Developing Realisability Semantics for Intersection Types and Expansion Variables

Fairouz Kamareddine, Karim Nour, Vincent Rahli and J. B. Wells

http://www.macs.hw.ac.uk/ultra

Abstract. *Expansion* was invented at the end of the 1970s for calculating *principal typings* for λ-terms in type systems with intersection types. *Expansion variables* (E-variables) were invented at the end of the 1990s to simplify and help mechanize expansion. Recently, E-variables have been further simplified and generalized to also allow calculating other type operators than just intersection. There has been much work on denotational semantics for type systems with intersection types, but none whatsoever before now on type systems with E-variables. Building a semantics for E-variables turns out to be challenging. To simplify the problem, we consider only E-variables, and not the corresponding operation of expansion. We develop a realizability semantics where each use of an E-variable in a type corresponds to an independent level at which evaluation occurs in the λ-term that is assigned the type. In the λ-term being evaluated, the only interaction possible between portions at different levels is that higher level portions can be passed around but never applied to lower level portions. We apply this semantics to two intersection type systems. We show these systems are sound, that completeness does not hold for the first system, and completeness holds for the second system when only one E-variable is allowed (although it can be used many times and nested). As far as we know, this is the first study of a denotational semantics of intersection type systems with E-variables (using realizability or any other approach) and of the difficulties involved.

1 Introduction

Intersection types were developed in the late 1970s to type λ-terms that are untypable with simple types; they do this by providing a kind of finitary type polymorphism where the usage of types is listed rather than quantified over. They have been useful in reasoning about the semantics of the λ-calculus, and have been investigated for use in static program analysis. Coppo, Dezani, and Venneri [4] introduced the operation of *expansion* on *typings* (pairs of a type environment and a result type) for calculating the possible typings of a term when using intersection types. Expansion is a crucial part of a procedure for calculating *principal typings* and thus helps support compositional type inference. As a simple example, the λ-term $M = (\lambda x.x(\lambda y.yz))$ can be assigned the typing $\Phi_1 = \langle (z : a) \vdash (((a \to b) \to b) \to c) \to c \rangle$, which happens to be its principal typing. The term M can also be assigned the typing $\Phi_2 = \langle(z : a_1 \sqcap a_2) \vdash (((a_1 \to b_1) \to b_1) \sqcap ((a_2 \to b_2) \to b_2) \to c) \to c\rangle$, and an expansion operation can obtain Φ_2 from Φ_1. Because the early definitions of expansion were complicated, E-variables were introduced in order to make the calculations easier to mechanize and reason about. For example, in System E [2], the typing Φ_1 from above is replaced by $\Phi_3 = \langle(z : ea) \vdash (e((a \to b) \to b) \to c)\rangle$, which differs from Φ_1 by the insertion of the E-variable e at two places, and Φ_2 can be obtained from Φ_3 by substituting for e the *expansion term* $E = (a := a_1, b := b_1) \sqcap (a := a_2, b := b_2)$. Carlier and Wells [3] have surveyed the history of expansion and also E-variables.

Various kinds of denotational semantics have helped in reasoning about the properties of entire type systems and also of specific typed terms. E-variables pose serious challenges for semantics. Most commonly, a type's semantics is given as a set of closed λ-terms with behavior related to the specification given by the type. In many kinds of semantics, the meaning of a type T is calculated by an expression $[T]_\nu$ that takes two parameters, the type T and also a valuation ν that assigns to type variables the same kind of meanings that are assigned to types. To extend this idea to types with E-variables, we would need to devise some space of possible meanings for E-variables. Given that a type $e\,T$ can be turned by expansion into a new type $S_1(T) \sqcap S_2(T)$, where S_1 and S_2 are arbitrary substitutions (in fact, they can be arbitrary further expansions), and that this can introduce an unbounded number of new variables (both E-variables and regular type variables), the situation is complicated.

Because it is unclear how to devise a space of meanings for expansions and E-variables, we instead develop a space of meanings for types that is hierarchical in the sense of having many levels. When assigning meanings to types, we make each use of E-variables simply change levels. We specifically avoid trying to give a semantics to the operation of expansion, and instead treat only the E-variables. Although this idea is not perfect, it seems to go quite far in giving an intuition for E-variables, namely that each E-variable acts as a kind of capsule that isolates parts of the λ-term being analyzed by the typing. Parts of the λ-term that are typed inside the uses of the E-variable-introduction typing rule for a particular E-variable e can interact with each other, and parts outside e can only pass the parts inside e around. The E-variable e of course also shows up in the types, and isolates the portions of the types contributed by the portions of the term inside the relevant uses of E-variable-introduction.

The semantic approach we use is realisability semantics. Atomic types are interpreted as sets of λ-terms that are *saturated*, meaning that they are closed under β-expansion (i.e., β-reduction in reverse). Arrow and intersection types are interpreted naturally by function spaces and set intersection. Realisability allows showing *soundness* in the sense that the meaning of a type T contains all closed λ-terms that can be assigned T as their result type. This has been shown useful in previous work for characterizing the behavior of typed λ-terms [12]. One also wants to show the converse of soundness which is called *completeness*, i.e., that every closed λ-term in the meaning of T can be assigned T as its result type.

Hindley [8–10] was the first to study this notion of completeness for a simple type system. Then, he generalised his completeness proof for an intersection type system [7]. Using his completeness theorem for the realisability semantics based on the sets of λ-terms saturated by $\beta\eta$-equivalence, Hindley has shown that simple types are uniquely realised by the λ-terms which are typable by these types. However, this result does not hold for his intersection type system and he established completeness using saturation by $\beta\eta$-equivalence. In this paper, our completeness depends instead only on the weaker requirement of β-equivalence. Also, we have managed to make simpler proofs that avoid needing η-reduction, confluence (a.k.a. Church/Rosser), or strong normalisation (SN) (although we do establish both confluence and SN for both β and $\beta\eta$).

Other work on realizability we consulted includes that by Labib-Sami [13], Farkh and Nour [6], and Coquand [5], although none of this work deals with intersection types or E-variables. Related work on realisability that deals with intersection types includes that by Kamareddine and Nour [11], which gives a realisability semantics with soundness and completeness for an intersection type system. This system is different from the ones in this paper, because it allows the universal type ω. We do not know how to build a semantics that supports both ω and E-variables. The method of levels we use in this paper would need to assign ω to every level, which is impossible. Further work will be needed on this point.

In this paper we study the λI-calculus typed with two representative intersection type systems. The restriction to λI (where in every $(\lambda x.M)$, x must be free in M) is motivated by not knowing how to support the ω type. For one of these systems, we show that subject reduction (SR) and hence completeness do not hold whereas for the second system, SR holds and completeness will hold if at most one single E-variable is used. This is the first paper, as far as we know, that studies denotational semantics of intersection type systems with E-variables (using realizability or any other approach) and outlines the difficulties of doing so.

