Combining Type Disciplines

F. Cardone U. de' Liguoro M. Dezani Ciancaglini

Dedicated to Corrado Böhm on the occasion of his 70th birthday

Abstract

We present a type inference system for pure λ-calculus which includes, in addition to arrow types, also universal and existential type quantifiers, intersection and union types, and type recursion. The interest of this system lies in the fact that it offers a possibility to study in a unified framework a wide range of type constructors. We investigate the main syntactical properties of the system, including an analysis of the preservation of types under parallel reduction strategies, leading to a form of the subject-reduction property. We describe a model for this system where types are special subsets of a D_∞ model for λ-calculus, without imposing any formal contractiveness constraint on types of the kind considered for a closely related system in MacQueen, Plotkin and Sethi [17].

Introduction

Current research in the theory of programming languages has emphasized the relevance of investigating the properties of type disciplines with a sufficiently rich set of type constructors, in order to achieve greater flexibility in the use of type annotations as descriptions of the functional behaviour of programs. One of many possible ways to meet this requirement is exemplified by the family of higher-order, strongly typed languages patterned after the Theory of Constructions (Coquand and Huet [14]), in which a type of a program can be considered as a *logical specification* of it. Another, orthogonal, line of research consists in studying first order type constructors more directly related to the intuition of a type as a collection of values. From the latter viewpoint it is important to emphasize the connections between computational properties of expressions (e.g. strong normalizability, subject reduction or even strictness properties) and the possible forms of their typings. There are by now several examples in the literature which demonstrate the interest of this line of research. The *intersection types* of Coppo and Dezani [12], originally introduced for studying the normalization properties of pure λ-terms, have been shown to characterize exactly the class of strongly normalizable λ-terms (Pottinger [22]). *Recursive types* in which the recursion variable occurs only posistively can be assigned only to strongly normalizing terms (Mendler [18]), and interesting developments in the use of intersection types to detect strictness properties of functional programs have been found recently (Coppo and Ferrari [13]). It turns out that combinations of some of these type constructors offer an abstract environment for studying features coming from the theory and practice of programming. In particular, recent research by John Reynolds (Reynolds [24, 25]) has shown how to encode record types using intersection, and their interaction with general type recursion allows, to a certain extent, to formalize examples from object oriented programming, like that of a class of points described by the type:

$$\text{Point} = \mu P.(\text{setX} \ : \mathsf{Int} \to P \wedge \text{getX} \ : \mathsf{Int}).$$

Benjamin Pierce (Pierce [21]) has also devised the following example, showing that some real expressive power is gained when *union types* are added to intersection types. The $\mathbf{if} \cdot \mathbf{then} \cdot \mathbf{else} \cdot$ operator is such that the type $\mathsf{NegNum} \vee \mathsf{PosNum}$ can be assigned to the term $\mathbf{n} = \mathbf{if}\ b\ \mathbf{then}\ 1\ \mathbf{else}\ -1$. To test whether $\mathbf{n}$ is zero, using the function

$$\mathit{IsZero} : (\mathsf{NegNum} \to \mathsf{False}) \wedge (\mathsf{Zero} \to \mathsf{True}) \wedge (\mathsf{PosNum} \to \mathsf{False}),$$

amounts to derive the typing

$$\mathit{IsZero}\ \mathbf{n} : \mathsf{False}.$$

Without using union types, the best information about *IsZero* **n** is that it is just a boolean value. Recently, ML types have been refined using intersection and union type constructors (Pfenning [15]). Moreover, some examples show the utility of adding intersection types to the Logical Frameworks (Pfenning [20]). In these extensions decidability of type inference (resp. type checking) is preserved by permitting only the intersection of types which are subtypes of a common ML (resp. LF) type.

The form of polymorphism described by universal and existential type quantification has been used in the description of free algebras (Böhm and Berarducci [5]) and abstract data types (Mitchell and Plotkin [19]). More recently, Pierce and Turner have proposed an alternative approach to the embedding of some features of object-oriented programming into higher-order typed systems with existential types, in order to avoid the use of recursive types which are, instead, the main motivation to the introduction of *F-bounded quantification* of Cook, Canning, Hill, Mitchell and Olthoff [6]. In the analysis of object-oriented programming, the interaction of polymorphism and first-order type constructors has also proved to be a flexible tool, especially when a notion of subtyping is added to the system and polymorphic type constructors include bounded universal quantification and record types as in the languages Fun (Cardelli and Wegner [8]) and Quest (Cardelli [7]).

In this perspective, the present paper studies a type inference system which, in addition to the standard function type constructor, has intersection and union operators, type recursion and both universal and existential type quantifiers. We are interested in exploring the consequences of having a wide range of type constructors on the syntactical and semantical properties of the resulting type system. Here we give two equivalent formulations of the type system, one in a natural deduction style and the other in the form of a sequent calculus, and show the invariance of types with respect to a general notion of *parallel reduction* of terms, a property which fails for ordinary β-reduction of λ-terms. The natural deduction formulation coincides with the system introduced by MacQueen, Plotkin and Sethi [17], where, however, no proof-theoretic investigation of the system is carried out, while a semantical interpretation based on a complete metric space of ideals is presented. In the present paper we describe a new interpretation for the system, which does not require any restriction on the formation rules for types as those imposed in that approach, and is based on the approximation properties of D_∞ λ-models (Scott [26]) of which the types are taken to be particular subsets. We do not consider subtyping rules for our system, for their formalization in sequent calculus destroys the symmetry of the rules. It is this symmetry that allows us to prove a restricted form of the Haupsatz, the main technical tool used in establishing the preservation of types under parallel reduction. Observe however that subtyping rules like those of Cardelli and Wegner [8] are satisfied by the semantical interpretation we propose, essentially because it formalizes in a natural way the intuitions on which these rules are based.

Section 1 and 2 are essentially due to the last two authors, while section 3 is essentially due to the first author.

1 The type assignment system.

In the present section we describe the syntax of the type inference system that we shall study in the sequel. The rules for the system are presented in two equivalent ways: the first one is that given by [17], and is patterned after a natural deduction system of the kind considered in [23]. The second formulation is a sequent calculus in the style of [27], for which a restricted form of cut elimination will be proved in the next section.

Definition 1.1 (Types)
The set T of types is inductively defined by

- $t_0, t_1, \ldots \in \mathbf{T}$ *($\mathbf{V}$ set of type variables, ranged over by $s, t, \ldots$)*
- $\omega \in \mathbf{T}$ *(type constant)*
- $\sigma, \tau \Rightarrow (\sigma \to \tau), (\sigma \wedge \tau), (\sigma \vee \tau) \in \mathbf{T}$
- $t \in \mathbf{V}, \tau \in \mathbf{T} \Rightarrow \mu t.\tau, \forall t.\tau, \exists t.\tau \in \mathbf{T}$.

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

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, $\mathrm{FV}(B)$ will denote the set of free term and type variables which occur in B. $B, x : \sigma$ will denote the basis $B \cup \{x : \sigma\}$ when B is a basis such that either $x \notin \mathrm{FV}(B)$ 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) that can be derived by the axioms and rules of Definition 1.2.

Definition 1.2 (Natural Deduction Formulation)
The axioms and rules to derive statements are:

$$(\mathsf{Ax})\ \frac{}{B, x : \sigma \vdash x : \sigma} \qquad (\omega)\ \frac{}{B \vdash M : \omega}$$

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

$$(\mu\mathrm{I})\ \frac{B \vdash M : \sigma[\mu t.\sigma/t]}{B \vdash M : \mu t.\sigma} \qquad (\mu\mathrm{E})\ \frac{B \vdash M : \mu t.\sigma}{B \vdash M : \sigma[\mu t.\sigma/t]}$$

$$(\forall\mathrm{I})\ \frac{B \vdash M : \sigma}{B \vdash M : \forall t.\sigma}\ \text{if } t \notin \mathrm{FV}(B) \qquad (\forall\mathrm{E})\ \frac{B \vdash M : \forall t.\sigma}{B \vdash M : \sigma[\tau/t]}$$

$$(\exists\mathrm{I})\ \frac{B \vdash M : \sigma[\tau/t]}{B \vdash M : \exists t.\sigma} \qquad (\exists\mathrm{E})\ \frac{B, x : \sigma \vdash M : \tau \quad B \vdash N : \exists t.\sigma}{B \vdash M[N/x] : \tau}\ \text{if } t \notin \mathrm{FV}(B) \cup \mathrm{FV}(\tau)$$

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

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

The admissibility of the following rules is straightforward.

$$(\text{Weakening})\ \frac{B \vdash M : \tau}{B, x : \sigma \vdash M : \tau} \qquad (\text{Strengthening})\ \frac{B, x : \sigma \vdash M : \rho}{B \vdash M : \rho}\ \text{if } x \notin \mathrm{FV}(M).$$

This presentation looks familiar and intuitive, and 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 is useful to translate it into a sequent calculus style. To this aim we introduce a type assignment system whose rules are symmetrically introductions to the left and to the right of type constructors. Although we do this primarily for technical reasons, we think that this system is of interest on its own.

This calculus is not a pure sequent calculus since we still consider bases as *sets* of premises, in order both to keep the two systems as close as possible, and to avoid in the following proofs the boring treatment of structural rules. The "multiplicative" character of its rules is however a typical feature of Gentzen's original calculus that we preserve.

A *sequent* is an expression of the form $B :- M : \sigma$ that can be derived by the axioms and rules of Definition 1.3.
We write B, B' to mean $B \cup B'$, provided that this is a still a basis, i.e. x is the subject of the same basic typing statement whenever x is in $\mathrm{FV}(B)$ and in $\mathrm{FV}(B')$.

Definition 1.3 (Sequent Calculus Formulation)
The axioms and rules to derive sequents are:

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

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

$$\frac{B, x:\sigma[\mu t.\sigma/t] :- M:\tau}{B, x:\mu t.\sigma :- M:\tau}\ \mu\mathrm{L} \qquad \frac{B :- M:\sigma[\mu t.\sigma/t]}{B :- M:\mu t.\sigma}\ \mu\mathrm{R}$$

$$\frac{B, x:\sigma[\rho/t] :- M:\tau}{B, x:\forall t.\sigma :- M:\tau}\ \forall\mathrm{L} \qquad \frac{B :- M:\tau}{B :- M:\forall t.\tau}\ \forall\mathrm{R}\ \ \text{if } t\notin \mathrm{FV}(B)$$

$$\frac{B, x:\sigma :- M:\tau}{B, x:\exists t.\sigma :- M:\tau}\ \exists\mathrm{L}\ \ \text{if } t\notin \mathrm{FV}(B)\cup\mathrm{FV}(\tau) \qquad \frac{B :- M:\tau[\rho/t]}{B :- M:\exists t.\tau}\ \exists\mathrm{R}$$

$$\frac{B, x:\sigma :- M:\rho}{B, x:\sigma\wedge\tau :- M:\rho} \quad \frac{B, x:\tau :- M:\rho}{B, x:\sigma\wedge\tau :- M:\rho}\ \wedge\mathrm{L} \qquad \frac{B\vdash M:\sigma \quad B' :- M:\tau}{B, B' :- M:\sigma\wedge\tau}\ \wedge\mathrm{R}$$

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

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

It is immediate to verify that rules (Weakening) and (Strengthening) are admissible also in the sequent formulation.
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, that is each term variable must occur at most once. There is no loss of generality in this restriction. In fact if $B :- M:\sigma$ and $B' :- N:\tau$, using $\wedge$L we can always build B'' and B''' such that $B'' :- M:\sigma$ and $B''' :- N:\tau$, and B'', B''' is a basis. More precisely B'' and B''' will contain $x:\rho\wedge\upsilon$ whenever $x:\rho\in B$ and $x:\upsilon\in B'$.

Notation

(i) If $\mathcal{D}$ is a derivation ending with $B :- M:\sigma$ we write $\mathcal{D} : B :- M:\sigma$.

(ii) We shall 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 involved terms and types will fit the rules as defined in Definition 1.3.

The proof of the equivalence between the two formulations is standard.

Lemma 1.4 (Substitution Lemma)

$$B, x:\sigma\vdash M:\tau \ \text{ and }\ B\vdash N:\sigma \ \Rightarrow\ B\vdash M[N/x]:\tau.$$

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:

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

□

Theorem 1.5 (Equivalence between $\vdash$ and $:-$)
For all bases B, $M \in \Lambda$ and types σ,

$$B \vdash M : \sigma \;\Leftrightarrow\; B :- M : \sigma.$$

Proof. ($\Rightarrow$) By induction on the derivation of $B \vdash M : \sigma$, distinguishing cases according to the last inference. We consider only the interesting cases, i.e. the elimination rules.

