Completeness and Soundness results for $\mathcal{X}$ with Intersection and Union Types

(To appear in Fundamenta Informaticae, 2011)

Steffen van Bakel

Department of Computing, Imperial College London,
180 Queen's Gate, London SW7 2BZ, UK
svb@doc.ic.ac.uk

Abstract

With the eye on defining a type-based semantics, this paper defines intersection and union type assignment for the sequent calculus $\mathcal{X}$, a substitution-free language that enjoys the Curry-Howard correspondence with respect to the implicative fragment of Gentzen's sequent calculus for classical logic.
We investigate the minimal requirements for such a system to be complete (*i.e.* closed under redex expansion), and show that the non-logical nature of both intersection and union types disturbs the soundness (*i.e.* closed uder reduction) properties. This implies that this notion of intersection-union type assignment needs to be restricted to satisfy soundness as well, making it unsuitable to define a semantics. We will look at two (confluent) notions of reduction, called Call-by-Name and Call-by-Value, and prove soundness results for those.

keywords: Classical Logic, Sequent Calculus, intersection types, union types.

Introduction

The Intersection Type Discipline has proven to be an expressive tool for studying termination and semantics for the λ-calculus [18, 15]. Intersection type assignment is defined as an extension of the standard notion of implicative type assignment known as Curry's system [22] (see also [32]), which expresses function composition and application; the extension made consists of relaxing the requirement that a parameter for a function should have a single type, adding the type constructor $\cap$ next to $\rightarrow$. This simple extension allows for a great leap in complexity: not only can a (filter) model can be built for the λ-calculus using intersection types, also strong normalisation (termination) can be characterised via assignable types; however, this expressiveness comes at a price, since type assignment becomes undecidable. The literature on intersection types is vast; it was first defined by Coppo and Dezani-Ciancaglini in [19], and its development took place over a number of years, culminating in the paper by Barendregt, Coppo, and Dezani-Ciancaglini [16], and has been explored by many people since[1].

It is natural to ask if these results can be achieved for other calculi (reduction systems) as well; in fact, partial results were achieved in the context of *term rewriting* by Fernàndez in collaboration with the author [10], and in the context of *object orientation* by de'Liguoro and the author [9]. In this paper we will bring intersection types to the context of *sequent calculi*, by presenting a notion of intersection and union type assignment for the (untyped) calculus $\mathcal{X}$, that was first defined by Lengrand in [36], is in part based on Urban and Bierman's work [43, 44], and was later extensively studied by Lengrand, Lescanne and the author in [11]. $\mathcal{X}$ has its foundations in Gentzen's sequent calculus LK [28], in contrast to the λ-calculus [15] which is related to natural deduction (see also [28]); in $\mathcal{X}$, duality is ubiquitous, as for example

[1]For an overview of names and papers, visit http://www.macs.hw.ac.uk/~jbw/itrs/

call-by-name is dual to call-by-value (see also [45]), and as intersection will be shown to be dual to union in this paper. The advantage of using the sequent approach here is that now we can explore the duality of intersection and union fully, through which we can study and explain various anomalies of union type assignment [41, 14] and quantification [29, 39].

The type system defined here will initially be shown to be the natural one, in that intersection and union play their expected roles for *witness expansion* (also called *completeness*). However, we show that *witness reduction* (also called *soundness*, the converse of completeness) no longer holds, and will reason that this is caused by the fact that both intersection and union lack a logical foundation: although departing from a type system where there is a direct Curry-Howard correspondence (also attributed to de Bruijn) between proofs and terms, an extension of the system with intersection (or union) types no longer enjoys that property, as already observed by Hindley for the λ-calculus [31]. There M can only be assigned type $A \cap B$ if it can be assigned both A and B, *i.e.* seen as propositions, A and B must have *the same proof*, so $\cap$ does not correspond to the logical &; it is easy to see that this problem is caused by the fact that accompanying syntax for the intersection and union type constructors is missing.

The problem of loss of soundness also appears in other contexts, such as that of ML with side-effects [29, 46, 39], and that of using intersection and union types in an operational setting [23, 27]. As here, also there the cause of the problem is that the type-assignment rules are not fully logical, making the context calls (which form part of the reduction in $\mathcal{X}$) unsafe; this has, in part, already been observed in [30] in the context of Curien and Herbelin's calculus $\overline{\lambda}\mu\tilde{\mu}$ [21]. This also explains why, for ML with side-effects, quantification is no longer sound [29, 39]: also the $(\forall I)$ and $(\forall E)$ rules of ML are not logical.