Section 2 introduces the $\lambda I^{\mathbb{N}}$-calculus, which is the λI-calculus with each variable marked by a natural number *degree*. Section 3 introduces the syntax and terminology for types, and also the realisability semantics. Section 4 introduces our two intersection type systems with E-variables where in one, the syntax of types is not restricted but in the other it is restricted but then extended with a subtyping relation. We show that SR and completeness do not hold for the first system, and that SR holds for the second system. In section 5 we show the soundness of the realisability semantics for both systems and give a number of examples. Section 6 shows completeness does not hold for the second system if more than one expansion variable is used, but does hold for a restriction of this system to one single E-variable. This is an important study in the semantics of intersection type systems with expansion variables since a unique expansion variable can be used many times and can occur nested. Section 7 concludes. In the appendices we establish confluence and strong normalisation results as well as results related to the usual unindexed λI-calculus.

2 The pure $\boldsymbol{\lambda I}^{\mathbb{N}}$-calculus

In this section we give an indexed version of the λI-calculus where indices (which range over the set of natural numbers $\mathbb{N}$) help categorise the good terms (where the degree (or level) of a function is always smaller than that of its arguments). This amounts to having the full λI-calculus at each level (index) and creating new λI-terms through a mixing recipe.

We assume that if a metavariable v ranges over a set $\mathcal{S}$ then v_i for $i \geq 0$ and $v', v'', etc.$ also range over $\mathcal{S}$.

Definition 1. *1. Let $\mathcal{V}$ be a denumerably infinite set of variables. The set of terms $\mathcal{M}$, the set of good terms $\mathbb{M} \subset \mathcal{M}$, the set of free variables $FV(M)$ of $M \in \mathcal{M}$, the degree $d(M)$ of a term M and the joinability $M \diamond N$ of terms M and N (which ensures that in any term, variables have unique degrees) are defined by simultaneous induction:*

– *If $x \in \mathcal{V}$, $n \in \mathbb{N}$, then $x^n \in \mathcal{M} \cap \mathbb{M}$, $FV(x^n) = \{x^n\}$, and $d(x^n) = n$.*
– *If $M, N \in \mathcal{M}$ such that $M \diamond N$ (see below), then*
  - *$(M\,N) \in \mathcal{M}$, $FV((M\,N)) = FV(M) \cup FV(N)$ and $d((M\;N)) = \min(d(M), d(N))$ (where* min *is the minimum)*
  - *If $M \in \mathbb{M}$, $N \in \mathbb{M}$ and $d(M) \leq d(N)$ then $(M\,N) \in \mathbb{M}$.*

– *If $M \in \mathcal{M}$ and $x^n \in FV(M)$, then*
  - *$(\lambda x^n.M) \in \mathcal{M}$, $FV((\lambda x^n.M)) = FV(M) \setminus \{x^n\}$, and $d(\lambda x^n.M_1) = d(M_1)$.*
  - *If $M \in \mathbb{M}$ then $\lambda x^n.M \in \mathbb{M}$.*

2. *Let $M, N \in \mathcal{M}$. We say that M and N are joinable and write $M \diamond N$ iff $\forall x \in \mathcal{V}$, if $x^m \in FV(M)$ and $x^n \in FV(N)$, then $m = n$.*
   If $\mathcal{X} \subseteq \mathcal{M}$ such that $\forall M, N \in \mathcal{X}, M \diamond N$, we write, $\diamond\mathcal{X}$.
   If $\mathcal{X} \subseteq \mathcal{M}$ and $M \in \mathcal{M}$ such that $\forall N \in \mathcal{X}, M \diamond N$, we write, $M \diamond \mathcal{X}$.
   We adopt the usual definition ([1, 12]) of subterms and the convention for parentheses and their omittance. Note that a subterm of $M \in \mathcal{M}$ (resp. $\mathbb{M}$) is also in $\mathcal{M}$ (resp. $\mathbb{M}$). We let $x, y, z,$ etc. range over $\mathcal{V}$ and $M, N, P,$ etc. range over $\mathcal{M}$ and use $=$ for syntactic equality.
3. *For each $n \in \mathbb{N}$, we let:* • $\mathcal{M}^n = \{M \in \mathcal{M} \;/\; d(M) = n\}$
   • $\mathcal{M}^{>n} = \mathcal{M}^{\geq n+1}$ • $\mathcal{M}^{\geq n} = \{M \in \mathcal{M} \;/\; d(M) \geq n\}$ • $\mathbb{M}^n = \mathbb{M} \cap \mathcal{M}^n$
4. *For $n \geq 0$, $M[(x_i^{n_i} := N_i)_{1 \leq i \leq n}]$ (or simply $M[(x_i^{n_i} := N_i)_n]$), the simultaneous substitution of N_i for all free occurrences of $x_i^{n_i}$ in M only matters when $\diamond\mathcal{X}$ where $\mathcal{X} = \{M\} \cup \{N_i / 1 \leq i \leq n\} \subseteq \mathcal{M}$. Hence we restrict substitution accordingly to incorporate the $\diamond$ condition. With $\mathcal{X}$ as above, $M[(x_i^{n_i} := N_i)_n]$ is only defined when $\diamond\mathcal{X}$. We write $M[(x_i^{n_i} := N_i)_{1 \leq i \leq 1}]$ as $M[x_1^{n_1} := N_1]$ obviously.*
5. *On $\mathcal{M}$, we take terms modulo α-conversion given by:*
   $$\lambda x^n.M = \lambda y^n.(M[x^n := y^n]) \text{ where } \forall m, y^m \notin FV(M)$$
   Moreover, we use the Barendregt convention (BC) where the names of bound variables differ from the free ones and where we rewrite terms so that not both λx^n and λx^m co-occur when $n \neq m$.
6. *A relation R on $\mathcal{M}$ is compatible iff for all $M, N, P \in \mathcal{M}$:*
   – *If MRN and $x^n \in FV(M) \cap FV(N)$, then $(\lambda x^n.M)R(\lambda x^n.N)$.*
   – *If MRN, $M \diamond P$ and $N \diamond P$, then $(MP)R(NP)$ and $(PM)R(PN)$.*
7. *The reduction relation $\rhd_\beta$ on $\mathcal{M}$ is defined as the least compatible relation closed under the rule: $(\lambda x^n.M)N \rhd_\beta M[x^n := N]$ if $d(N) = n$.*
8. *We denote by $\rhd_\beta^*$ the reflexive and transitive closure of $\rhd_\beta$. We also denote by $\simeq_\beta$ the equivalence relation induced by $\rhd_\beta^*$.*

The beta reduction is well defined on the $\lambda I^{\mathbb{N}}$-calculus. I.e., if $M \in \mathcal{M}$ and $M \rhd_\beta N$ then $N \in \mathcal{M}$. Hence, $\rhd_\beta^*$ is a also well defined relation on $\mathcal{M}$. Moreover it preserves the free variables, degrees and goodness of terms. I.e., $FV(M) = FV(N)$, $\mathrm{d}(M) = \mathrm{d}(N)$ and M is good iff N is good.

The next definition turns terms of degree n into terms of higher degrees and also, if $n > 0$, they can be turned into terms of lower degrees.