Case ($\to$E):

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

becomes

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

Case (χE) where $\chi \in \{\mu, \forall, \wedge\}$:

$$(\chi\mathrm{E})\;\frac{B \vdash M : \tau}{B \vdash M : \sigma}$$

becomes

$$\dfrac{\dfrac{\dfrac{}{x : \sigma :- x : \sigma}\,\mathsf{Ax}}{x : \tau :- x : \sigma}\,\chi\mathrm{L} \qquad \dfrac{\text{ind. hyp.}}{B \vdash M : \tau}}{B :- M : \sigma}\,\mathsf{cut}$$

Case ($\exists$E):

$$(\exists\mathrm{E})\;\frac{B, x : \sigma \vdash M : \rho \quad B \vdash \exists t.N : \sigma}{B \vdash M[N/x] : \rho}$$

becomes

$$\dfrac{\dfrac{\dfrac{\text{ind. hyp.}}{B, x : \sigma :- M : \rho}}{B, x : \exists t.\sigma :- M : \rho}\,\exists\mathrm{L} \qquad \dfrac{\text{ind. hyp.}}{B \vdash N : \exists t.\sigma}}{B :- M[N/x] : \rho}\,\mathsf{cut}$$

since $t \notin \mathrm{FV}(B) \cup \mathrm{FV}(\tau)$.

Case ($\vee$E):

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

becomes

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

($\Leftarrow$) Symmetrically, by induction on the derivation of $B :- M : \sigma$. In case of cut use the Substitution Lemma 1.4. The other non trivial cases are the left rules.

Case $\to$L:

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

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

$$B, B', y : \sigma \to \tau, x : \tau \vdash M : \rho \;\text{ and }\; B, B', y : \sigma \to \tau \vdash N : \sigma.$$

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

Case χL where $\chi \in \{\mu, \forall, \vee\}$:

$$\frac{B, x : \sigma :- M : \tau}{B, x : \rho :- M : \tau}\ \chi\mathrm{L}\ .$$

We have that $B, x : \sigma \vdash M : \tau$ implies $B, x : \rho \vdash M : \tau$ since each occurrence of the axiom

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

can be replaced by the following deduction

$$(\chi\mathrm{E})\ \frac{(\mathsf{Ax})\ \dfrac{}{B, x : \rho \vdash x : \rho}}{B, x : \rho \vdash x : \sigma\ .}$$

Case $\exists$L:

$$\frac{B, x : \sigma :- M : \rho}{B, x : \exists t.\sigma :- M : \rho}\ \exists\mathrm{L} \quad \text{where } t \notin \mathrm{FV}(B) \cup \mathrm{FV}(\rho).$$

By the induction hypothesis $B, x : \sigma \vdash M : \rho$, hence, if y is fresh, $B, y : \sigma \vdash M[y/x] : \rho$, being deductions independent of the names of variables. By weakening $B, x : \exists t.\sigma, y : \sigma \vdash M[y/x] : \rho$ so we can conclude:

$$(\exists\mathrm{E})\ \frac{B, x : \exists t.\sigma, y : \sigma \vdash M[y/x] : \rho \quad B, x : \exists t.\sigma \vdash x : \exists t.\sigma}{B, x : \exists t.\sigma \vdash M : \rho}\ \text{if } t \notin \mathrm{FV}(B) \cup \mathrm{FV}(\tau)$$

since $M[y/x][x/y] \equiv M$.

Case $\vee$L:

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

By the induction hypothesis $B, x : \sigma \vdash M : \rho$, hence, if y is fresh, $B, B', x : \sigma \vee \tau, y : \sigma \vdash M[y/x] : \rho$, being deductions independent of the names of variables and using weakening. Similarly, $B, B', x : \sigma \vee \tau, y : \tau \vdash M[y/x] : \rho$; hence the thesis follows from:

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

□

2 Invariance of types under parallel reduction of subjects.

The type assignments $\vdash$ and $:-$ are not invariant under β-reduction of subjects. The problem is that in rule cut we lose the correspondence between subterms and subdeductions; many occurrences of the same subterm correspond infact to a unique subdeduction. For example one can deduce the type $(\sigma \to \sigma \to \tau) \wedge (\rho \to \rho \to \tau) \to (\xi \to \sigma \vee \rho) \to \xi \to \tau$ both for $\lambda xyz.x(yz)(yz)$ and for $\lambda xyz.x((\lambda u.u)yz)((\lambda u.u)yz)$, but this type cannot be deduced for $\lambda xyz.x(yz)((\lambda u.u)yz)$ or $\lambda xyz.x((\lambda u.u)yz)(yz)$ (Benjamin Pierce showed us this example). Analogously the type $(\forall t.\sigma \to \sigma \to \tau) \to (\xi \to \exists t.\sigma) \to \xi \to \tau$ (where t does not occur in τ) can be deduced for the first two but not for the second two λ-terms. Then in this system types are preserved neither under β-reduction nor under β-expansion. Similar examples show that types are not preserved under η-reduction.

One may ask why the second example does not provide a counterexample to the Type Preservation theorem of [19] which states the invariance of types under β-reduction of subjects. Indeed Mitchell and Plotkin's calculus is a strongly typed calculus and the erasure map from typed terms to pure λ-terms is not onto; now it turns out that the term of type $(\forall t.\sigma \to \sigma \to \tau) \to (\xi \to \exists t.\sigma) \to \xi \to \tau$ whose erasure is $\lambda xyz.x((\lambda u.u)yz)((\lambda u.u)yz)$ does not reduce to any term corresponding either to $\lambda xyz.x(yz)((\lambda u.u)yz)$ or to $\lambda xyz.x((\lambda u.u)yz)(yz)$; on the contrary its immediate reduct corresponds to $\lambda xyz.x(yz)(yz)$.

Invariance under β-reduction is a desirable property, since it gives us the feeling that the rules in the system are, in a sense, correct. But in the present case rules $\exists$E, $\vee$E and cut could hardly be considered

uncorrect; indeed, beside their similarity with rules of the sequent calculus in logic, they turn out to be sound e.g. with respect to the semantics we propose in section 3 of the present paper.

Instead it seems that the phenomenon we are about arises from a mismatch between the type system and the possibility of unbalanced reductions of the subject. In fact we prove the system $:-$ to be invariant under parallel reduction, hence, by the equivalence of $:-$ and $\vdash$, the same property is established for $\vdash$. More precisely types are invariant under β-reductions which are done simultaneously on all occurences of the same subterm which corresponds to the same subdeduction. Since this reduction relation is cofinal with the usual β-reduction, we conclude that each term will have, soon or later, all the types of its expansions, which is in particular true for normal forms.

Our proof is inspired by Gentzen's proof of the Hauptsatz [27]. Following his lines we introduce the notions of rank and degree of cuts, that are used to show that the cut elimination procedure terminates. A similar but simpler proof is carried out for a sistem without μ, $\forall$ and $\exists$ in [2].

The main difference between our proof and that by Gentzen lies in the fact that we do not have a normal from property (indeed in presence of the universal type and of recursive types also non strongly noramlizable terms are typable), but only a property of preservation of types under the complete development of a suitable set $\mathcal{F}$ of redexes (a "uniform set", see Definition 2.5).

Some technical remarks about the definition of degree are in order. Firstly the degree of a cut is defined to be different from 0 only if this cut is relative to a redex belonging to $\mathcal{F}$ (see Definition 2.11). Secondly union and intersection are proof functional connectives, in the sense of Lopez-Escobar [**?**]: 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: this is the reason why arrows have a minimal degree. Finally problems arise from rules μL, μR, $\forall$L and $\exists$R, since the cut elimination in general results in a cut whose cut type is more complex than the original one: this contradicts the fact that the proof has been shortened, so that the obvious solution is to make degree a function of the "history" of the type occurrence we are considering.

In each rule of the sequent calculus it is natural to distinguish between the type occurrence which has been just generated from the other ones. This can be formalized as follows.

Definition 2.1 (Fathers and Generated Types)

(i) Given any arbitrary rule

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

we say that:

- *the occurrence of σ_i in a premise is the* father *of the occurrence of τ iff $\sigma_i \equiv \tau$ (this is true in all left rules and in rule* cut*);*
- *if $x : \nu \in B_i$ and $x : \xi \in B$ with $\nu \equiv \xi$ then the occurrence of ν is a* father *of the occurrence of ξ;*
- *if the rule isa cut of the shape in Figure 1 then both occurrences of $x : \sigma$ and of $y : \sigma$ in theleft premise are fathers of the occurrence of $x : \sigma$ in the conclusion.*

(ii) Looking at the shapes of logical rules as shown in Figure 1 we say that:

- *$\sigma \to \tau$ is the type* generated *by $\to L$ and $\to R$ from σ and τ;*
- *σ is the type* generated *by χL and $\chi' R$ from τ;*
- *$\sigma \wedge \tau$ is the type* generated *by $\wedge R$ from σ and τ;*
- *$\sigma \vee \tau$ is the type* generated *by $\vee L$ from σ and τ.*

Clearly if $i \neq 0$ each type either is generated or it has at least one father, the only exception being rule $\to$L, where the generated type may have two fathers..

The standard definition of type degree is the number of type symbols. We need a more refined notion of type degree which depends on the structure of the derivation rather than on the syntax of the types; in particular it should give the same degree to all arrow types and distinguish between occurrences of the same non arrow type. In fact we need take to take into account the number of rules which have been used in generating the occurrence of non arrow types. This will be clarified in remark 2.12.

$$\frac{B, x:\sigma, y:\sigma :- M:\rho \quad B, x:\sigma :- x:\sigma}{B, x:\sigma :- M[x/y]:\rho}\ \mathsf{cut}$$

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

$$\chi\mathrm{L}\ \frac{B, x:\tau :- M:\rho}{B, x:\sigma :- M:\rho}\ \chi\in\{\wedge,\forall,\exists,\mu\} \qquad \chi'\mathrm{R}\ \frac{B :- M:\tau}{B :- M:\sigma} \quad \chi'\in\{\vee,\forall,\exists,\mu\}$$

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

Figure 1: Possible shapes of logical rules.

Definition 2.2 (Degree of Type Occurrences) *The* degree *of an occurrence of a type σ in a deduction $\mathcal{D}$ is defined by induction on $\mathcal{D}$:*

- *if σ occurs in the conclusion of* Ax *or ω then its degree is* 0*;*
- *if σ occurs in the conclusion of a logical rule or of the cut and if it is not the generated type, then its degree is the maximum of the degrees of its fathers;*
- *if $\sigma\rightarrow\tau$ is the type generated by $\rightarrow L$ then its degree is the maximum between the degrees of its fathers and* 0.5*;*
- *if $\sigma\rightarrow\tau$ is the type generated by $\rightarrow R$ then its degree is* 0.5*;*
- *if $\sigma\wedge\tau$ ($\sigma\vee\tau$) is the type generated by $\wedge R$ ($\vee L$) from σ and τ then its degree is the maximum of the degrees of σ and τ plus* 1*;*
- *if σ is the type generated by χL ($\chi\in\{\mu,\forall,\exists,\wedge\}$) or by $\chi' R$ ($\chi'\in\{\mu,\forall,\exists,\vee\}$) from τ then its degree is the degree of τ plus* 1.

Definition 2.3 (Cut-type and Rank) *Given a cut*

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

we call σ the cut-type.

(i) The left rank *of the cut is the largest number of consecutive statements in a path such that the lowest of these is the left premise of the cut and the basis of each of these statements contains $x:\sigma$.*

(ii) The right rank *of the cut is the largest number of consecutive statements in a path such that the lowest of these statements is the right premise of the cut and σ is the predicate of the succedent of each of these statements.*

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

Definition 2.4 (Ready Cuts and Ready Derivations)

(i) A cut *is* ready *iff it has rank* 2*;*

(ii) a derivation *is* ready *iff it contains only ready cuts.*

$$\text{(a)}\quad \dfrac{\dfrac{B, z : \xi :- P : \tau \qquad B' :- R : \nu}{B, B', x : \nu \to \xi :- P[xR/z] : \tau}\,{\to}\mathrm{L} \qquad \dfrac{B'', y : \nu :- Q : \xi}{B'' :- \lambda y.Q : \nu \to \xi}\,{\to}\mathrm{R}}{B, B', B'' :- P[xR/z][\lambda y.Q/x] : \tau}\,\mathsf{cut}$$

$$\text{(b)}\quad \dfrac{\dfrac{B, x : \xi :- P : \tau}{B, x : \nu :- P : \tau}\,\chi\mathrm{L} \qquad \dfrac{B' :- Q : \xi'}{B' :- Q : \nu}\,\chi\mathrm{R}}{B, B' :- P[Q/x] : \tau}\,\mathsf{cut}$$

where $\chi \in \{\mu, \exists, \forall\}$

$$\text{(c)}\quad \dfrac{\dfrac{B, x : \xi :- P : \tau}{B, x : \xi \wedge \nu :- P : \tau}\,{\wedge}\mathrm{L} \qquad \dfrac{B' :- Q : \xi \qquad B'' :- Q : \nu}{B', B'' :- Q : \xi \wedge \nu}\,{\wedge}\mathrm{R}}{B, B', B'' :- P[Q/x] : \tau}\,\mathsf{cut}$$

$$\text{(d)}\quad \dfrac{\dfrac{B, x : \xi :- P : \tau \qquad B'', x : \nu :- P : \tau}{B, B'', x : \xi \vee \nu :- P : \tau}\,{\vee}\mathrm{L} \qquad \dfrac{B' :- Q : \xi}{B' :- Q : \xi \vee \nu}\,{\vee}\mathrm{R}}{B, B', B'' :- P[Q/x] : \tau}\,\mathsf{cut}$$

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

$$\text{(f)}\quad \dfrac{\dfrac{}{B, x : \sigma :- x : \sigma}\,\mathsf{Ax} \qquad \dfrac{B_i :- M : \sigma_i}{B' :- M : \sigma}\,\chi\mathrm{R}}{B, B' :- M : \sigma}\,\mathsf{cut}$$

where χ is any type constructor

$$\text{(g)}\quad \dfrac{\dfrac{}{B, x : \omega :- x : \omega}\,\Theta \qquad \dfrac{}{B' :- M : \omega}\,\omega}{B, B' :- M : \omega}\,\mathsf{cut}$$

where Θ is either Ax or ω

Figure 2: Possible shapes of ready cuts.

