp. 241–262, North-Holland, Amsterdam.

Coppo, M., Dezani–Ciancaglini, M., and Longo, G. (1983), Applicative information systems, *in* "CAAP '83," pp. 35–64, Lecture Notes in Computer Science, Vol. 159, Springer-Verlag, Berlin.

Coppo, M., Dezani–Ciancaglini, M., and Venneri, B. (1981), Functional characters of solvable terms, *Z. Math. Logik* **27**, 45–58.

Coppo, M., and Ferrari, A. (1993), Type inference, abstract interpretation and strictness analysis, *Theoret. Comput. Sci.* **121**, 113–144.

Curry, H., and Feys, R. (1958), "Combinatory Logic," North-Holland, Amsterdam.

Dezani–Ciancaglini, M., Ghilezan, S., and Venneri, B. (1992), "The 'Relevance' of Intersection and Union Types," Internal Report, Università di Torino.

Dezani–Ciancaglini, M., de'Liguoro, U., and Piperno, A. (1993), Filter models for a parallel and non deterministic λ-calculus, *in* "MFCS '93," pp. 403–412, Lecture Notes in Computer Science, Vol. 7111, Springer-Verlag, Berlin.

Dezani–Ciancaglini, M., de'Liguoro, U., and Piperno, A. (1994), Fully abstract semantics for concurrent λ-calculus, *in* "TACS '94," pp. 16–35, Lecture Notes in Computer Science, Vol. 789, Springer-Verlag, Berlin.

Gentzen, G. (1969), "The Collected Papers of Gerhard Gentzen," (M. Szabo, Ed.), North-Holland, Amsterdam.

Harrop, R. (1960), Concerning formulas of the types $A \rightarrow B \vee C$, $A \rightarrow (Ex)$ $B(x)$ in intuitionistic formal systems, *J. Symbolic Logic* **25**, 27–32.

Hindley, R. (1982), The simple semantics for Coppo–Dezani–Sallé Types, *in* "ISP '82," pp. 212–226, Lecture Notes in Computer Science, Vol. 137, Springer-Verlag, Berlin.

Hindley, R. (1983), The completeness theorem for typing lambda terms, *Theoret. Comput. Sci.* **22**, 1–17.

Hindley, R. (1984), Coppo–Dezani types do not correspond to propositional logic, *Theoret. Comput. Sci.* **28**, 235–236.

Jensen, T. P. (1992), Disjunctive strictness analysis, *in* "LICS '92," pp. 174–185, IEEE Comput. Soc. Press, Silver Spring, MD.

Knuth, D. E. (1970), Examples of formal semantics, *in* "Symposium on Semantics and Algorithmic Languages," pp. 212–235, Lecture Notes in Mathematics, Vol. 188, Springer-Verlag, Berlin/New York.

Kripke, S. (1959), A completeness theorem in modal logic, *J. Symbolic* **24**, 1–14.

Leivant, D. (1990), Discrete polymorphism, *in* "VI ACM Conference on LISP and Functional Programming," pp. 288–297.

Lopez–Escobar, E. G. K. (1985), Proof functional connectives, pp. 208–221, Lecture Notes in Mathematics, Vol. 1130, Springer-Verlag, Berlin.

MacQueen, D., Plotkin, G., and Sethi, R. (1986), An ideal model for recursive polymorphic types, *Inform. and Control* **71**, 95–130.

Meyer, A. (1982), What is a model of the lambda calculus?, *Inform. and Control* **52**, 87–122.

Meyer, A., Mitchell, J. C., Moggi, E., and Statman, R. (1985), Empty types in polymorphic lambda calculus, *in* "14th ACM Symposium on Principles of Programming Languages," 253–262.

Mitchell, J., and Moggi, E. (1991), Kripke-style models for typed lambda calculus, *Ann. Pure Appl. Logic* **51**, 99–124.

Ong, C.-H. L. (1992), The concurrent lambda calculus 1: A general precongruence theorem for applicative bisimulation, unpublished manuscript, Cambridge Univ. Computer Laboratory.

Ong, C.-H. L. (1993), Non-determinism in a functional setting, *in* "LICS '93," pp. 275–286, IEEE Comput. Soc. Press, Silver Spring, MD.

Pierce, B. (1989), "A Decision Procedure for the Subtype Relation on Intersection Types with Bounded Variables," Tecnical Report CMU-CS-89-169, Carnegie Mellon University.

Pierce, B. (1990), "Preliminary Investigation of a Calculus with Intersection and Union types," Internal Report, Carnegie Mellon University.

Pierce, B. (1991a), "Programming with Intersection Types, Union Types and Polymorphism," Technical Report CMU-CS-91-106, Carnegie Mellon University.

Pierce, B. (1991b), personal communication.

Prawitz, D. (1965), *Natural Deduction*, Almqvist & Wiksell, Stockolm.

Reynolds, J. C. (1988), "Preliminary Design of the Programming Language FORSYTHE," Report CMU-CS-88-159, Carnegie Mellon University.

Reynolds, J. C. (1989), Syntactic control of inference part 2, *in* "ICALP '89," pp. 704–722, Lecture Notes in Computer Science, Vol. 372, Springer-Verlag, Berlin.

Ronchi della Rocca, S., and Venneri, B., (1984), Principal type schemes for an extended type theory, *Theoret. Comput. Sci.* **28**, 151–171.

Routley, R., and Meyer, R. K. (1972), The semantics of entailment, III, *J. Philosophical Logic* **1**, 192–208.

Scott, D. E. (1981), "Lectures on a Mathematical Theory of Computation," Monograph PRG-19, Computing Laboratory, Oxford Univ.

Scott, D. E. (1982), Domains for denotational semantics, *in* "ICALP '82," pp. 577–613, Lecture Notes in Computer Science, Vol. 140, Springer-Verlag, Berlin.

Verschuur, L. (1993), Semantic agreement of anaphors and extended type theory, *in* "Proc. Eastern States Conf. on Linguistics '93," to appear.

Wadsworth, C. (1976), The relation between computational and denotational properties for D_∞-models of the lambda calculus, *SIAM J. Comput.* **5**, 488–521.