Definition 2. *1. We define $^+ : \mathcal{M} \mapsto \mathcal{M}$ and $^- : \mathcal{M}^{>0} \mapsto \mathcal{M}$ by:*
   • $(x^n)^+ = x^{n+1}$ • $(M_1\ M_2)^+ = M_1^+\ M_2^+$ • $(\lambda x^n.M)^+ = \lambda x^{n+1}.M^+$
   • $(x^n)^- = x^{n-1}$ • $(M_1\ M_2)^- = M_1^-\ M_2^-$ • $(\lambda x^n.M)^- = \lambda x^{n-1}.M^-$
2. *Let $\mathcal{X} \subseteq \mathcal{M}$. If $\forall M \in \mathcal{X}$, $d(M) > 0$, we write $d(\mathcal{X}) > 0$. We define:*
   • $\mathcal{X}^+ = \{M^+ \;/\; M \in \mathcal{X}\}$ • *If $d(\mathcal{X}) > 0$, $\mathcal{X}^- = \{M^- \;/\; M \in \mathcal{X}\}$.*

3. *We define M^{-n} by induction on $d(M) \geq n \geq 0$. If $n = 0$ then $M^{-n} = M$ and if $n \geq 0$ then $M^{-(n+1)} = (M^{-n})^{-}$.*

$^{+}$ and $^{-}$ are well behaved operations with respect to all that matters (free variables, reduction, joinability, substitution, etc.).

3 The types and their realisability semantics

In this section, we introduce the basic sets of types we use and the notions of a degree of a type and of a good type. We also introduce the realisability semantics where good types can only contain good terms.

3.1 The types

This paper studies two type systems. In the first, there are no restrictions on where the arrow occurs. In the second, arrows cannot occur to the left of intersections or expansions. The next definition gives these two basic sets of types and the notions of a degree of a type and of a good type.

Definition 3 (Types, good types, degree of a type).

1. *Assume two denumerably infinite sets $\mathcal{A}$ (atomic types) and $\mathcal{E}$ (expansion variables). Let $a, b, c, etc.$ range over $\mathcal{A}$ and e range over $\mathcal{E}$.*
2. *The sets of types $\mathcal{T}$, $\mathbb{U}$ and $\mathbb{T}$ are defined by (note that $\mathbb{T} \subseteq \mathbb{U} \subseteq \mathcal{T}$):*

$$\mathcal{T} ::= \mathcal{A} \mid \mathcal{T} \to \mathcal{T} \mid \mathcal{T} \sqcap \mathcal{T} \mid \mathcal{E}\mathcal{T}$$
$$\mathbb{U} ::= \mathbb{U} \sqcap \mathbb{U} \ \mid \ \mathcal{E}\mathbb{U} \ \mid \ \mathbb{T} \qquad \textit{where} \qquad \mathbb{T} ::= \ \mathcal{A} \ \mid \ \mathbb{U} \to \mathbb{T}$$

   T, U, V, W (resp. T, resp. U, V, W) ranges over $\mathcal{T}$ (resp. $\mathbb{T}$, resp. $\mathbb{U}$). We quotient types by taking $\sqcap$ to be commutative (i.e. $U_1 \sqcap U_2 = U_2 \sqcap U_1$), associative (i.e. $U_1 \sqcap (U_2 \sqcap U_3) = (U_1 \sqcap U_2) \sqcap U_3$), idempotent (i.e. $U \sqcap U = U$) and to satisfy $e(U_1 \sqcap U_2) = eU_1 \sqcap eU_2$.
3. *Denote $e_{i_l} \ldots e_{i_n}$ by $\boldsymbol{e}_{i(l:n)}$ and $U_n \sqcap U_{n+1} \ldots \sqcap U_m$ by $\sqcap_{i=n}^{m} U_i$ ($n \leq m$).*
4. *We define a function $d : \mathcal{T} \mapsto \mathbb{N}$ by (hence d is also defined on $\mathbb{U}$):*
   - $d(a) = 0$
   - $d(U \to T) = \min(d(U), d(T))$
   - $d(eU) = d(U) + 1$
   - $d(U \sqcap V) = \min(d(U), d(V))$.
5. *We define the good types on $\mathcal{T}$ by (this also defines good types on $\mathbb{U}$):*
   - *If $a \in \mathcal{A}$, then a is good*
   - *If U is good and $e \in \mathcal{E}$, then eU is good*
   - *If U, T are good and $d(U) \geq d(T)$, then $U \to T$ is good*
   - *If U, V are good and $d(U) = d(V)$, then $U \sqcap V$ is good*

Definition 4 (Environments).

1. *A type environment is a set $\{x_i^{n_i} : U_i / 1 \leq i \leq n$ where $n \geq 0$ and $\forall 1 \leq i, j \leq n$, if $i \neq j$ then $x_i^{n_i} \neq x_j^{n_j}\}$. We denote such environment (call it Γ) by $x_1^{n_1} : U_1, x_2^{n_2} : U_2, \ldots, x_n^{n_n} : U_n$ or simply by $(x_i^{n_i} : U_i)_n$ and*

define $dom(\Gamma) = \{x_i^{n_i} / 1 \leq i \leq n\}$. *We use* Γ, Δ *to range over environments and write* $()$ *for the empty environment.*
Of course on $\mathcal{T}$*, type environments take variables in* $\mathcal{V}$ *to* $\mathcal{T}$*. On* $\mathbb{U}$*, they take variables in* $\mathcal{V}$ *to* $\mathbb{U}$*. We say that:*
- Γ *is good iff , for every* $1 \leq i \leq k$*,* U_i *is good.*
- $d(\Gamma) > 0$ *iff for every* $1 \leq i \leq k$*,* $d(U_i) > 0$ *and* $n_i > 0$*.*

2. *If* $\Gamma = (x_i^{n_i} : U_i)_n$ *and* $x^m \notin dom(\Gamma)$*, then we write* $\Gamma, x^m : U$ *for the type environment* $x_1^{n_1} : U_1, \ldots, x_n^{n_n} : U_n, x^m : U$*.*
3. *Let* $\Gamma_1 = (x_i^{n_i} : U_i)_n, (y_j^{m_j} : V_j)_m$ *and* $\Gamma_2 = (x_i^{n_i} : U_i')_n, (z_k^{r_k} : W_k)_r$*. We write* $\Gamma_1 \sqcap \Gamma_2$ *for the type environment* $(x_i^{n_i} : U_i \sqcap U_i')_n, (y_j^{m_j} : V_j)_m, (z_k^{r_k} : W_k)_r$*. Note that* $dom(\Gamma_1 \sqcap \Gamma_2) = dom(\Gamma_1) \cup dom(\Gamma_2)$ *and that* $\sqcap$ *is commutative, associative and idempotent on environments.*
4. $e\Gamma = (x_i^{n_i+1} : eT_i)_n$ *where* $\Gamma = (x_i^{n_i} : T_i)_n$*. So* $e(\Gamma_1 \sqcap \Gamma_2) = e\Gamma_1 \sqcap e\Gamma_2$*.*
5. *We say that* Γ_1 *is joinable with* Γ_2 *and write* $\Gamma_1 \diamond \Gamma_2$ *iff*
$$\forall x \in \mathcal{V}, \text{ if } x^m \in dom(\Gamma_1) \text{ and } x^n \in dom(\Gamma_2), \text{ then } m = n.$$

Definition 5 (Degree decreasing of a type).