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. Figure 2 shows all possible shapes of ready cuts.

We introduce now a notion of parallel reduction. The parallelism comes out from the fact that in each reduction step more than a single redex is contracted.

In order to formalize this idea we define the notion of *uniform* set of redex occurrences in a term. Informally a set of redex occurrences in a term M is called uniform if, whenever it contains a redex occurrence, every other occurrence of the same redex is in the set as well.

As usual any λ-term M can be identified with its binary syntactical tree, where nodes are represented by the subset $\{0,1\}^*$ which is 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 $\langle \alpha, N \rangle$.

Definition 2.5 (Uniform and 1-uniform Set of Redexes) *Let M be any λ-term, then:*

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

(ii) $\mathsf{Red}(M) = \{\langle \alpha, (\lambda x.P)Q \rangle \in \mathsf{Occ}(M) \mid P, Q$ *any terms*, x *any variable*$\}$;

(iii) $\mathcal{F} \subseteq \mathsf{Red}(M)$ *is* uniform *iff*

$$\langle \alpha, R \rangle \in \mathcal{F} \wedge \langle \beta, R \rangle \in \mathsf{Red}(M) \Rightarrow \langle \beta, R \rangle \in \mathcal{F};$$

(iv) $\mathcal{F} \subseteq \mathsf{Red}(M)$ *is* 1-uniform *iff* $\mathcal{F}$ *is uniform and* $\{R \mid \exists \alpha.\ \langle \alpha, R \rangle \in \mathcal{F}\}$ *is a singleton.*

In the sequel we need the notions of residual and development, which are well known concepts of λ-calculus theory: we refer the reader to [3] chapter 11, from which we take the notation.

Definition 2.6 (Parallel Reduction) *The reduction relation $\Rightarrow_p$ over λ-terms is defined by:*

$$M \Rightarrow_p N \text{ iff } \exists \mathcal{F} \subseteq \mathsf{Red}(M).\ \mathcal{F} \text{ is uniform and } (M, \mathcal{F}) \twoheadrightarrow_{cpl} N$$

where $(M, \mathcal{F}) \twoheadrightarrow_{cpl} N$ is the complete development of $(M, \mathcal{F})$ (see [3], 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 [4] and [16] (see also [3], 13.2.7).

Definition 2.7 (Gross-Knuth Reduction)

$$M \rightarrow_{gk} N \text{ iff } (M, \mathsf{Red}(M)) \twoheadrightarrow_{cpl} N.$$

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

Lemma 2.8 *Let $\mathcal{F} \subseteq \mathsf{Red}(M)$ be uniform and $\mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_m$ where each $\mathcal{F}_i$ is 1-uniform. If $(M, \mathcal{F}) \rightarrow \rightarrow_{cpl} N$, then there are $N_0, \ldots, N_m$ and a permutation π of order m such that $M \equiv N_0$, $N \equiv N_m$ and for all $1 \leq i \leq m$,*

$$(N_{i-1}, \mathcal{F}'_{\pi(i)}) \twoheadrightarrow_{cpl} N_i$$

where $\mathcal{F}'_{\pi(i)}$ is the set of residuals of redexes in $\mathcal{F}_{\pi(i)}$ with respect to the β-reduction $M \xrightarrow{*} N_{i-1}$, and $\mathcal{F}'_{\pi(i)}$ is 1-uniform.

Proof. We introduce an ordering on the 1-uniform sets of redexes of M. Let $\mathcal{F} = \{\langle \alpha_h, R \rangle \mid h \in H\}$ and $\mathcal{F}' = \{\langle \alpha_k, R' \rangle \mid k \in K\}$; then

$$\mathcal{F} \leq \mathcal{F}' \text{ iff } \exists h \in H\ \forall k \in K.\ \alpha_k \leq_{lex} \alpha_h,$$

where $\leq_{lex}$ is the lexicographic order. This is a total ordering and we take π as the permutation s.t. for all i, $\mathcal{F}_{\pi(i)} < \mathcal{F}_{\pi(i+1)}$. If $i < j$ then $\mathcal{F}_{\pi(i)} < \mathcal{F}_{\pi(j)}$, so that all occurrences of $R' \in \mathcal{F}_{\pi(j)}$ either contain the same copies of redexes $R \in \mathcal{F}_{\pi(i)}$ by 1-uniformity, in which case they are modified in the same way while

contracting the R; or they do not contain any such occurrence, in which case they remain unchanged. In any case their residuals are syntactically equal, so that the $\mathcal{F}'_{\pi(i)}$ are 1-uniform. Now the thesis follows from the uniqueness of the result of a complete development (see [3], Theorem 11.2.25) since then the reduction $(M, \mathcal{F}) \twoheadrightarrow_{cpl} N$ can be rearranged as we need.

□

Definition 2.9 (Cuts generating $\mathcal{F}$-redexes) *Let $\mathcal{D}$ be a deduction whose conclusion subject is M; fix a uniform $\mathcal{F} \subseteq \mathsf{Red}(M)$; finally let*

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

be a cut in $\mathcal{D}$. We say that the above cut generates $\mathcal{F}$-redexes of M *iff P has at least a subterm of the shape xR, such that all occurrences of $(\lambda y.Q')R'$ in M belong to $\mathcal{F}$, where $(\lambda y.Q')R'$ is a fixed instance of $(\lambda y.Q)R$.*

Remark 2.10 We have to consider the substitution instance $(\lambda y.Q')R'$ instead of $(\lambda y.Q)R$ itself because of the possibility that other cuts, placed under the given one in the current deduction, substitute variables inside $(\lambda y.Q)R$.

Definition 2.11 (Cut Degree) *Let $\mathcal{D}$ be a deduction whose conclusion subject is M; fix a uniform $\mathcal{F} \subseteq \mathsf{Red}(M)$. Moreover let*

$$\mathsf{cut}\ \frac{B, x : \sigma :- P : \tau \qquad B' :- Q : \sigma}{B, B' :- P[Q/x] : \tau}$$

be a cut inference in $\mathcal{D}$. Then the degree *of this cut in $\mathcal{D}$ relative to $\mathcal{F}$ is defined by cases:*

- *if this cut does not generate $\mathcal{F}$-redexes then its degree is 0;*
- *otherwise its degree is the sum of the degrees of the occurrences of its cut-type in the left and right premises of the cut.*

Remark 2.12 Our definition of cut degree is not the classical one, which simply records the number of symbols of the cut-type. The distinctive features of our cut degree are:

(i) if the degree is 0 then the cut does not generate $\mathcal{F}$-redexes;

(ii) if the degree is 1 then the cut is ready and the cut-type is an arrow;

(iii) if the degree is greater than 1 and the cut-type is an arrow then the cut is not ready;

(iv) otherwise the degree is a function of the number of inferences of the form χL an χR which have been used in generating the occurrences of the cut-type.

Feature (i) has to be so because we will eliminate only cuts generating $\mathcal{F}$-redexes. Moreover when eliminating a cut generating an $\mathcal{F}$-redex, whose cut-type is of the form $\sigma \to \tau$, the redexes generated by the new cuts produced by the contraction cannot be in $\mathcal{F}$; hence their degree relative to $\mathcal{F}$ will be 0. Indeed if the contraction of a redex in $\mathcal{F}$ produces new redexes, these will never be contracted in the development of $\mathcal{F}$.
Feature (ii) will imply that, if in a ready deduction the maximum degree of a cut generating $\mathcal{F}$-redexes is 1, then all cuts generating $\mathcal{F}$-redexes have as cut-type an arrow type (i.e. they have the shape of Figure 2 (a)). As a matter of fact our proof of subject reduction (proof of 2.16) actually hints to an algorithm which performs a preliminary elimination of all "logical" cuts, which does not affect the subject, ending with the "parallel" elimination of arrow cuts, which causes actual β-contraction.
Feature (iv) takes into account that when we shall eliminate a ready cut whose premises are conclusions of say $\forall$L and $\forall$R, the number of symbols of the cut-type of the newly generated cut may well be greater than that of the original cut-type, despite of the intuition that the derivation has been semplified. Note that in this third case the degree of the cut is always greater than 2.

Definition 2.13 (Degree of Deductions)

(i) Let $\mathcal{D}$ be a deduction whose conclusion subject is M; fix a uniform $\mathcal{F} \subseteq \mathsf{Red}(M)$. The degree $\delta(\mathcal{D}, \mathcal{F})$ *of $\mathcal{D}$ relative to $\mathcal{F}$ is the highest degree of a cut in $\mathcal{D}$.*

(ii) Let $\mathcal{D} : B :- M : \sigma$ and $\mathcal{D}' : B' :- M : \sigma$; then $\mathcal{D}'$ is non-increasing *with respect to $\mathcal{D}$ iff $\delta(\mathcal{D}', \mathcal{F}) \leq \delta(\mathcal{D}, \mathcal{F})$ for all uniform $\mathcal{F} \subseteq \mathsf{Red}(M)$.*

We prove that we can transform deductions by adding assumptions, by changing the names of (free) term variables and by eliminating the cuts with shapes (f) and (g) of Figure 2, preserving readiness and non increasing the cut degrees.

We say that a deduction $\mathcal{D}'$ is *similar* to a deduction $\mathcal{D}$ iff $\mathcal{D}'$ is obtained from $\mathcal{D}$ by adding (basic) typing statements to the bases, renaming term variables and substituting type expressions to free occurrences of type variables. This means that $\mathcal{D}'$ and $\mathcal{D}$ have the same deduction tree and differ only for the bases and for the names of term variables and, possibly, because type-occurrences are substitution instances of their counterparts in $\mathcal{D}$. Since the notion of degree does not depend on the length of type expressions involved in cuts, but rather on the structure of the derivation in which they occurr, similar deductions have the same degree with respect to any uniform $\mathcal{F}$.

Lemma 2.14

(i) If $\mathcal{D} : B :- M : \sigma$ and $B' \supseteq B$ then there is $\mathcal{D}' : B' :- M : \sigma$ such that $\mathcal{D}'$ is similar to $\mathcal{D}$.

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

(iii) If $\mathcal{D} : B :- M : \sigma$, τ is a type and t is any type variable, then there is $\mathcal{D}' : B[\tau/t] :- M : \sigma[\tau/t]$ such that $\mathcal{D}'$ is similar to $\mathcal{D}$.

(iv) All occurrences of ready cuts of the shapes (f) and (g) of Figure 2 can be eliminated preserving readiness and obtaining a non-increasing deduction of the same statement.

Proof. (i). Easy by adding the premises of $B' - B$ to each statement, possibly renaming some variables not in B.
(ii). Immediate, since deductions are independent of the names of (free) term variables.
(iii). By definition of similarity.
(iv). Let us consider a cut of the shape (f) or (g):

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

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

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 which satisfies the given conditions. □

The following Lemma (proved in the Appendix) claims that every deduction can be transformed into a ready deduction of the same statement. This can be accomplished without increasing the complexity of the proof. Formally:

Lemma 2.15 (Rank Lemma) *For any derivation $\mathcal{D} : B :- M : \sigma$ there exists a ready derivation $\mathcal{D}' : B :- M : \sigma$ such that $\mathcal{D}'$ is non increasing with respect to $\mathcal{D}$.*

Theorem 2.16 (Invariance of Types under Parallel Reduction)

(i) $B :- M : \rho$ and $M \Rightarrow_p N$ imply $B :- N : \rho$.

(ii) $B \vdash M : \rho$ *and* $M \Rightarrow_p N$ *imply* $B \vdash N : \rho$.

Proof. (i) If $M \Rightarrow_p N$ then for some $\mathcal{F} \subseteq \mathsf{Red}(M)$, $(M, \mathcal{F}) \twoheadrightarrow_{cpl} N$. By Lemma 2.8 this reduction can be splitted into a finite number of steps, whose $\mathcal{F}$'s are 1-uniform. Hence it suffices to prove the thesis when $\mathcal{F}$ is 1-uniform. On the other hand by Lemmas 2.14 (iv) and 2.15, we can assume that the given $\mathcal{D} : B :- M : \rho$ is ready and does not contain cuts of shapes (f) and (g). Since redexes in $\mathcal{F}$ are exactly those which determine the positive degrees of cuts, we prove the theorem as a kind of cut elimination: indeed it is a such, being parametrized with respect to $\mathcal{F}$.
If $\delta(\mathcal{D}, \mathcal{F}) = 0$ and $\mathcal{D}$ is ready, then no cut generates $\mathcal{F}$-redexes, hence $\mathcal{F}$-redexes may occur only as subterms of terms of type ω: in this case clearly $B :- N : \rho$ is derivable introducing by rule ω the reducta of those subterms of M.
If $\delta(\mathcal{D}, \mathcal{F}) = 1$ then all cuts which generate $\mathcal{F}$-redexes have an arrow type as cut-type; hence, by the readiness of $\mathcal{D}$, they have the shape

$$\dfrac{\dfrac{B, z : \xi :- P : \tau \quad B' :- R : \nu}{B, B', x : \nu \to \xi :- P[xR/z] : \tau}\,{\to}\mathrm{L} \qquad \dfrac{B'', y : \nu :- Q : \xi}{B'' :- \lambda y.Q : \nu \to \xi}\,{\to}\mathrm{R}}{B, B', B'' :- P[xR/z][\lambda y.Q/x] : \tau}\ \mathsf{cut}$$

Then all these cuts can be replaced as follows:

$$\dfrac{B, z : \xi :- P : \tau \qquad \dfrac{B'', y : \nu :- Q : \xi \quad B' :- R : \nu}{B', B'' :- Q[R/y] : \xi}\ \mathsf{cut}}{B, B', B'' :- P[Q[R/y]/z] : \tau}\ \mathsf{cut}$$

and in the remaining parts of $\mathcal{D}$ the subjects have to be reduced (by $\Rightarrow_p$) in order to match $P[Q[R/y]/z]$. Eventually we get a deduction of degree 0.
If $\delta(\mathcal{D}, \mathcal{F}) > 1$ then all cuts with degree $\delta(\mathcal{D}, \mathcal{F})$ are of the shape (b), (c) or (d) of Figure 2. In fact all cuts are ready and moreover all cuts of shape (e) have degree 0, since they cannot generate any redex. We replace the cuts of shape (b) when $\chi = \exists$:

$$\dfrac{\dfrac{B, x : \sigma :- P : \tau}{B, x : \exists t.\sigma :- P : \tau}\,\exists\mathrm{L} \qquad \dfrac{B' :- Q : \sigma[\rho/t]}{B' :- Q : \exists t.\sigma}\,\exists\mathrm{R}}{B, B' :- P[Q/x] : \tau}\ \mathsf{cut}$$

by

$$\dfrac{B, x : \sigma[\rho/t] :- P : \tau \quad B' :- Q : \sigma[\rho/t]}{B, B' :- P[Q/x] : \tau}\ \mathsf{cut}$$

where the left premise of the new cut is obtained by a derivation similar to the subderivation whose conclusion was $B, x : \sigma :- P : \tau$, which is non-increasing by Lemma 2.14 (iii). The degree of each occurrence of the cut-type decreases by 1, so that the degree of the cut is lowered by 2. The cases $\chi = \forall$ or $\chi = \mu$ are similar. Finally cuts of shape (c) or (d) are replaced by

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

where some basic statements are added to the bases of the premises using Lemma 2.14 (i). It is easy to verify that we obtain cuts of lower degrees.
After these transformations have been performed we get a derivation, say $\mathcal{D}'$, of lower degree; we then apply Lemma 2.15 to obtain a ready derivation which is non-increasing with respect to $\mathcal{D}'$, so that the inductive hypothesis applies.
(ii) Immediate form (i) and from Theorem 1.5. □

Corollary 2.17 (Invariance under Gross-Knuth Reduction)

(i) $B \vdash M : \sigma,\ M \to_{gk} N \Rightarrow B \vdash N : \sigma$.

(ii) $B \vdash M : \sigma,\ M \xrightarrow{*}_\beta N \Rightarrow \exists L.\ N \xrightarrow{*}_\beta L$ *and* $B \vdash L : \sigma$.

Proof. (i) Immediate from Theorem 2.16 (ii) and the remark after Definition 2.7.
(ii) Is a consequence of (i) and of the cofinality of $\rightarrow_{gk}$ with respect to $\rightarrow_\beta$, proved in [3] 13.2.11. □

Note that the above proof cannot be transformed into a proof of normalization of λ-terms (which does not hold since we allow recursive type and the universal type ω). Indeed the parameter $\mathcal{F}$ is essential to the proof and cannot be dropped; what we have actually shown is a kind of typed version of the finite development theorem of the untyped λ-calculus.

3 Semantics

In the present section we describe the construction of a model for our type system, based on a technique introduced in Coppo [11] that permits to interpret type recursion even when type constructors are not monotonic with respect to the (semantical) relation of subtyping induced by the interpretation of types as sets of values. This technique differs from that used by MacQueen, Plotkin and Sethi [17] in exploiting only the approximation properties of the domain in which untyped terms are interpreted, without any appeal to a complete metric space structure that can be defined on the collection of (denotations of) types. This approach to type interpretations has some advantages over that followed by MacQueen, Plotkin and Sethi [17], in not requiring any restriction on the formation rules for types and also in offering a powerful proof technique for many semantical properties of the type system (see Cardone and Coppo [10] for some examples).

Starting from a model D_∞ of the λ-calculus, obtained as the inverse limit of a sequence of complete lattices as in Scott [26], each type will be interpreted as a subset of D_∞ satisfying constraints derived from the order-theoretic nature of this model. For example, as every type can be assigned to the combinator $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$, and the interpretation of Ω in every continuous λ-model D is the least element $\perp$ (see Barendregt [3]), we require that each of the subsets of D_∞ that we will consider as candidates for the interpretation of types should contain $\perp$. Furthermore, if $A \subseteq D_\infty$ is such a subset and $\Delta \subseteq A$ is a directed set, then also $\bigsqcup \Delta \in A$: this condition yields the correctness of the typing of the fixed-point combinator Y as a term of type $(\sigma \rightarrow \sigma) \rightarrow \sigma$, for any type σ.

The explicit construction of the domain D_∞ (refer to Barendregt [3] for a detailed description) starts from the two-points lattice $D_0 = \{\perp, \top\}$ and, given the lattice D_n, the lattice D_{n+1} is defined to be the lattice of *continuous functions* $[D_n \rightarrow D_n]$ from D_n to itself, with an embedding $\phi_n : D_n \rightarrow D_{n+1}$ and a projection $\psi_n : D_{n+1} \rightarrow D_n$. The inverse limit of the diagram

$$\ldots \xleftarrow{\psi_{n-2}} D_{n-1} \xleftarrow{\psi_{n-1}} D_n \xleftarrow{\psi_n} D_{n+1} \xleftarrow{\psi_{n+1}} \ldots$$

satisfies the recursive domain equation $D_\infty \cong [D_\infty \rightarrow D_\infty]$, with isomorphisms $\Phi : D_\infty \rightarrow [D_\infty \rightarrow D_\infty]$ and $\Psi : [D_\infty \rightarrow D_\infty] \rightarrow D_\infty$. This induces an operation of application over D_∞ defined by

$$d \cdot e = \Phi(d)(e) \text{ for } d, e \in D_\infty.$$

The construction of D_∞ endows it with an approximation structure, orthogonal to its order structure, which takes the form of a denumerable sequence of continuous mappings $(\cdot)_n : D_\infty \rightarrow D_\infty$, that provide an image of each D_n inside D_∞, and in fact we shall identify in the sequel D_n with the finite subset $\{d_n \mid d \in D_\infty\}$ of D_∞. In particular, the following properties will be used without explicit mention throughout this section (see Barendregt [3], 18.2.8 and 18.2.9):

Proposition 3.1 (Properties of approximation mappings) *The family* $\{(\cdot)_n : D_\infty \rightarrow D_\infty\}_{n \in I\!N}$ *satisfies the following conditions, for all* $f, d \in D_\infty$ *and* $m, n, k \in I\!N$*:*

1. *If* $d = \top$ *then* $d_0 = \top$*, otherwise* $d_0 = \perp$.
2. $(d_n)_m = (d_m)_n = d_{\min(m,n)}$.
3. $d = \bigsqcup_{n \in I\!N} d_n$.
4. *If* $n \leq k$ *then* $f_{n+1} \cdot d_k = f_{n+1} \cdot d_n$.

5. *If* $n \leq k$ *then* $(f_{k+1} \cdot d_n)_n = f_{n+1} \cdot d_n$.

6. $f_{n+1} \cdot d_n = f_{n+1} \cdot d = (f \cdot d_n)_n$.

7. $f = f_{n+1}$ *if and only if* $\forall d \in D_\infty . f \cdot d = (f \cdot d_n)_n$.

A consequence of this Proposition is that for $d, e \in D_\infty$, $d \cdot e = \bigsqcup_{n \in I\!N} d_{n+1} \cdot e_n$.

We shall require each subset A of D_∞ which interprets a type to be closed under each of these mappings, in the sense that $d \in A$ implies that $d_n \in A$ for all $n \in I\!N$. We obtain in this way a collection of subsets of D_∞ that is closed under a wide range of type constructors, and also offers some technical advantages over *ideals* that have been often used in the semantics of type inference systems (see, for example, MacQueen, Plotkin and Sethi [17]).

Definition 3.2 (Profinite Subsets) *(i) The class* $\mathbf{P}$ *of* profinite subsets *of* D_∞ *is defined to consist of those subsets* A *of* D_∞ *such that:*

- $\bot, \top \in A$;
- A *is* complete*: if* $\langle d^{(i)} \rangle_{i \in I\!N}$ *is an increasing chain in* D_∞ *such that* $\forall i \in I\!N . d^{(i)} \in A$*, then* $\bigsqcup_{i \in I\!N} d^{(i)} \in A$;
- A *is* closed under approximations*:* $d \in A$ *implies* $d_n \in A$*, for all* $n \in I\!N$.

(ii) For profinite subsets A, B *of* D_∞*, the relation* $A \lhd_n B$ *is defined to hold if and only if* $A \subseteq B$ *and* $d_n \in A$ *whenever* $d \in B$.

Note that $\top$ belongs to every profinite subset of D_∞: this would not be possible if we used ideals (the only ideal containing $\top$ is D_∞). Observe the close connection between this class of subsets of D_∞ and the class of relations used in Amadio [1] and Cardone [9] in the construction of models for strongly typed, polymorphic languages with type recursion and subtyping.

Obviously, the intersection of an arbitrary family $\mathcal{X}$ of profinite subsets of D_∞ is profinite, so the structure $\langle \mathbf{P}, \subseteq \rangle$ is a complete lattice having $\{\bot, \top\}$ and D_∞ as bottom and top elements, respectively. We shall denote the infinitary lattice operations by **inf** and **sup** whenever convenient (observe however that the supremum of a finite family of profinite subsets coincides with their set-theoretic union). The following Proposition collects the main properties of this lattice. For a profinite subset A, the notation A_n denotes the set $\{d_n \mid d \in A\}$.

Proposition 3.3 (Properties of Profinite Subsets) *For* $A \in \mathbf{P}$ *and any* $n \in I\!N$*:*

1. $A_n \subseteq A$;
2. $A_n \in \mathbf{P}$;
3. $A_n \lhd_n A_{n+1}$;
4. *For any family* $\mathcal{X} \subseteq \mathbf{P}$, $(\bigcup_{A \in \mathcal{X}} A_n) \in \mathbf{P}$;
5. *For any family* $\mathcal{X} \subseteq \mathbf{P}$ *we have*
$$(\mathbf{sup}\mathcal{X})_n = \bigcup_{A \in \mathcal{X}} A_n.$$

Proof.(1) Immediate, as A is closed under approximations.
For proving (2), observe that A_n is complete because every finite subset of D_∞ is obviously complete. Furthermore, in order to show that A_n is also closed under approximations, let $d \in A_n$. There are two cases:

- $m > n$: $d_m = d$ (because $d \in D_n$), so $d_m \in A_n$.
- $m \leq n$: As $d \in A$ (by (1)), $d_m \in A$, and $d_m = (d_m)_n \in A_n$,

concluding the proof of (2).
(3) To show that $A_n \lhd_n A_{n+1}$, assume $d_n \in A_n$. This implies $d_n \in A$ by (1) so $d_n \in A_{n+1}$ because $(d_n)_{n+1} = d_n$. If $d \in A_{n+1}$ then $d \in A$ by (1), so $d_n \in A_n$.
(4) Easy, observing that we have a finite, hence also complete, subset of D_∞.
(5) It is easily checked that the following two properties hold:

- $\bigcup_{A\in\mathcal{X}} A_n \subseteq \bigcup_{A\in\mathcal{X}} A_{n+1}$
- $d \in \bigcup_{A\in\mathcal{X}} A_{n+1} \Longrightarrow d_n \in \bigcup_{A\in\mathcal{X}} A_n$

yielding that the set $U = \{d \in D_\infty \mid \forall n \in \mathbb{N}.d_n \in \bigcup_{A\in\mathcal{X}} A_n\}$ is profinite (using (4)), and is such that $\bigcup_{A\in\mathcal{X}} A_n \subseteq U$ for each $n \in \mathbb{N}$. Given any upper bound $S \in \mathbf{P}$ of $\bigcup\mathcal{X}$ then $S \supseteq U$ because

$$\begin{aligned} d \in U &\Rightarrow \forall n \in \mathbb{N}.d_n \in \bigcup_{A\in\mathcal{X}} A_n \\ &\Rightarrow \forall n \in \mathbb{N}.d_n \in \bigcup\mathcal{X} \text{ as } A_n \subseteq A \text{ by (3)} \\ &\Rightarrow \forall n \in \mathbb{N}.d_n \in S \\ &\Rightarrow d = \bigsqcup_{n\in\mathbb{N}} d_n \in S. \end{aligned}$$

So $\mathbf{sup}\mathcal{X} = U$ and $(\mathbf{sup}\mathcal{X})_n = \bigcup_{A\in\mathcal{X}} A_n$. □

Definition 3.4 *If A, B are profinite subsets of D_∞, we introduce the following notations:*

- $A \to B = \{d \in D_\infty \mid \forall a \in A.d \cdot a \in B\}$;
- $A_n \to^{n+1} B_n = \{d \in D_{n+1} \mid \forall a \in A_n.d \cdot a \in B_n\}$.