1. *If* $d(U) > 0$*, we inductively define the type* U^- *as follows:*
   - $(U_1 \sqcap U_2)^- = U_1^- \sqcap U_2^-$ • $(eU)^- = U$

   If $d(U) \geq n \geq 0$*,* U^{-n} *is defined as for* M^{-n} *in definition 2.*
2. *If* $\Gamma = (x_i^{n_i} : U_i)_k$ *and* $d(\Gamma) > 0$*, then we let* $\Gamma^- = (x_i^{n_i - 1} : U_i^-)_k$*.*
   If $d(\Gamma) \geq n \geq 0$*,* Γ^{-n} *is defined as for* M^{-n} *in definition 2.*
3. *If* U *is a type and* Γ *is a type environment such that* $d(\Gamma) > 0$ *and* $d(U) > 0$*, then we let* $(\langle \Gamma \vdash_2 U \rangle)^- = (\langle \Gamma^- \vdash_2 U^- \rangle)$*.*

3.2 The realisability semantics

Crucial to a realisability semantics are the notion of a saturated set and the interpretations and meanings of types:

Definition 6 (Saturated sets). *Let* $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{M}$*.*

1. *We use* $\mathcal{P}(\mathcal{X})$ *to denote the powerset of* $\mathcal{X}$*, i.e.* $\{\mathcal{Y} \,/\, \mathcal{Y} \subseteq \mathcal{X}\}$*.*
2. *We let* $\mathcal{X} \rightsquigarrow \mathcal{Y} = \{M \in \mathcal{M} \,/\, \forall N \in \mathcal{X}, \text{ if } M \diamond N \text{ then } M\ N \in \mathcal{Y}\}$*.*
3. $\mathcal{X}$ *is saturated iff whenever* $M \rhd_\beta^* N$ *and* $N \in \mathcal{X}$*, then* $M \in \mathcal{X}$*.*

Definition 7 (Interpretations and meaning of types). *Let* $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ *where* $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ *and* $\mathcal{V}_1, \mathcal{V}_2$ *are both denumerably infinite.*

1. *Let* $x \in \mathcal{V}_1$ *and* $n \in \mathbb{N}$*. We define* $\mathcal{N}_x^n = \{x^n\ N_1 ... N_k \in \mathbb{M} \,/\, k \geq 0\}$*.*
   It is easy to show that if $x^n\ N_1 ... N_k \in \mathcal{N}_x^n$ *then* $\forall\, 1 \leq i \leq k$*,* $d(N_i) \geq n$*.*
2. *An interpretation* $\mathcal{I} : \mathcal{A} \mapsto \mathcal{P}(\mathcal{M}^0)$ *is a function such that for all* $a \in \mathcal{A}$*:* • $\mathcal{I}(a)$ *is saturated* *and* • $\forall x \in \mathcal{V}_1$*,* $\mathcal{N}_x^0 \subseteq \mathcal{I}(a) \subseteq \mathbb{M}^0$*.*

3. *Let an interpretation* $\mathcal{I} : \mathcal{A} \mapsto \mathcal{P}(\mathcal{M}^0)$ *We extend an interpretation* $\mathcal{I}$ *to* $\mathcal{T}$ *(hence this includes* $\mathbb{U}$*) as follows:* • $\mathcal{I}(eU) = \mathcal{I}(U)^+$
   • $\mathcal{I}(U \sqcap V) = \mathcal{I}(U) \cap \mathcal{I}(V)$ • $\mathcal{I}(U \to T) = \mathcal{I}(U) \rightsquigarrow \mathcal{I}(T)$
   Since $\cap$ *is commutative, associative and idempotent, and since* $(\mathcal{X} \cap \mathcal{Y})^+ = \mathcal{X}^+ \cap \mathcal{Y}^+$*, then the function* $\mathcal{I}$ *is well defined.*
4. *Let* $U \in \mathcal{T}$ *(hence* U *can be in* $\mathbb{U}$*). We define the meaning* $[U]$ *of* U *by:* $[U] = \{M \in \mathcal{M}$ / M *is closed and* $M \in \bigcap_{\mathcal{I}\ \mathit{interpretation}} \mathcal{I}(U)\}$.

We show that type interpretations are saturated, interpretations and meanings of good types only contain good terms.

4 The typing systems $\vdash_1$ and $\vdash_2$

In this section we give $\vdash_1$ and $\vdash_2$, two intersection type systems with expansion variables. In $\vdash_1$, types are not restricted and SR fails. In $\vdash_2$, types are restricted in the sense that arrows cannot occur to the left of intersections or expansions. In order to guarantee SR for $\vdash_2$, we introduce a subtyping relation which will allow intersection type elimination.

4.1 The typing rules

In this section we introduce the typing rules and establish properties needed for the proof of completeness, such as SR or subject expansion.

Definition 8. *Let* $i \in \{1, 2\}$.

The type system $\vdash_1$ *(resp.* $\vdash_2$*) uses the set* $\mathcal{T}$ *(resp.* $\mathbb{U}$*) of definition 3. We follow [3] and write type judgments as* $M : \langle \Gamma \vdash U \rangle$ *instead of the traditional format of* $\Gamma \vdash M : U$*. The typing rules of* $\vdash_i$ *are (recall that when used for* $\vdash_1$*,* U *and* T *range over* $\mathcal{T}$*, and when used for* $\vdash_2$*,* U *ranges over* $\mathbb{U}$ *and* T *ranges over* $\mathbb{T}$*) of figure 4.1 (left). In the last clause, the binary relation* $\sqsubseteq$ *is defined on* $\mathbb{U}$ *by the rules of figure 4.1 (right).*

We let Φ *denote types in* $\mathbb{U}$*, or environments* Γ *or typings* $\langle \Gamma \vdash_2 U \rangle$*. When* $\Phi \sqsubseteq \Phi'$*,* Φ *and* Φ' *belong to the same set (*$\mathbb{U}$*/environments/typings).*

Let Γ *be an environment,* $U \in \mathcal{T}$ *and* $M \in \mathcal{M}$*. We say that:* •Γ *is* $\vdash_i$*-legal iff there are* M, U *such that* $M : \langle \Gamma \vdash_i U \rangle$*;* •$\langle \Gamma \vdash_i U \rangle$ *is good iff* Γ *and* U *are good; and* •$d(\langle \Gamma \vdash_i U \rangle) > 0$ *iff* $d(\Gamma) > 0$ *and* $d(U) > 0$.

We show that typable terms are good, have good types, and have the same degree as their types and that all legal contexts are good. We also show that no β-redexes are blocked in a typable term.

SR for β using $\vdash_1$ fails: let a, b, c be different elements of $\mathcal{A}$. Although $(\lambda x^0.x^0 x^0)(y^0 z^0) \rhd_\beta (y^0 z^0)(y^0 z^0)$ and $(\lambda x^0.x^0 x^0)(y^0 z^0) : \langle y^0 : b \to ((a \to$

$\dfrac{T \text{ good} \quad \mathrm{d}(T)=n}{x^n : \langle (x^n : T) \vdash_1 T\rangle}\ (ax)$	$\dfrac{}{\Phi \sqsubseteq \Phi}\ (ref)$
$\dfrac{T \text{ good}}{x^0 : \langle (x^0 : T) \vdash_2 T\rangle}\ (ax)$	$\dfrac{\Phi_1 \sqsubseteq \Phi_2 \quad \Phi_2 \sqsubseteq \Phi_3}{\Phi_1 \sqsubseteq \Phi_3}\ (tr)$
$\dfrac{M : \langle \Gamma, (x^n : U) \vdash_i T\rangle}{\lambda x^n.M : \langle \Gamma \vdash_i U \to T\rangle}\ (\to_I)$	$\dfrac{U_2 \text{ good} \quad \mathrm{d}(U_1)=\mathrm{d}(U_2)}{U_1 \sqcap U_2 \sqsubseteq U_1}\ (\sqcap_e)$
$\dfrac{M_1 : \langle \Gamma_1 \vdash_i U \to T\rangle \quad M_2 : \langle \Gamma_2 \vdash_i U\rangle \quad \Gamma_1 \diamond \Gamma_2}{M_1 M_2 : \langle \Gamma_1 \sqcap \Gamma_2 \vdash_i T\rangle}\ (\to_E)$	$\dfrac{U_1 \sqsubseteq V_1 \quad U_2 \sqsubseteq V_2}{U_1 \sqcap U_2 \sqsubseteq V_1 \sqcap V_2}\ (\sqcap)$
$\dfrac{M : \langle \Gamma_1 \vdash_i U_1\rangle \quad M : \langle \Gamma_2 \vdash_i U_2\rangle}{M : \langle \Gamma_1 \sqcap \Gamma_2 \vdash_i U_1 \sqcap U_2\rangle}\ (\sqcap)$	$\dfrac{U_2 \sqsubseteq U_1 \quad T_1 \sqsubseteq T_2}{U_1 \to T_1 \sqsubseteq U_2 \to T_2}\ (\to)$
$\dfrac{M : \langle \Gamma \vdash_i U\rangle}{M^+ : \langle e\Gamma \vdash_i eU\rangle}\ (exp)$	$\dfrac{U_1 \sqsubseteq U_2}{eU_1 \sqsubseteq eU_2}\ (\sqsubseteq_{exp})$
$\dfrac{M : \langle \Gamma \vdash_2 U\rangle \quad \langle \Gamma \vdash_2 U\rangle \sqsubseteq \langle \Gamma' \vdash_2 U'\rangle}{M : \langle \Gamma' \vdash_2 U'\rangle}\ (\sqsubseteq)$	$\dfrac{U_1 \sqsubseteq U_2}{\Gamma, (y^n : U_1) \sqsubseteq \Gamma, (y^n : U_2)}\ (\sqsubseteq_c)$
	$\dfrac{U_1 \sqsubseteq U_2 \quad \Gamma_2 \sqsubseteq \Gamma_1}{\langle \Gamma_1 \vdash_2 U_1\rangle \sqsubseteq \langle \Gamma_2 \vdash_2 U_2\rangle}\ (\sqsubseteq_{\langle\rangle})$

Fig. 1. Typing rules / Subtyping rules

$c) \sqcap a), z^0 : b \vdash_1 c\rangle$, it is not possible that $(y^0 z^0)(y^0 z^0) : \langle y^0 : b \to ((a \to c) \sqcap a), z^0 : b \vdash_1 c\rangle$.

Nevertheless, we show that SR and subject expansion for β using $\vdash_2$ holds. This will be used in the proof of completeness (more specifically in lemma 18 which is basic for the completeness theorem 19).

Lemma 9 (Subject reduction and expansion for β).
1. *If $M : \langle \Gamma \vdash_2 U\rangle$ and $M \rhd_\beta^* N$, then $N : \langle \Gamma \vdash_2 U\rangle$.*
2. *If $N : \langle \Gamma \vdash_2 U\rangle$ and $M \rhd_\beta^* N$ then $M : \langle \Gamma \vdash_2 U\rangle$.*

5 Soundness of the realisability semantics and examples

Since, if $\mathcal{I}$ is an interpretation and $U \sqsubseteq V$ then $\mathcal{I}(U) \subseteq \mathcal{I}(V)$, the semantics given in section 3.2 is sound with respect to $\vdash_1$ and $\vdash_2$.

Lemma 10 (Soundness of $\vdash_1/\vdash_2$). *Let $i \in \{1,2\}$, $\mathcal{I}$ be an interpretation, $M : \langle (x_j^{n_j} : U_j)_n \vdash_i U\rangle$ and $\forall 1 \le j \le n$, $N_j \in \mathcal{I}(U_j)$. If $M[(x_j^{n_j} := N_j)_n] \in \mathcal{M}$, then $M[(x_j^{n_j} := N_j)_n] \in \mathcal{I}(U)$.*

As a corollary: if $M : \langle () \vdash_i U\rangle$, then $M \in [U]$.

The next lemma puts the realisability semantics in use.

Lemma 11. *1.* $[(a \sqcap b) \to a] = \{M \in \mathbb{M}^0 \;/ M \rhd_\beta^* \lambda y^0.y^0\}$.
2. It is not possible that $\lambda y^0.y^0 : \langle () \vdash_1 (a \sqcap b) \to a\rangle$.
3. $\lambda y^0.y^0 : \langle () \vdash_2 (a \sqcap b) \to a\rangle$.

Remark 12 (Failure of completeness for $\vdash_1$). Lemma 11 shows that we can not have a completeness result (a converse of lemma 10 for closed terms) for $\vdash_1$. To type the term $\lambda y^0.y^0$ by the type $(a \sqcap b) \to a$, we need an elimination rule for $\sqcap$ which we have in $\vdash_2$. However, we will see that we have completeness for $\vdash_2$ if only one expansion variable is used.

6 Completeness of $\vdash_2$ with one expansion variable

Let $a \in \mathcal{A}$, $e_1, e_2 \in \mathcal{E}$, $e_1 \neq e_2$ and $Nat_0 = (e_1 a \to a) \to (e_2 a \to a)$. Then: 1) $\lambda f^0.f^0 \in [Nat_0]$ and 2) It is not possible that $\lambda f^0.f^0 : \langle () \vdash_2 Nat_0\rangle$.

Hence $\lambda f^0.f^0 \in [Nat_0]$ but $\lambda f^0.f^0$ is not typable by Nat_0 and we do not have completeness in the presence of more than one expansion variable. The problem comes from the fact that for the realizability semantics that we considered, we identify all expansion variables. In order to give a completeness theorem we will in what follows restrict our system to only one expansion variable. In the rest of this section, we assume that the set $\mathcal{E}$ contains only one expansion variable e_c.

The need of one single expansion variable is clear in part 2) of lemma 13 which would fail if we use more than one expansion variable. For example, if $e_1 \neq e_2$ then $e_1(e_2 a)^- = e_1 a \neq e_2 a$. This lemma is crucial for the rest of this section and hence, a single expansion variable is also crucial.

Lemma 13. *Let $U, V \in \mathbb{U}$ and $d(U) = d(V) > 0$. Then 1) $e_c U^- = U$ and 2) If $U^- = V^-$, then $U = V$.*

Next, we divide $\{y^n \;/\; y \in \mathcal{V}_2\}$ disjointly amongst types of order n.