Then $A \to B$ and $A_n \to^{n+1} B_n$ are also profinite. We consider only the case of $A_n \to^{n+1} B_n$. Completeness follows from the fact that $A_n \to^{n+1} B_n$ is a finite subset of D_∞. In order to show that this subset is also closed under approximations, take $d \in A_n \to^{n+1} B_n$. For any $k > 0$ and $a \in A_n$, observe that $a_{k-1} \in A_n$ being A_n a profinite subset. So $(d \cdot a_{k-1}) \in B_n$ and also $d_k \cdot a = d_k \cdot a_{k-1} = (d \cdot a_{k-1})_{k-1} \in B_n$.

We shall define the interpretation of a type by means of a construction which involves denumerably many approximations, matching the approximation structure of the λ-model in which the untyped terms are interpreted.

Definition 3.5 (Approximate Interpretations of Types) *For all $n \in \mathbb{N}$, any type σ and any type environment η, i.e. any function mapping type variables to profinite subsets of D_∞, define $\mathcal{I}^n[\![\sigma]\!]\eta$ (the n-th approximation of the interpretation of σ in the environment η) as follows, by induction on n and the complexity of the type σ:*

- $\mathcal{I}^0[\![\sigma]\!]\eta = \{\bot, \top\}$;
- $\mathcal{I}^{n+1}[\![\omega]\!]\eta = D_{n+1}$;
- $\mathcal{I}^{n+1}[\![t]\!]\eta = (\eta(t))_{n+1}$;
- $\mathcal{I}^{n+1}[\![\sigma_1 \wedge \sigma_2]\!]\eta = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta \cap \mathcal{I}^{n+1}[\![\sigma_2]\!]\eta$;
- $\mathcal{I}^{n+1}[\![\sigma_1 \vee \sigma_2]\!]\eta = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta \cup \mathcal{I}^{n+1}[\![\sigma_2]\!]\eta$;
- $\mathcal{I}^{n+1}[\![\sigma_1 \to \sigma_2]\!]\eta = \mathcal{I}^{n}[\![\sigma_1]\!]\eta \to^{n+1} \mathcal{I}^{n}[\![\sigma_2]\!]\eta$;
- $\mathcal{I}^{n+1}[\![\mu t.\sigma_1]\!]\eta = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := \mathcal{I}^{n}[\![\mu t.\sigma_1]\!]\eta]$;
- $\mathcal{I}^{n+1}[\![\forall t.\sigma_1]\!]\eta = \bigcap_{A\in\mathbf{P}} \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A]$;
- $\mathcal{I}^{n+1}[\![\exists t.\sigma_1]\!]\eta = \bigcup_{A\in\mathbf{P}} \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A]$.

By a simple inductive argument we prove that each $\mathcal{I}^n[\![\sigma]\!]\eta$ is a profinite subset of D_n. We shall make frequent use in what follows of the fact that for any type σ and any $n \in \mathbb{N}$, $\mathcal{I}^n[\![\sigma]\!]\eta$ depends only on the values $(\eta(t))_n$ of the type variables occurring in σ, as can be easily seen from the definition of the approximate type interpretations.

Lemma 3.6 *For all types σ, for all $n \in \mathbb{N}$, all type environments η, all $A \in \mathbf{P}$ and all type variables t:*

(a) $\mathcal{I}^{n+1}[\![\sigma]\!]\eta[t := A_n] \subseteq \mathcal{I}^{n+1}[\![\sigma]\!]\eta[t := A_{n+1}]$;

(b) *If $d \in \mathcal{I}^{n+1}[\![\sigma]\!]\eta[t := A_{n+1}]$ and $d \notin \mathcal{I}^{n+1}[\![\sigma]\!]\eta[t := A_n]$, then $d \in A_{n+1}$.*

*Proof.*By induction on the complexity of the type σ, the result being obvious when σ is a type variable or ω. For the induction step, we consider each of the possible cases:

- $\sigma \equiv \sigma_1 \wedge \sigma_2$:

(a) $\mathcal{I}^{n+1}[\![\sigma_1 \wedge \sigma_2]\!]\eta[t := A_n] = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_n] \cap \mathcal{I}^{n+1}[\![\sigma_2]\!]\eta[t := A_n]$ which is included in $\mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}] \cap \mathcal{I}^{n+1}[\![\sigma_2]\!]\eta[t := A_{n+1}] = \mathcal{I}^{n+1}[\![\sigma_1 \wedge \sigma_2]\!]\eta[t := A_{n+1}]$ by induction hypothesis.
(b) If $d \in \mathcal{I}^{n+1}[\![\sigma_1 \wedge \sigma_2]\!]\eta[t := A_{n+1}] = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}] \cap \mathcal{I}^{n+1}[\![\sigma_2]\!]\eta[t := A_{n+1}]$ but $d \notin \mathcal{I}^{n+1}[\![\sigma_1 \wedge \sigma_2]\!]\eta[t := A_n] = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_n] \cap \mathcal{I}^{n+1}[\![\sigma_2]\!]\eta[t := A_n]$ then, for $i = 1$ or $i = 2$, $d \in \mathcal{I}^{n+1}[\![\sigma_i]\!]\eta[t := A_{n+1}]$ and $d \notin \mathcal{I}^{n+1}[\![\sigma_i]\!]\eta[t := A_n]$. Using the induction hypothesis we have that $d \in A_{n+1}$.

- $\sigma \equiv \sigma_1 \vee \sigma_2$: dual to the preceding case.
- $\sigma \equiv \sigma_1 \to \sigma_2$:

$$\begin{aligned}
&\mathcal{I}^{n+1}[\![\sigma_1 \to \sigma_2]\!]\eta[t := A_{n+1}] = \\
&\quad = \mathcal{I}^n[\![\sigma_1]\!]\eta[t := A_{n+1}] \to^{n+1} \mathcal{I}^n[\![\sigma_2]\!]\eta[t := A_{n+1}] \\
&\quad = \mathcal{I}^n[\![\sigma_1]\!]\eta[t := A_n] \to^{n+1} \mathcal{I}^n[\![\sigma_2]\!]\eta[t := A_n] \\
&\quad = \mathcal{I}^{n+1}[\![\sigma_1 \to \sigma_2]\!]\eta[t := A_n],
\end{aligned}$$

hence **(a)** holds, while **(b)** is vacuously true.

- $\sigma \equiv \mu s.\sigma_1$:

$$\begin{aligned}
&\mathcal{I}^{n+1}[\![\mu s.\sigma_1]\!]\eta[t := A_{n+1}] = \\
&\quad = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}, s := \mathcal{I}^n[\![\mu s.\sigma_1]\!]\eta[t := A_{n+1}]] \\
&\quad = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}, s := \mathcal{I}^n[\![\mu s.\sigma_1]\!]\eta[t := A_n]]
\end{aligned}$$

and

$$\begin{aligned}
&\mathcal{I}^{n+1}[\![\mu s.\sigma_1]\!]\eta[t := A_n] = \\
&\quad = \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_n, s := \mathcal{I}^n[\![\mu s.\sigma_1]\!]\eta[t := A_n]].
\end{aligned}$$

(a) and **(b)** follow from the induction hypothesis.

- $\sigma \equiv \forall s.\sigma_1$:

(a) For any $B \in \mathbf{P}$, $\mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_n, s := B] \subseteq \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}, s := B]$ by the induction hypothesis. Therefore, if $d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_n, s := B]$ for all $B \in \mathbf{P}$, it is also true that $d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}, s := B]$ for all $B \in \mathbf{P}$.
(b) Assume that $d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}, s := B]$ for all $B \in \mathbf{P}$, but there is some B' such that $d \notin \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_n, s := B']$. Then, by induction hypothesis, we have that $d \in A_{n+1}$.

- $\sigma \equiv \exists s.\sigma_1$:

(a) If $d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_n, s := B]$ for some $B \in \mathbf{P}$, then by induction hypothesis we also have that $d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}, s := B]$, so $d \in \mathcal{I}^{n+1}[\![\exists s.\sigma_1]\!]\eta[t := A_{n+1}]$.
(b) If $d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_{n+1}, s := B']$ for some $B' \in \mathbf{P}$ and $d \notin \mathcal{I}^{n+1}[\![\exists s.\sigma_1]\!]\eta[t := A_n]$ then, in particular, $d \notin \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A_n, s := B']$, so the induction hypothesis applies and yields $d \in A_{n+1}$. □

Lemma 3.7 *For all types σ, $n > 0$ and any type environment η:*

If, for all type environments η',
(i) $\mathcal{I}^n[\![\sigma]\!]\eta' \lhd_n \mathcal{I}^{n+1}[\![\sigma]\!]\eta'$ and
(ii) $\mathcal{I}^{n-1}[\![\mu t.\sigma]\!]\eta' \lhd_{n-1} \mathcal{I}^n[\![\mu t.\sigma]\!]\eta'$,
then $\mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}^{n-1}[\![\mu t.\sigma]\!]\eta] \lhd_n \mathcal{I}^{n+1}[\![\sigma]\!]\eta[t := \mathcal{I}^n[\![\mu t.\sigma]\!]\eta]$.

Proof. Let $d \in \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}^{n-1}[\![\mu t.\sigma]\!]\eta]$: observe that

$$\mathcal{I}^{n-1}[\![\mu t.\sigma]\!]\eta = (\mathcal{I}^n[\![\mu t.\sigma]\!]\eta)_{n-1}$$

according to Assumption (ii), so we can apply Lemma 3.6(**a**) in order to show that $d \in \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}^n[\![\mu t.\sigma]\!]\eta]$ and from (i) it follows that $d \in \mathcal{I}^{n+1}[\![\sigma]\!]\eta[t := \mathcal{I}^n[\![\mu t.\sigma]\!]\eta]$. If $d \in \mathcal{I}^{n+1}[\![\sigma]\!]\eta[t := \mathcal{I}^n[\![\mu t.\sigma]\!]\eta]$, then by (i) $d_n \in \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}^n[\![\mu t.\sigma]\!]\eta]$ and by Lemma 3.6(**b**) $d_n \in \mathcal{I}^n[\![\mu t.\sigma]\!]\eta = \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}^{n-1}[\![\mu t.\sigma]\!]\eta]$. □

Lemma 3.8 *For any $n \in \mathbb{N}$, all types σ and any type environment η:*

$$\mathcal{I}^n[\![\sigma]\!]\eta \lhd_n \mathcal{I}^{n+1}[\![\sigma]\!]\eta.$$

Proof. By induction on n. The basis is obvious, and the induction step is proved by induction on the complexity of the type σ.

• $\sigma \equiv t$, a type variable:
$\mathcal{I}^n[\![t]\!]\eta = (\eta(t))_n$, and the result follows from Proposition 3.3(3).

• $\sigma \equiv \omega$: it follows from Proposition 3.3(3) that $D_n \lhd_n D_{n+1}$.