Definition 14. *Let $U \in \mathbb{U}$. We define sets of variables $\mathbb{V}_U$ by induction on $d(U)$. If $d(U) = 0$, then: $\mathbb{V}_U$ is an infinite set of variables of degree 0; if $y^0 \in \mathbb{V}_U$, then $y \in \mathcal{V}_2$; and if $U \neq V$ and $d(U) = d(V) = 0$, then $\mathbb{V}_U \cap \mathbb{V}_V = \emptyset$. If $d(U) = n+1$, then we put $\mathbb{V}_U = \{y^{n+1} \;/\; y^n \in \mathbb{V}_{U^-}\}$.*

Our above partition of $\mathcal{V}_2$ will allow useful infinite sets which contain type environments that will play a crucial role in one particular type interpretation. These sets and environments are given in the next definition.

Definition 15. *1. Let $n \in \mathbb{N}$. We let $\mathbb{G}^n = \{(y^n : U) \;/ U \in \mathbb{U}$, $d(U) = n$ and $y^n \in \mathbb{V}_U\}$ and $\mathbb{H}^n = \bigcup_{m \geq n} \mathbb{G}^m$. Note that $\mathbb{G}^n$ and $\mathbb{H}^n$ are not type environments because they are infinite sets.*

2. *Let $n \in \mathbb{N}$, $M \in \mathcal{M}$ and $U \in \mathbb{U}$, we write $M : \langle \mathbb{H}^n \vdash_2 U \rangle$ iff there is a type environment $\Gamma \subset \mathbb{H}^n$ where $M : \langle \Gamma \vdash_2 U \rangle$*

Now, for every n, we define the set of the good terms of order n which contain some free variable x^i where $x \in \mathcal{V}_1$ and $i \geq n$.

Definition 16. *Let $n \in \mathbb{N}$ and $\mathcal{V}^n = \{M \in \mathbb{M}^n \ / \ x^i \in FV(M)$ where $x \in \mathcal{V}_1$ and $i \geq n\}$. Obviously, if $n \in \mathbb{N}$ and $x \in \mathcal{V}_1$, then $\mathcal{N}_x^n \subseteq \mathcal{V}^n$.*

Here is the crucial interpretation $\mathbb{I}$ for the proof of completeness:

Definition 17. *Let $\mathbb{I}$ be the interpretation defined by: for all type variables a, $\mathbb{I}(a) = \mathcal{V}^0 \cup \{M \in \mathcal{M}^0 \ / \ M : \langle \mathbb{H}^0 \vdash_2 a \rangle\}$.*

$\mathbb{I}$ is indeed an interpretation and the interpretation of a type of order n contains the good terms of order n which are typable in these special environments which are parts of the infinite sets of definition 15:

Lemma 18. *1. $\mathbb{I}$ is an interpretation. I.e., $\forall a \in \mathcal{A}$, $\mathbb{I}(a)$ is saturated and $\forall x \in \mathcal{V}_1$, $\mathcal{N}_x^0 \subseteq \mathbb{I}(a) \subseteq \mathbb{M}^0$.*
2. *If $U \in \mathbb{U}$ is good and $d(U) = n$, then $\mathbb{I}(U) = \mathcal{V}^n \cup \{M \in \mathbb{M}^n \ / \ M : \langle \mathbb{H}^n \vdash_2 U \rangle\}$.*

$\mathbb{I}$ is used to prove completeness (the proof is on the authors web pages).

Theorem 19 (Completeness). *Let $U \in \mathbb{U}$ be good such that $d(U) = n$.*
1. *$[U] = \{M \in \mathbb{M}^n \ / \ M : \langle () \vdash_2 U \rangle\}$.*
2. *$[U]$ is stable by reduction: i.e., if $M \in [U]$ and $M \rhd_\beta^* N$, then $N \in [U]$.*
3. *$[U]$ is stable by expansion: i.e., if $N \in [U]$ and $M \rhd_\beta^* N$, then $M \in [U]$.*

7 Conclusion and future work

In this paper, we studied the $\lambda I^{\mathbb{N}}$-calculus, an indexed version of the λI-calculus. This indexed version was typed using first a basic intersection type system with expansion variables but without an intersection elimination rule, and then using an intersection type system with expansion variables and an elimination rule.

We gave a realisability semantics for both type systems showing that the first type system is not complete in the sense that there are types whose semantic meaning is not the set of $\lambda I^{\mathbb{N}}$-terms having this type. In particular, we showed that $\lambda y^0.y^0$ is in the semantic meaning of $(a \sqcap b) \to a$ but it is not possible to give $\lambda y^0.y^0$ the type $(a \sqcap b) \to a$. The main reason for the failure of completeness in the first system is associated with the failure of the subject reduction property for this first system. Hence, we moved to the second system which we show to have the desirable properties of subject reduction and expansion and strong normalisation. However, for this second system, we show again that completeness fails if we use more than one expansion variable but that completeness succeeds if we restrict the system to one single expansion variable.

Since we show in the appendices that each of these type systems, when restricted to the normal λI-calculus represents a well known intersection type system with expansion variables, our study can be said to be the first denotational semantics study of intersection type systems with expansion variables (using realizability or any other approach) and outlines the difficulties of doing so. Although we have in this paper limited the study to the λI-calculus, future work will include extending this work to the full λ-calculus and with an ω-type rule as well.

References

1. H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland, revised edition, 1984.
2. S. Carlier, J. Polakow, J. B. Wells, A. J. Kfoury. System E: Expansion variables for flexible typing with linear and non-linear types and intersection types. In *Programming Languages & Systems, 13th European Symp. Programming*, vol. 2986 of *Lecture Notes in Computer Science*. Springer, 2004.
3. S. Carlier, J. B. Wells. Expansion: the crucial mechanism for type inference with intersection types: A survey and explanation. ITRS '04, vol. 136 (2005-07-19) of *Elec. Notes in Theoret. Comp. Sci.*
4. M. Coppo, M. Dezani-Ciancaglini, B. Venneri. Principal type schemes and λ-calculus semantics. In J. R. Hindley, J. P. Seldin, eds., *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press, 1980.
5. T. Coquand. Completeness theorems and lambda-calculus. In TLCA, vol. 3461 of *Lecture Notes in Computer Science*, Nara, Japan, 2005. Springer.
6. S. Farkh, K. Nour. Résultats de complétude pour des classes de types du système AF2. *Theoretical Informatics and Applications*, 31(6), 1998.
7. J. R. Hindley. The simple semantics for Coppo-Dezani-Sallé types. In M. Dezani-Ciancaglini, U. Montanari, eds., *International Symposium on Programming, 5th Colloquium*, vol. 137 of *LNCS*, Turin, 1982. Springer-Verlag.
8. J. R. Hindley. The completeness theorem for typing λ-terms. *Theoretical Computer Science*, 22, 1983.
9. J. R. Hindley. Curry's types are complete with respect to F-semantics too. *Theoretical Computer Science*, 22, 1983.
10. J. R. Hindley. *Basic Simple Type Theory*, vol. 42 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1997.
11. F. Kamareddine, K. Nour. A completeness result for a realisability semantics for an intersection type system. Annal of Pure and Applied Logic, 146: 180-198, 2007.
12. J. Krivine. *Lambda-Calcul : Types et Modèles*. Etudes et Recherches en Informatique. Masson, 1990.
13. R. Labib-Sami. Typer avec (ou sans) types auxilières.

A Confluence, SN, SR

All the technical details and proofs can be found on the extended version on the authors web pages.

Definition 20. *1.* $\rhd_\eta$ *on* $\mathcal{M}$ *is the least compatible relation closed under:* $\lambda x^n.Mx^n \rhd_\eta M$ *if* $x^n \notin FV(M)$ *and* $d(M) \leq n$.
2. *We define* $\rhd_{\beta\eta} = \rhd_\beta \cup \rhd_\eta$.
3. *For* $r \in \{\eta, \beta\eta\}$, *we denote by* $\rhd_r^*$ *the reflexive and transitive closure of* $\rhd_r$. *We also denote by* $\simeq_r$ *the equivalence relation induced by* $\rhd_r^*$.

Lemma 21 (No η-redexes are blocked in typable terms). *Let* $i \in \{1,2\}$ *and* $M : \langle \Gamma \vdash_i U \rangle$. *If* $\lambda x^n.Nx^n$ *is a subterm of* M, *then* $d(N) \leq n$ *and hence if* $x^n \notin FV(N)$ *then* $\lambda x^n.Nx^n \rhd_\eta N$.

Lemma 22 (Subject η-reduction fails for $\vdash_1$). *Let* a, b, c *be different elements of* $\mathcal{A}$. *We have:*
1. $\lambda x^0.y^0x^0 \rhd_\eta y^0$.
2. $\lambda x^0.y^0x^0 : \langle y^0 : (a \to b) \sqcap (a \to c) \vdash_1 a \to (b \sqcap c)\rangle$.
3. *It is not possible that*
   $y^0 : \langle y^0 : (a \to b) \sqcap (a \to c) \vdash_1 a \to (b \sqcap c)\rangle$.
   Hence, subject reduction for η *using* $\vdash_1$ *fails.*

Theorem 23 (Subject reduction for $\beta\eta$).
If $M : \langle \Gamma \vdash_2 U \rangle$ *and* $M \rhd_{\beta\eta}^* N$, *then* $N : \langle \Gamma \vdash_2 U \rangle$.

We establish confluence using the standard parallel reduction method.

Theorem 24 (Confluence of $\rhd_\beta^*/\rhd_{\beta\eta}^*$ on $\mathcal{M}$). *Let* $r \in \{\beta, \beta\eta\}$.

1. *If* $M \rhd_r^* M_1$ *and* $M \rhd_r^* M_2$, *then there is* M' *such that* $M_1 \rhd_r^* M'$ *and* $M_2 \rhd_r^* M'$.
2. $M_1 \simeq_\beta M_2$ *iff there is a term* M *such that* $M_1 \rhd_\beta^* M$ *and* $M_2 \rhd_\beta^* M$.

To show the strong normalisation of our two type systems, we will use the well-known reducibility method.

Definition 25. *1. We say that* $M \in \mathcal{M}$ *is strongly normalising if there are no infinite derivations* $M \rhd_\beta M_1 \rhd_\beta \ldots$.
2. *For every* $n \in \mathbb{N}$, *we define* $\mathbb{SN}^n = \{M \in \mathcal{M}^n \ / \ M$ *is strongly normalizing*$\}$ *and* $\mathbb{SN} = \bigcup_{i \in \mathbb{N}} \mathbb{SN}^i$. *Note that* $\mathbb{SN} = \{M \in \mathcal{M} \ / \ M$ *is strongly normalizing*$\}$ *and* $\mathbb{SN}^n = \mathbb{SN} \cap \mathcal{M}^n$.
3. $\mathcal{X} \subseteq \mathcal{M}$ *is SN-saturated iff when* $M, N, N_1, ..., N_k \in \mathbb{SN}$, $x^n \in FV(M)$, $d(N) = n$ *and* $M[x^n := N]N_1...N_k \in \mathcal{X}$, *then* $(\lambda x^n.M)NN_1...N_k \in \mathcal{X}$.

4. *Let $x \in \mathcal{V}$ and $n \in \mathbb{N}$. We define $\mathbb{SN}^n_x = \{x^n\ N_1...N_k \in \mathcal{M}\ /\ k \geq 0$ and $\forall 1 \leq i \leq k,\ N_i \in \mathbb{SN}^{m_i}$ and $m_i \geq n\}$.*
5. *An SN-interpretation $\mathcal{I} : \mathcal{A} \mapsto \mathcal{P}(\mathcal{M}^0)$ is a function such that:*
   $\bullet$ *$\mathcal{I}(a)$ is SN-saturated* $\quad \bullet\ \forall x \in \mathcal{V},\ \mathbb{SN}^0_x \cap \mathbb{M} \subseteq \mathcal{I}(a) \subseteq \mathbb{SN}^0 \cap \mathbb{M}$
6. *We extend an SN-interpretation $\mathcal{I}$ to $\mathcal{T}$ by:* $\bullet\ \mathcal{I}(eU) = \mathcal{I}(U)^+$
   $\bullet\ \mathcal{I}(U_1 \sqcap U_2) = \mathcal{I}(U_1) \cap \mathcal{I}(U_2)$ $\bullet\ \mathcal{I}(U \to T) = \mathcal{I}(U) \rightsquigarrow \mathcal{I}(T)$.

Remark 26. We can show that since we work with the $\lambda I^{\mathbb{N}}$-calculus, strong normalisation is equivalent to weak normalisation. However, since this result is not needed for this paper, we do not discuss it further.

Lemma 27. *Let $i \in \{1, 2\}$ and $M : \langle (x_i^{n_i} : U_i)_n \vdash_i U\rangle$, $\mathcal{I}$ be an SN-interpretation and $\forall 1 \leq i \leq n$, $N_i \in \mathcal{I}(U_i)$. If $M[(x_i^{n_i} := N_i)_n] \in \mathcal{M}$, then $M[(x_i^{n_i} := N_i)_n] \in \mathcal{I}(U)$.*

Theorem 28. *Let $i \in \{1, 2\}$. If $M : \langle (x_i^{n_i} : U_i)_n \vdash_i U\rangle$, then M is strongly normalizing.*

B Removing indices from $\vdash_1$

We assume familiarity with the λI-calculus (the $\lambda I^{\mathbb{N}}$-calculus without indices). We use the same syntax of types for the λI-calculus and we define λI-environments to be exactly those of the $\lambda I^{\mathbb{N}}$-calculus but where all upper indices disappear from the variables. We use the same meta-variables in the λI- and $\lambda I^{\mathbb{N}}$-calculi. If Γ_1 and Γ_2 are two λI-environments, then we define $\Gamma_1 \sqcap \Gamma_2$ as usual. Moreover, if $\Gamma = (x_i : T_i)_n$ is a λI-environment, then we define $e\Gamma = (x_i : eT_i)_n$.

Definition 29. *1. We define the very good types on $\mathcal{T}$ by:*
- *If $a \in \mathcal{A}$, then a is very good.*
- *If U, T are very good and $d(U) = d(T)$, then $U \to T$ and $U \sqcap T$ are very good.*
- *If U is very good and $e \in \mathcal{E}$, then eU is very good.*

Note that if U is very good then U is very good.