• $\sigma \equiv \sigma_1 \to \sigma_2$:
In order to prove that

$$\mathcal{I}^{n-1}[\![\sigma_1]\!]\eta \to^n \mathcal{I}^{n-1}[\![\sigma_2]\!]\eta \subseteq \mathcal{I}^n[\![\sigma_1]\!]\eta \to^{n+1} \mathcal{I}^n[\![\sigma_2]\!]\eta$$

assume that $d \in \mathcal{I}^{n-1}[\![\sigma_1]\!]\eta \to^n \mathcal{I}^{n-1}[\![\sigma_2]\!]\eta$ and $a \in \mathcal{I}^n[\![\sigma_1]\!]\eta$. By induction hypothesis $a_{n-1} \in \mathcal{I}^{n-1}[\![\sigma_1]\!]\eta$, so

$$d \cdot a_{n-1} = d \cdot a \in \mathcal{I}^{n-1}[\![\sigma_2]\!]\eta \subseteq \mathcal{I}^n[\![\sigma_2]\!]\eta$$

using the induction hypothesis again, and Proposition 3.1. If $d \in \mathcal{I}^n[\![\sigma_1]\!]\eta \to^{n+1} \mathcal{I}^n[\![\sigma_2]\!]\eta$ and $a \in \mathcal{I}^{n-1}[\![\sigma_1]\!]\eta$, $a \in \mathcal{I}^n[\![\sigma_1]\!]\eta$ by induction hypothesis, so $d \cdot a \in \mathcal{I}^n[\![\sigma_2]\!]\eta$ and by induction hypothesis $(d \cdot a)_{n-1} \in \mathcal{I}^{n-1}[\![\sigma_2]\!]\eta$. The result follows by observing that, using Proposition 3.1, $(d \cdot a)_{n-1} = d_n \cdot a$, so

$$d_n \in \mathcal{I}^{n-1}[\![\sigma_1]\!]\eta \to^n \mathcal{I}^{n-1}[\![\sigma_2]\!]\eta.$$

• $\sigma \equiv \sigma_1 \wedge \sigma_2$:
$\mathcal{I}^n[\![\sigma_1 \wedge \sigma_2]\!]\eta = \mathcal{I}^n[\![\sigma_1]\!]\eta \cap \mathcal{I}^n[\![\sigma_2]\!]\eta \lhd_n \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta \cap \mathcal{I}^{n+1}[\![\sigma_2]\!]\eta = \mathcal{I}^{n+1}[\![\sigma_1 \wedge \sigma_2]\!]\eta$, using the induction hypothesis.

• $\sigma \equiv \sigma_1 \vee \sigma_2$:
$\mathcal{I}^n[\![\sigma_1 \vee \sigma_2]\!]\eta = \mathcal{I}^n[\![\sigma_1]\!]\eta \cup \mathcal{I}^n[\![\sigma_2]\!]\eta \lhd_n \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta \cup \mathcal{I}^{n+1}[\![\sigma_2]\!]\eta = \mathcal{I}^{n+1}[\![\sigma_1 \vee \sigma_2]\!]\eta$, using the induction hypothesis.

• $\sigma \equiv \mu t.\sigma_1$:
The induction hypotheses make possible the application of Lemma 3.7:

$$\begin{array}{rl} \mathcal{I}^n[\![\mu t.\sigma_1]\!]\eta = & \\ = & \mathcal{I}^n[\![\sigma_1]\!]\eta[t := \mathcal{I}^{n-1}[\![\mu t.\sigma_1]\!]\eta] \\ \lhd_n & \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := \mathcal{I}^n[\![\mu t.\sigma_1]\!]\eta] \\ = & \mathcal{I}^{n+1}[\![\mu t.\sigma_1]\!]\eta. \end{array}$$

• $\sigma \equiv \forall t.\sigma_1$:
Assume that $d \in \mathcal{I}^n[\![\forall t.\sigma_1]\!]\eta$ and $A \in \mathbf{P}$. Then, by induction hypothesis, $d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A]$ and therefore $d \in \mathcal{I}^{n+1}[\![\forall t.\sigma_1]\!]\eta$ as $A \in \mathbf{P}$ was arbitrary. Conversely, if $d \in \mathcal{I}^{n+1}[\![\forall t.\sigma_1]\!]\eta$ and $A \in \mathbf{P}$, by induction hypothesis we have that $d_n \in \mathcal{I}^n[\![\sigma_1]\!]\eta[t := A]$, and finally that $d_n \in \mathcal{I}^n[\![\forall t.\sigma_1]\!]\eta$.

• $\sigma \equiv \exists t.\sigma_1$:
Assume that $d \in \mathcal{I}^n[\![\exists t.\sigma_1]\!]\eta$. Then, for some $A \in \mathbf{P}$, $d \in \mathcal{I}^n[\![\sigma_1]\!]\eta[t := A]$ and by induction hypothesis

$d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A]$; therefore $d \in \mathcal{I}^{n+1}[\![\exists t.\sigma_1]\!]\eta$. Conversely, if $d \in \mathcal{I}^{n+1}[\![\sigma_1]\!]\eta[t := A]$ for some $A \in \mathbf{P}$, by induction hypothesis we have that $d_n \in \mathcal{I}^n[\![\sigma_1]\!]\eta[t := A]$, and finally also that $d_n \in \mathcal{I}^n[\![\exists t.\sigma_1]\!]\eta$. □

The preceding Lemma justifies the definition of the interpretation of a type by cumulating the information given by its approximate interpretations.

Definition 3.9 (Type Interpretations) *For every type σ and type environment η,*

$$\mathcal{I}[\![\sigma]\!]\eta \stackrel{\text{def}}{=} \{d \in D_\infty \mid \forall n \in \mathbb{N}.d_n \in \mathcal{I}^n[\![\sigma]\!]\eta\}.$$

Proposition 3.10 *1. $\forall n \in \mathbb{N}.\mathcal{I}^n[\![\sigma]\!]\eta \subseteq \mathcal{I}[\![\sigma]\!]\eta$.*

2. For any type σ and environment η, $\mathcal{I}[\![\sigma]\!]\eta$ is a profinite subset of D_∞.

Proof.(1) Easy, using Lemma 3.8.
(2) Let $\langle d^{(i)}\rangle_{i\in\mathbb{N}} \subseteq \mathcal{I}[\![\sigma]\!]\eta$ be an increasing chain. We have for all $n \in \mathbb{N}$ that $\langle (d^{(i)})_n\rangle_{i\in\mathbb{N}} \subseteq \mathcal{I}^n[\![\sigma]\!]\eta$, so $(\bigsqcup_i d^{(i)})_n \in \mathcal{I}^n[\![\sigma]\!]\eta$ by the continuity of the mappings $(\cdot)_n$, yielding the completeness of $\mathcal{I}[\![\sigma]\!]\eta$. If $d \in \mathcal{I}[\![\sigma]\!]\eta$ then $d_n \in \mathcal{I}^n[\![\sigma]\!]\eta \subseteq \mathcal{I}[\![\sigma]\!]\eta$ by point (1). □

This Proposition entails that, for any type σ and environment η,

$$(\mathcal{I}[\![\sigma]\!]\eta)_n = \mathcal{I}^n[\![\sigma]\!]\eta,$$

for all $n \in \mathbb{N}$.

Lemma 3.11 *For any type σ and environment η:*

$$\mathcal{I}[\![\mu t.\sigma]\!]\eta = \mathcal{I}[\![\sigma]\!]\eta[t := \mathcal{I}[\![\mu t.\sigma]\!]\eta]$$

Proof.($\subseteq$)

$$\begin{aligned}
& d \in \mathcal{I}[\![\mu t.\sigma]\!]\eta \Longrightarrow \\
& \Longrightarrow \quad \forall n \in \mathbb{N}.d_n \in \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}^{n-1}[\![\mu t.\sigma]\!]\eta] \text{ by Definition 3.5} \\
& \Longrightarrow \quad \forall n \in \mathbb{N}.d_n \in \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}^n[\![\mu t.\sigma]\!]\eta] \text{ by Lemma 3.6(a)} \\
& \Longrightarrow \quad \forall n \in \mathbb{N}.d_n \in \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}[\![\mu t.\sigma]\!]\eta] \\
& \Longrightarrow \quad d \in \mathcal{I}[\![\sigma]\!]\eta[t := \mathcal{I}[\![\mu t.\sigma]\!]\eta].
\end{aligned}$$

($\supseteq$)

$$\begin{aligned}
& d \in \mathcal{I}[\![\sigma]\!]\eta[t := \mathcal{I}[\![\mu t.\sigma]\!]\eta] \Longrightarrow \\
& \Longrightarrow \quad \forall n \in \mathbb{N}.d_n \in \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}[\![\mu t.\sigma]\!]\eta] \\
& \Longrightarrow \quad \forall n \in \mathbb{N}.d_n \in \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}^n[\![\mu t.\sigma]\!]\eta] \\
& \Longrightarrow \quad \forall n \in \mathbb{N}.d_n \in \mathcal{I}^n[\![\mu t.\sigma]\!]\eta \text{ by Lemma 3.6(b)} \\
& \Longrightarrow \quad d \in \mathcal{I}[\![\mu t.\sigma]\!]\eta.
\end{aligned}$$

□

Lemma 3.12 *For any pair of types σ, τ, any type variable t and any environment η,*

$$\mathcal{I}[\![\sigma[\tau/t]]\!]\eta = \mathcal{I}[\![\sigma]\!]\eta[t := \mathcal{I}[\![\tau]\!]\eta].$$

*Proof.*We show by induction that:

$$\forall n \in \mathbb{N}.\mathcal{I}^n[\![\sigma[\tau/t]]\!]\eta = \mathcal{I}^n[\![\sigma]\!]\eta[t := \mathcal{I}[\![\tau]\!]\eta].$$

The basis is obvious, and the induction step is proved by induction on the complexity of σ, the only interesting case being that in which $\sigma \equiv \mu s.\sigma_1$, where s is not free in τ. Then $(\mu s.\sigma_1)[\tau/t] \equiv \mu s.(\sigma_1[\tau/t]) \equiv \mu s.\tilde{\sigma_1}$, where $\tilde{\sigma_1} \equiv \sigma_1[\tau/t]$.

$$\begin{aligned} \mathcal{I}^n[\![\mu s.\tilde{\sigma_1}]\!]\eta &= \mathcal{I}^n[\![\sigma_1[\tau/t]]\!]\eta[s := \mathcal{I}^{n-1}[\![\mu s.\tilde{\sigma_1}]\!]\eta] \\ &= \mathcal{I}^n[\![\sigma_1]\!]\eta[t := \mathcal{I}[\![\tau]\!]\eta, s := \mathcal{I}^{n-1}[\![\mu s.\tilde{\sigma_1}]\!]\eta] \end{aligned}$$

by induction hypothesis and because $s \notin FV(\tau)$

$$= \mathcal{I}^n[\![\sigma_1]\!]\eta[t := \mathcal{I}[\![\tau]\!]\eta, s := \mathcal{I}^{n-1}[\![\mu s.\sigma_1]\!]\eta[t := \mathcal{I}[\![\tau]\!]\eta]]$$

by induction hypothesis on n applied to $(\mu s.\sigma_1)[\tau/t]$

$$= \mathcal{I}^n[\![\mu s.\sigma_1]\!]\eta[t := \mathcal{I}[\![\tau]\!]\eta].$$

□

Theorem 3.13 (Properties of Type Interpretations) *The following conditions are satisfied, for any type environment η and all types σ, τ:*

1. $\mathcal{I}[\![t]\!]\eta = \eta(t)$
2. $\mathcal{I}[\![\omega]\!]\eta = D_\infty$
3. $\mathcal{I}[\![\sigma \to \tau]\!]\eta = \mathcal{I}[\![\sigma]\!]\eta \to \mathcal{I}[\![\tau]\!]\eta$
4. $\mathcal{I}[\![\sigma \wedge \tau]\!]\eta = \mathcal{I}[\![\sigma]\!]\eta \cap \mathcal{I}[\![\tau]\!]\eta$
5. $\mathcal{I}[\![\sigma \vee \tau]\!]\eta = \mathcal{I}[\![\sigma]\!]\eta \cup \mathcal{I}[\![\tau]\!]\eta$
6. $\mathcal{I}[\![\mu t.\sigma]\!]\eta = \mathcal{I}[\![\sigma[\mu t.\sigma/t]]\!]\eta$
7. $\mathcal{I}[\![\forall t.\sigma]\!]\eta = \mathbf{inf}\{\mathcal{I}[\![\sigma]\!]\eta[t := A] \mid A \in \mathbf{P}\}$
8. $\mathcal{I}[\![\exists t.\sigma]\!]\eta = \mathbf{sup}\{\mathcal{I}[\![\sigma]\!]\eta[t := A] \mid A \in \mathbf{P}\}$.