2. We define $\overset{\circ}{\vdash}$ to be the typing relation on the $\lambda I^{\mathbb{N}}$-calculus given by all the rules of $\vdash_1$ except for ax which is replaced by:

$$\frac{T \text{ very good} \qquad d(T) = n}{x^n : \langle (x^n : T) \vdash_1 T\rangle}\ ax^{\circ}$$

Definition 30. *1. Let $r \in \{s, v\}$. We define the typing system $\vdash_r$ for the λI-calculus, based on the rules $\{ax_r, \to_{ir}, \to_{er}, \sqcap_{ir}, exp_r\}$ given by:*

$$\frac{}{x : \langle (x : T) \vdash_s T\rangle}\ ax_s \qquad\qquad \frac{T \text{ very good}}{x : \langle (x : T) \vdash_v T\rangle}\ ax_v$$

$$\frac{M : \langle \Gamma, (x : T_1) \vdash_r T_2 \rangle}{\lambda x.M : \langle \Gamma \vdash_r T_1 \to T_2 \rangle} \quad \to_{i_r}$$

$$\frac{M_1 : \langle \Gamma_1 \vdash_r T_1 \to T_2 \rangle \quad M_2 : \langle \Gamma_2 \vdash_r T_1 \rangle}{M_1\ M_2 : \langle \Gamma_1 \sqcap \Gamma_2 \vdash_r T_2 \rangle} \quad \to_{e_r}$$

$$\frac{M : \langle \Gamma_1 \vdash_s T_1 \rangle \quad M : \langle \Gamma_2 \vdash_s T_2 \rangle}{M : \langle \Gamma_1 \sqcap \Gamma_2 \vdash_s T_1 \sqcap T_2 \rangle} \quad \sqcap_{is}$$

$$\frac{M : \langle \Gamma_1 \vdash_v T_1 \rangle \quad M : \langle \Gamma_2 \vdash_v T_2 \rangle \quad d(T_1) = d(T_2)}{M : \langle \Gamma_1 \sqcap \Gamma_2 \vdash_v T_1 \sqcap T_2 \rangle} \quad \sqcap_{iv}$$

$$\frac{M : \langle \Gamma \vdash_r T \rangle}{M : \langle e\Gamma \vdash_r eT \rangle} \quad exp_r$$

2. *We associate to each $\lambda I^{\mathbb{N}}$-term M a λI-term $\overline{M}$ by induction:*
   $\overline{x^n} = x \qquad \overline{M_1\ M_2} = \overline{M_1}\ \overline{M_2} \qquad \overline{\lambda x^n.M} = \lambda x.\overline{M}$
3. *If $\Gamma = (x_i^{n_i} : T_i)_k$, then we let $\overline{\Gamma} = (x_i : T_i)_k$.*

The next lemma shows that if indices are removed from a legal typing judgement, then the resulting typing judgement is legal in the λI-calculus using the corresponding intersection type system. The lemma also establishes the result in the other direction for very good types.

Lemma 31. *1. If $M : \langle \Gamma \vdash_1 T \rangle$, then $\overline{M} : \langle \overline{\Gamma} \vdash_s T \rangle$.*
2. *If $M : \langle \Gamma \vdash_v T \rangle$, then there are M', Γ' such that $\overline{M'} = M$, $\overline{\Gamma'} = \Gamma$ and $M' : \langle \Gamma' \overset{\circ}{\vdash} T \rangle$. Moreover, such M' and Γ' are unique.*

C Removing indices from $\vdash_2$

Here, we show that our results for $\vdash_2$ can be translated to the λI-calculus (i.e., where indices are removed). We use the notations of section B.

Definition 32. *1. The typing rules of the λI-calculus are given by:*

$$\frac{}{x : \langle (x : T) \vdash_2' T \rangle} \quad ax$$

$$\frac{M : \langle \Gamma, (x : U) \vdash_2' T \rangle}{\lambda x.M : \langle \Gamma \vdash_2' U \to T \rangle} \quad \to_i$$

$$\frac{M_1 : \langle \Gamma_1 \vdash_2' U \to T \rangle \qquad M_2 : \langle \Gamma_2 \vdash_2' U \rangle}{M_1 M_2 : \langle \Gamma_1 \sqcap \Gamma_2 \vdash_2' T \rangle} \quad \to_e$$

$$\frac{M : \langle \Gamma \vdash_2' U_1 \rangle \qquad M : \langle \Gamma \vdash_2' U_2 \rangle}{M : \langle \Gamma \vdash_2' U_1 \sqcap U_2 \rangle} \quad \sqcap_i$$

$$\frac{M : \langle \Gamma \vdash_2' U \rangle}{M : \langle e\Gamma \vdash_2' eU \rangle} \quad exp$$

$$\frac{M : \langle \Gamma \vdash_2' U \rangle \qquad \langle \Gamma \vdash_2' U \rangle \sqsubseteq' \langle \Gamma' \vdash_2' U' \rangle}{M : \langle \Gamma' \vdash_2' U' \rangle} \quad \sqsubseteq'$$

In the last clause, $\sqsubseteq'$ is defined by the following rules:

$$\frac{}{\Phi \sqsubseteq' \Phi} \quad ref$$

$$\frac{\Phi_1 \sqsubseteq' \Phi_2 \qquad \Phi_2 \sqsubseteq' \Phi_3}{\Phi_1 \sqsubseteq' \Phi_3} \quad tr$$

$$\frac{}{U_1 \sqcap U_2 \sqsubseteq' U_1} \quad \sqcap_e$$

$$\frac{U_1 \sqsubseteq' V_1 \quad \& \quad U_2 \sqsubseteq' V_2}{U_1 \sqcap U_2 \sqsubseteq' V_1 \sqcap V_2} \quad \sqcap$$

$$\frac{U_2 \sqsubseteq' U_1 \quad \& \quad T_1 \sqsubseteq' T_2}{U_1 \to T_1 \sqsubseteq' U_2 \to T_2} \quad \to$$

$$\frac{U_1 \sqsubseteq' U_2}{eU_1 \sqsubseteq' eU_2} \quad \sqsubseteq'_{exp}$$

$$\frac{U_1 \sqsubseteq' U_2}{\Gamma, (y : U_1) \sqsubseteq' \Gamma, (y : U_2)} \quad \sqsubseteq'_c$$

$$\frac{U_1 \sqsubseteq' U_2 \quad \& \quad \Gamma_2 \sqsubseteq' \Gamma_1}{\langle \Gamma_1 \vdash_2' U_1 \rangle \sqsubseteq' \langle \Gamma_2 \vdash_2' U_2 \rangle} \quad \sqsubseteq'_{\langle\rangle}$$

2. *We define $\overline{M}$ and $\overline{\Gamma}$ as in definition 30.*
   If $\langle \Gamma \vdash_2 U \rangle$ is a typing, then we let $\overline{\langle \Gamma \vdash_2 U \rangle} = \langle \overline{\Gamma} \vdash_2' U \rangle$.

Lemma 33. *If $M : \langle \Gamma \vdash_2 U \rangle$, then $\overline{M} : \langle \overline{\Gamma} \vdash_2' U \rangle$*