Proof.(1) $d \in \mathcal{I}[\![t]\!]\eta$ iff, for all $n \in \mathbb{N}$, $d_n \in \mathcal{I}^n[\![t]\!]\eta = (\eta(t))_n$ iff $d \in \eta(t)$.
(2) Analogous to the preceding case.
(3) Assume $d \in \mathcal{I}[\![\sigma \to \tau]\!]\eta$, so for all $n > 0$

$$d_n \in \mathcal{I}^{n-1}[\![\sigma]\!]\eta \to^n \mathcal{I}^{n-1}[\![\tau]\!]\eta.$$

If $a \in \mathcal{I}[\![\sigma]\!]\eta$, then $a_{n-1} \in \mathcal{I}^{n-1}[\![\sigma]\!]\eta$ and

$$d_n \cdot a_{n-1} \in \mathcal{I}^{n-1}[\![\tau]\!]\eta \subseteq \mathcal{I}[\![\tau]\!]\eta$$

uniformly for all $n \in \mathbb{N}$. But $\mathcal{I}[\![\tau]\!]\eta$ is complete, so

$$d \cdot a = \bigsqcup_{n \in \mathbb{N}} (d_{n+1} \cdot a_n) \in \mathcal{I}[\![\tau]\!]\eta$$

showing that:

$$\mathcal{I}[\![\sigma \to \tau]\!]\eta \subseteq \mathcal{I}[\![\sigma]\!]\eta \to \mathcal{I}[\![\tau]\!]\eta.$$

Conversely, if $d \in \mathcal{I}[\![\sigma]\!]\eta \to \mathcal{I}[\![\tau]\!]\eta$ and $a \in \mathcal{I}^{n-1}[\![\sigma]\!]\eta \subseteq \mathcal{I}[\![\sigma]\!]\eta$, we have that $d \cdot a \in \mathcal{I}[\![\tau]\!]\eta$ and $(d \cdot a)_{n-1} \in \mathcal{I}^{n-1}[\![\tau]\!]\eta$, but now it is sufficient to observe that $(d \cdot a)_{n-1} = d_n \cdot a$ to conclude that

$$d_n \in \mathcal{I}^{n-1}[\![\sigma]\!]\eta \to^n \mathcal{I}^{n-1}[\![\tau]\!]\eta.$$

Then $d_n \in \mathcal{I}^n[\![\sigma \to \tau]\!]\eta$ for all $n \in \mathbb{N}$, yielding the conclusion.
(4) If $d \in \mathcal{I}[\![\sigma \wedge \tau]\!]\eta$, then for any $n \in \mathbb{N}$ $d_n \in \mathcal{I}^n[\![\sigma \wedge \tau]\!]\eta = \mathcal{I}^n[\![\sigma]\!]\eta \cap \mathcal{I}^n[\![\tau]\!]\eta \subseteq \mathcal{I}[\![\sigma]\!]\eta \cap \mathcal{I}[\![\tau]\!]\eta$. Conversely,

given $d \in \mathcal{I}[\![\sigma]\!]\eta \cap \mathcal{I}[\![\tau]\!]\eta$, for every $n \in \mathbb{N}$ $d_n \in \mathcal{I}^n[\![\sigma]\!]\eta \cap \mathcal{I}^n[\![\tau]\!]\eta = \mathcal{I}^n[\![\sigma \wedge \tau]\!]\eta$, therefore $d \in \mathcal{I}[\![\sigma \wedge \tau]\!]\eta$.
(5) If $d \in \mathcal{I}[\![\sigma \vee \tau]\!]\eta$, then for any $n \in \mathbb{N}$ $d_n \in \mathcal{I}^n[\![\sigma \vee \tau]\!]\eta = \mathcal{I}^n[\![\sigma]\!]\eta \cup \mathcal{I}^n[\![\tau]\!]\eta \subseteq \mathcal{I}[\![\sigma]\!]\eta \cup \mathcal{I}[\![\tau]\!]\eta$. Conversely, given $d \in \mathcal{I}[\![\sigma]\!]\eta \cup \mathcal{I}[\![\tau]\!]\eta$, for every $n \in \mathbb{N}$ $d_n \in \mathcal{I}^n[\![\sigma]\!]\eta$ or $d_n \in \mathcal{I}^n[\![\tau]\!]\eta$, so $d_n \in \mathcal{I}^n[\![\sigma]\!]\eta \cup \mathcal{I}^n[\![\tau]\!]\eta$ and finally $d \in \mathcal{I}[\![\sigma \vee \tau]\!]\eta$.
(6) From Lemmas 3.11 and 3.12.
(7)

$$\begin{aligned}
& d \in \mathcal{I}[\![\forall t.\sigma]\!]\eta \Leftrightarrow \\
& \Leftrightarrow \quad \forall n \in \mathbb{N}. d_n \in \mathcal{I}^n[\![\forall t.\sigma]\!]\eta = \bigcap_{A \in \mathbf{P}} \mathcal{I}^n[\![\sigma]\!]\eta[t := A] \\
& \Leftrightarrow \quad \forall n \in \mathbb{N}. \forall A \in \mathbf{P}. d_n \in \mathcal{I}^n[\![\sigma]\!]\eta[t := A] \\
& \Leftrightarrow \quad \forall n \in \mathbb{N}. \forall A \in \mathbf{P}. d_n \in \mathcal{I}[\![\sigma]\!]\eta[t := A] \\
& \Leftrightarrow \quad \forall A \in \mathbf{P}. d \in \mathcal{I}[\![\sigma]\!]\eta[t := A] \\
& \Leftrightarrow \quad d \in \bigcap_{A \in \mathbf{P}} \mathcal{I}[\![\sigma]\!]\eta[t := A].
\end{aligned}$$

(8) $(\mathbf{sup}\{\mathcal{I}[\![\sigma]\!]\eta[t := A] \mid A \in \mathbf{P}\})_n = \bigcup_{A \in \mathbf{P}} \mathcal{I}^n[\![\sigma]\!]\eta[t := A] = \mathcal{I}^n[\![\exists t.\sigma]\!]\eta$ by Proposition 3.3(5) and the remark following Proposition 3.10. The result follows immediately by observing that for any profinite subset A of D_∞, $d \in A$ if and only if $d_n \in A_n$ for every $n \in \mathbb{N}$. □

One important consequence of the above properties of type interpretations is that the soundness of the rules of our type inference system can now be shown quite naturally.

In the sequel, by a *term environment* we mean any function mapping term variables to elements of D_∞. The interpretation of λ-terms in the λ-model D_∞ is defined as usual (see Barendregt [3], Chapter 18.2), by induction on the structure of terms:

- $[\![x]\!]\rho = \rho(x)$;
- $[\![MN]\!]\rho = [\![M]\!]\rho \cdot [\![N]\!]\rho$;
- $[\![\lambda x.M]\!]\rho = \Psi(f)$, where $f(d) = [\![M]\!]\rho[x := d]$,

observing that every such an f is a continuous function from D_∞ to itself.

Definition 3.14 (Satisfaction) *Given a type environment η and a term environment ρ, the pair $\langle \eta, \rho \rangle$* satisfies *a typing statement $M : \sigma$, written $\langle \eta, \rho \rangle \models M : \sigma$, if $[\![M]\!]\rho \in \mathcal{I}[\![\sigma]\!]\eta$. Given a basis B, $\langle \eta, \rho \rangle \models B$ if $\langle \eta, \rho \rangle \models x : \tau$ for all basic typing statements $x : \tau \in B$. We write $\langle \eta, \rho \rangle : B \models M : \sigma$ when $\langle \eta, \rho \rangle \models B$ implies $\langle \eta, \rho \rangle \models M : \sigma$, and $B \models M : \sigma$ when $\langle \eta, \rho \rangle : B \models M : \sigma$ for all η and ρ.*

Theorem 3.15 (Soundness) *For any basis B, any λ-term M and any type σ:*

$$\textit{if } B \vdash M : \sigma \textit{ then } B \models M : \sigma.$$

*Proof.*By a straightforward induction on the derivation of the statement $B \vdash M : \sigma$. We only outline the proof in the case in which the last rule applied in the derivation is:

$$\frac{B, x : \sigma \vdash M : \tau \qquad B \vdash N : \exists t.\sigma}{B \vdash M[N/x] : \tau}$$

where $t \notin FV(B) \cup FV(\tau)$.
Assume that $\langle \eta, \rho \rangle \models B$, and observe that $[\![M[N/x]]\!]\rho = [\![M]\!]\rho[x := [\![N]\!]\rho]$, for all λ-terms M, N. By induction hypothesis $[\![N]\!]\rho \in \mathcal{I}[\![\exists t.\sigma]\!]\eta$, and therefore $([\![N]\!]\rho)_n \in \mathcal{I}^n[\![\exists t.\sigma]\!]\eta = \bigcup_{A \in \mathbf{P}} \mathcal{I}^n[\![\sigma]\!]\eta[t := A]$, so there exists $C \in \mathbf{P}$ such that $([\![N]\!]\rho)_n \in \mathcal{I}^n[\![\sigma]\!]\eta[t := C]$. Now, $\langle \eta[t := C], \rho[x := ([\![N]\!]\rho)_n] \rangle \models B, x : \sigma$, and this entails that $[\![M]\!]\rho[x := ([\![N]\!]\rho)_n] \in \mathcal{I}[\![\tau]\!]\eta$, because $t \notin FV(\tau)$. This argument works uniformly for all $n \in \mathbb{N}$, and we can conclude that

$$[\![M]\!]\rho[x := [\![N]\!]\rho] = \bigsqcup_{n \in \mathbb{N}} [\![M]\!]\rho[x := ([\![N]\!]\rho)_n] \in \mathcal{I}[\![\tau]\!]\eta,$$

using the continuity properties of the interpretation mapping for terms. □

Another feature of the technique we have used in building type interpretations is that recursive types satisfy equivalences which hold, typically, when recursion is interpreted by means of least fixed points. Notably, Bekic's theorem, i.e. the equation:

$$\mu s.(\sigma[\mu t.\tau/t]) == (\mu s.\sigma)[\mu t.(\tau[\mu s.\sigma/s])/t],$$

can be shown to hold in our semantics by induction on the denumerably many levels of approximation of both terms of the equation.

Appendix: Proof of the Rank Lemma

It suffices to show that, if $\mathcal{D} : B :- M : \sigma$ is a derivation ending with one occurrence of a cut of rank $r > 2$, in which all other cuts are ready, then $\mathcal{D}$ can be transformed into a derivation $\mathcal{D}' : B :- M : \sigma$ 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. This is enough provided that $\mathcal{D}'$ will be not increasing wrt $\mathcal{D}$, i.e. we will not generate cuts with higher degrees. By inspection of all the transformations we will shown, it is easily seen that the degrees of the transformed cuts do not increase, and the degrees of newly generated cuts is either the same of the old ones or 0. This is established by routine calculations: we shall actually perform it just in the first case we treate and in some other relevant ones. Note that these calculations are parametric wrt $\mathcal{F}$, hence $\mathcal{F}$ is actually an arbitrary uniform set of redex occurrences.

By Lemma 2.14 (iv) we can assume that $\mathcal{D}$ does not contain ready cuts of shape (f) or (g).

We split the proof into two parts: in the first one we lower the right rank of the considered cut, which is supposed to be > 1; in the second part we assume the right rank to be 1 and we lower the left rank supposed to be > 1. Since all transformations we do leave unchanged the left rank in part 1 an the right rank in part 2, this establishes the thesis.

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 the right rank is > 1. For the same reason cases of rules introducing type constructors to the right are impossible.
Case $\rightarrow$L:

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

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

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

Note that the new cut in the figure above is legal since $u \notin FV(B, B'')$ and B, B', B'' is a basis. Finally $u \notin FV(M)$ implies that $M[N[vP/u]/x] \equiv M[N/x][vP/u]$. To verify the non-increasing property, suppose that the cut inference in the given derivation $\mathcal{D}$ is generating an $\mathcal{F}$-redex (otherwise the thesis is trivially true); then the corresponding cut in the transformed derivation $\mathcal{D}'$ is generating an $\mathcal{F}$-redex too. We have immediately that the degrees of the occurrences of the cut-type do not change. In fact the degrees of the occurrences of σ in $B', B'', v : \xi \rightarrow \tau :- N[vP/u] : \sigma$ and in $B', u : \tau :- N : \sigma$ are clearly equal by definition.
Case χL ($\chi \in \{\mu, \forall, \exists, \wedge, \vee\}$):

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

If $y \notin FV(B)$, we transform it into

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

Here and in what follows we assume to have one or two cuts according to the value of the index i.
Otherwise $B = B'', y : \tau'$ for some B'', since $B, B', y : \tau'$ is a basis. Let w be any fresh variable; define $N' \equiv N[w/y]$, so that by 2.14 (ii) there exists a derivation of $B_i, w : \tau_i :- N' : \sigma$ similar to the given derivation $B_i, y : \tau_i :- N : \sigma$. Hence we make the following transformation

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

where $M[N'/x][y/w] \equiv M[N/x]$ since $w \notin FV(M)$.

In case $\exists$L it has to be true that $t \notin FV(B, B_i) \cup FV(\rho)$: since the original instance of $\exists$L was correct by hypothesis we know that $t \notin FV(B_i) \cup FV(\sigma)$, and clearly we can always choose $t \notin FV(B) \cup FV(\rho)$, so that the restriction is met.

Finally note that the last cut whose right premise is an axiom is ready and surely it does not create any redex, hence it has degree 0.
Case cut: in this case, by hypothesis, the upper cut is ready. We must distinguish subcases according to the shapes of Figure 2. We can assume that $x \notin FV(B, B', B'')$ and $z, y, u \notin FV(B''')$.
(a)

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

can be replaced by

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

Notice 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), (c) and (d):

$$\dfrac{B'', x : \sigma :- M : \rho \qquad \dfrac{\dfrac{B_i, y : \xi_i :- N : \sigma}{B, y : \nu :- N : \sigma}\ \chi\mathrm{L} \qquad \dfrac{B_j :- Q : \xi'_j}{B' :- Q : \nu}\ \chi\mathrm{R}}{B, B' :- N[Q/y] : \sigma}\ \mathsf{cut}}{B, B', B'' :- M[N[Q/y]/x] : \rho}\ \mathsf{cut}$$

where $\chi \in \{\mu, \forall, \exists, \wedge, \vee\}$, can be replaced by

$$\dfrac{\dfrac{\dfrac{B'', x : \sigma :- M : \rho \qquad B_i, y : \xi_i :- N : \sigma}{B_i, B'', y : \xi_i :- M[N/x] : \rho}\ \mathsf{cut}}{B, B'', y : \nu :- M[N/x] : \rho}\ \chi\mathrm{L} \qquad \dfrac{B_j :- Q : \xi'_j}{B' :- Q : \nu}\ \chi\mathrm{R}}{B, B', B'' :- M[N/x][Q/y] : \rho}\ \mathsf{cut}.$$

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

(e): we are given a figure of the shape

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

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

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

(the difference being $z : \tau$ in the conclusion of χL). In both cases we assume that $y \notin FV(B'')$ and $y \notin FV(M)$. We transform them into

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

and into

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

respectively. Notice that there is a deduction of $B_i[z/y], B' :- N[z/y] : \sigma$ similar to the given deduction of $B_i :- N : \sigma$ by Lemma 2.14 (i) and (ii). Moreover by the assumption $y \notin FV(M)$ we have $M[N[z/y]/x] \equiv M[N/x][z/y]$.

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

$$\dfrac{\dfrac{B_i :- M_i : \tau_i}{B, x : \sigma :- M : \rho}\,\mathrm{rule} \qquad B' :- N : \sigma}{B, B' :- M[N/x] : \rho}\,\mathsf{cut}.$$

In the conclusion of rule the cut type σ is either generated in that inference, or it has a father in one or both premises of rule. In the last case we shall assume that the statement $x : \sigma$ occurs in both B_1 and B_2 (if $i = 2$), being the case in which it occurs just once similar and 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 the left rank is > 1.
Case $\to$L: we are given either a figure of the shape

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

or of the shape

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

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

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

and into

$$\dfrac{\dfrac{\dfrac{B, u:\tau, x:\sigma\to\tau :- M:\rho \quad B'' :- N:\sigma\to\tau}{B, B'', u:\tau :- M[N/x]:\rho}\ \mathsf{cut} \quad \dfrac{B', x:\sigma\to\tau :- P:\sigma \quad B'' :- N:\sigma\to\tau}{B', B'' :- P[N/x]:\sigma}\ \mathsf{cut}}{B, B', B'', z:\sigma\to\tau :- M[N/x][zP[N/x]/u]:\rho}\ \to\mathrm{L} \quad B'' :- N:\sigma\to\tau}{B, B', B'' :- M[N/x][zP[N/x]/u][N/z]:\rho}\ \mathsf{cut}$$

respectively; observe that z is a fresh variable. Now this and $u \notin FV(N)$ imply $M[N/x][vP[N/x]/u] \equiv M[vP/u][N/x]$ and $M[N/x][zP[N/x]/u][N/z] \equiv M[xP/u][N/x]$.
Case $\to$R:

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

Clearly we can assume $y \notin FV(N)$ and $y \notin FV(B')$. The derivation transforms into

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

and, by the fact that $y \notin FV(N)$, $(\lambda y.M)[N/x] \equiv \lambda y.M[N/x]$.
Case χL ($\chi \in \{\mu, \forall, \exists, \wedge, \vee\}$):

$$\dfrac{\dfrac{B_i, y:\tau_i, x:\sigma :- M:\rho}{B, y:\tau', x:\sigma :- M:\rho}\ \chi\mathrm{L} \quad B' :- N:\sigma}{B, B', y:\tau' :- M[N/x]:\rho}\ \mathsf{cut}$$

If $y \notin FB(B')$, this derivation is transformed into

$$\dfrac{\dfrac{B_i, y:\tau_i, x:\sigma :- M:\rho \quad B' :- N:\sigma}{B_i, B', y:\tau_i :- M[N/x]:\rho}\ \mathsf{cut}}{B, B', y:\tau' :- M[N/x]:\rho}\ \chi\mathrm{L}$$

Otherwise we use the same renaming technique of the case χL in part 1 of this proof.
Case χR ($\chi \in \{\mu, \forall, \exists, \wedge, \vee\}$):

$$\dfrac{\dfrac{B_i, x:\sigma :- M:\tau_i}{B, x:\sigma :- M:\tau}\ \chi\mathrm{R} \quad B' :- N:\sigma}{B, B' :- M[N/x]:\tau}\ \mathsf{cut}$$

becomes

$$\dfrac{\dfrac{B_i, x:\sigma :- M:\tau_i \quad B' :- N:\sigma}{B_i, B' :- M[N/x]:\tau_i}\ \mathsf{cut}}{B, B' :- M[N/x]:\tau}\ \chi\mathrm{R}$$

Case cut: in this case by hypothesis the upper cut is ready. We must distinguish subcases according to the shapes of Figure 2. In the sequel there is no loss of generality in assuming that $y \notin FV(N)$ and $y \notin FV(B''')$.
(a):

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

is replaced by

$$\dfrac{\begin{array}{c}\mathcal{D}\\ B, B', B''', y:\nu\to\xi :- M[N/x][yR[N/x]/z]:\rho\end{array} \quad \dfrac{\dfrac{B'', u:\nu, x:\sigma :- Q:\xi \quad B''' :- N:\sigma}{B'', B''', u:\nu :- Q[N/x]:\xi}\ \mathsf{cut}}{B'', B''' :- \lambda u.Q[N/x]:\nu\to\xi}\ \to\mathrm{R}}{B, B', B'', B''' :- M[N/x][yR[N/x]/z][(\lambda u.Q[N/x])/y]:\rho}\ \mathsf{cut}$$

where the derivation $\mathcal{D}$ is given by

$$\dfrac{\dfrac{B, z:\xi, x:\sigma :- M:\rho \quad B''' :- N:\sigma}{B, B''', z:\xi :- M[N/x]:\rho}\,\mathsf{cut} \qquad \dfrac{B', x:\sigma :- R:\nu \quad B''' :- N:\sigma}{B', B''' :- R[N/x]:\nu}\,\mathsf{cut}}{B, B', B''', y:\nu \to \xi :- M[N/x][yR[N/x]/z]:\rho}\,\to\mathrm{L}$$

Notice that $M[yR/z][(\lambda u.Q)/y][N/x] \equiv M[N/x][yR[N/x]/z][(\lambda u.Q[N/x])/y]$ since we can assume that $z \notin FV(N)$.
(b), (c) and (d):

$$\dfrac{\dfrac{\dfrac{B_i, x:\sigma, y:\xi_i :- M:\rho}{B, x:\sigma, y:\nu :- M:\rho}\,\chi\mathrm{L} \qquad \dfrac{B_j, x:\sigma :- P:\xi_j}{B', x:\sigma :- P:\nu}\,\chi\mathrm{R}}{B, B', x:\sigma :- M[P/y]:\rho}\,\mathsf{cut} \qquad B'' :- N:\sigma}{B, B', B'' :- M[P/y][N/x]:\rho}\,\mathsf{cut}$$

where $\chi \in \{\mu, \forall, \exists, \wedge, \vee\}$, can be replaced by

$$\dfrac{\dfrac{B_i, y:\xi_i, x:\sigma :- M:\rho \quad B'' :- N:\sigma}{\dfrac{B_i, B'', y:\xi_i :- M[N/x]:\rho}{B, B'', y:\nu :- M[N/x]:\rho}\,\chi\mathrm{L}}\,\mathsf{cut} \qquad \dfrac{B_j, x:\sigma :- P:\xi_j \quad B'' :- N:\sigma}{\dfrac{B_j, B'' :- P[N/x]:\xi_j}{B', B'' :- P[N/x]:\nu}\,\chi\mathrm{R}}\,\mathsf{cut}}{B, B', B'' :- M[N/x][P[N/x]/y]:\rho}\,\mathsf{cut}.$$

Again we have that $M[P/y][N/x] \equiv M[N/x][P[N/x]/y]$ by the assumption about y.
(e): we are given a figure either of the shape

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

or of the shape

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

where χ is any type constructor. We transform them into

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

and into

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

respectively. We observe that the newly generated cut is ready, since by hypothesis the right rank of the current cut was 1. Moreover the degree of this new cut is the degree of the lower cut of the original figure.

□

References

[1] R. Amadio. Recursion over realizability structures. *Information and Computation*, 91(1):55–85, 1991.

[2] F. Barbanera, M. Dezani-Ciancaglini, and U. de' Liguoro. Intersection and union types: syntax and semantics. Internal report, Università di Torino, 1992.

[3] H. P. Barendregt. *The Lambda Calculus. Its Syntax and Semantics.* North-Holland, 1984.

[4] C. Böhm. The CUCH as a formal and description language. In T.B. Steel Jr., editor, *Formal Language Description for Computer Programming*, pages 179–197. North-Holland, 1966.

[5] C. Böhm and A. Berarducci. Automatic synthesis of typed λ-programs on term algebras. *Theoretical Computer Science*, 39:135–154, 1985.

[6] P. Canning, W. Cook, W. Hill, W. Olthoff, and J.C. Mitchell. F-bounded quantification for object-oriented programming. In *Proceedings Fourth International Conference on Functional Programming Languages and Computer Architecture*, pages 273–280. IEEE, September 1989.

[7] L. Cardelli. Typeful programming. Research Report 45, DEC SRC, May 24, 1989.

[8] L. Cardelli and P. Wegner. On understanding types, data abstraction and polymorphism. *ACM Computing Surveys*, 17:471–522, 1985.

[9] F. Cardone. Recursive types for Fun. *Theoretical Computer Science*, 83:29–56, 1991.

[10] F. Cardone and M. Coppo. Type inference with recursive types. Syntax and Semantics. *Information and Computation*, 92(1):48–80, 1991.

[11] M. Coppo. A completeness theorem for recursively defined types. In W. Brauer, editor, *Proc. 12th International Colloquium on Automata, Languages and Programming, LNCS 194*, pages 120–129. Springer, 1985.

[12] M. Coppo and M. Dezani-Ciancaglini. An extension of basic functionality theory for lambda-calculus. *Notre Dame Journal of Formal Logic*, 21(4):685–693, 1980.

[13] M. Coppo and G.L. Ferrari. Type inference, abstract interpretation and strictness analysis. Internal report, Università di Torino, 1992.

[14] T. Coquand and G. Huet. The calculus of constructions. *Information and Computation*, 76:95–120, 1988.

[15] T. Freeman and F. Pfenning. Refinement types for ML. In *Proceedings of the SIGPLAN '91 Symposium on Language Design and Implementation, Toronto, Ontario*, pages 268–277. ACM Press, June 1991.

[16] D.E. Knuth. Examples of formal semantics. In E. Engeler, editor, *Symposium on Semantics and Algorithmic Languages, LNM 188*, pages 212–235. Springer, 1970.

[17] D. MacQueen, G.D. Plotkin, and R. Sethi. An ideal model for recursive polymorphic types. *Information and Control*, 71((1/2)):95–130, 1986.

[18] N.P. Mendler. Recursive types and type constraints in second order lambda calculus. In *Proceedings Second Symposium on Logic in Computer Science*, pages 30–36. IEEE, 1987.

[19] J.C. Mitchell and G.D. Plotkin. Abstract types have existential type. *ACM Transactions on Programming Languages and Systems*, 10:470–502, 1988.

[20] F. Pfenning. Intersection types for a logical framework. POP Report 92-006, School of Computer Science, Carnegie-Mellon University, December 1988.

[21] B. Pierce. Preliminary investigation of a calculus with intersection and union types. Internal report, Carnegie-Mellon University, 1990.

[22] G. Pottinger. A type assignment for the strongly normalizable λ-terms. In J.P. Seldin and R.J. Hindley, editors, *To H.B. Curry: Essays in Combinatory Logic, Lambda Calculus and Formalism*, pages 561–577. Academic Press, 1980.

[23] D. Prawitz. *Natural Deduction.* Almqvist & Wiksell, 1965.

[24] J.C. Reynolds. Preliminary design of the programming language FORSYTHE. Technical Report CMU-CS-88-159, Carnegie-Mellon University, 1988.

[25] J.C. Reynolds. Syntactic control of interference. Part II. In G. Ausiello et al. editor, *International Colloquium on Automata, Languages and Programming, LNCS 372*, pages 704–722. Springer, 1989.

[26] D. Scott. Continuous lattices. In F.W. Lawvere, editor, *Toposes, Algebraic Geometry and Logic, LNM 274*, pages 97–136. Springer, 1972.

[27] M. Szabo. *The Collected Papers of Gerhard Gentzen.* North-Holland, 1969.