The advantage of studying this problem in the context of the highly symmetric sequent calculi will be made clear: intersection and union are truly dual for these calculi, and the at the time surprising loss of soundness for the system with intersection and union types in [41, 14] becomes now natural and inevitable. Also, contrary to the case for the λ-calculus, we will show that it is not union alone that causes problems, but that the problem is much more profound: also intersection disturbs soundness. Intersection and union have been studied in the context of classical sequents in [5, 24, 30, 26, 25], and all these systems suffer from the same kind of problem with respect to reduction as we encounter here. In this paper we will improve on those results by showing counterexamples in detail, and presenting two more restrictive systems that address the problems successfully.

The origin of $\mathcal{X}$ lies within the quest for a language designed to give a Curry-Howard correspondence to LK, a logical system in which the rules only introduce connectives (but on both sides of a sequent), in contrast to natural deduction which uses introduction and elimination rules. The only way to eliminate a connective is to eliminate the whole formula in which it appears via an application of the (cut)-rule. A number of variants exist for LK; the one we will consider in this paper, and that lies at the basis of the calculus $\mathcal{X}$, is the implicative fragment of the calculus known as Kleene's G_3 [34], which can be defined by:

$$(Ax):\ \frac{}{\Gamma, A \vdash A, \Delta} \qquad (cut):\ \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}$$

$$(\Rightarrow L):\ \frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta} \qquad (\Rightarrow R):\ \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta}$$

It allows sequents of the form $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$, where $A_1, \ldots, A_n$ all have to hold (so is to be understood as $A_1 \wedge \ldots \wedge A_n$) and at least one of $B_1, \ldots, B_m$ has to hold (so is to be understood as $B_1 \vee \ldots \vee B_m$), and has implicit weakening and contraction.

Starting from different approaches in that area [21, 43, 44], the calculus $\mathcal{X}$ was introduced in

[36], and shown to be equivalent to the $\overline{\lambda}\mu\tilde{\mu}$-calculus in terms of expressivity[2]. Using this correspondence, a strong normalisation result is shown for $\overline{\lambda}\mu\tilde{\mu}$. In fact, [36] did not study any property of untyped $\mathcal{X}$, but focused only on its type aspects in connection with the sequent calculus, by considering only terms that correspond to proofs. $\mathcal{X}$ is defined as a substitution-free calculus that is well equipped to describe the behaviour of functional programming languages at a very low level of granularity (see [36, 11]), and provides an excellent general purpose machine, very well suited to encode various calculi (for details, see [11]). Amongst the calculi studied in [11], Bloo and Rose's Calculus of Explicit Substitutions $\lambda\mathbf{x}$ [17] stands out. In fact, a 'subatomic' level was reached by decomposing explicit substitutions into smaller components.

As far as the Curry-Howard isomorphism is concerned, $\mathcal{X}$ is the first calculus to achieve that in full for LK. For example, in $\overline{\lambda}\mu\tilde{\mu}$, all provable propositions can be inhabited, but not all typeable terms correspond to proofs in LK (in fact, the underlying logic of $\overline{\lambda}\mu\tilde{\mu}$ is that of one with *focus*, as in Parigot's $\lambda\mu$ [40]), and in $\lambda\mu$ not all proof contractions can be represented, since there reduction is confluent, whereas proof contraction in LK is not. Even more, the calculus $\mathcal{X}$ is actually symmetric [13]; the '*cut*', represented by $P\widehat{\alpha} \dagger \widehat{x}Q$ represents, in a sense, the parameter passing of explicit substitution of P for x in Q, but also that of calling the context Q via α in P.

Perhaps the main feature of $\mathcal{X}$ is that it constitutes a *variable* and *substitution*-free method of computation. Rather than having variables like x representing places where terms can be inserted, in $\mathcal{X}$ the symbol x represents a *socket*, to which a term can be *attached* via a *plug* α. In fact, the main computational operation in $\mathcal{X}$ is *renaming* in that reduction in $\mathcal{X}$ boils down to renaming of *sockets* and *plugs*, in addition to the restructuring of proofs. This particular feature is shared with the π-calculus [37, 38]; this was investigated further and reported on in [8], where $\mathcal{X}$-terms are mapped into processes of the asynchronous π-calculus [33], enriched by pairing [1], preserving types and reduction.

In this paper we will treat $\mathcal{X}$ as a pure, untyped calculus, and ignore its logical origin in that we define a notion of sequent-style intersection type assignment on $\mathcal{X}$. We will see that, in view of the special nature of $\mathcal{X}$ as an input-output calculus, to achieve a natural notion of type assignment that is closed under expansion, we will also need to add union types as dual to intersection types.

The system we define in this paper is a conservative extension of the Krivine's '*Système D*' Intersection Type Assignment System for the λ-calculus ([35]; see also [16, 2, 3, 4]), in that lambda terms typeable in that system translate to $\mathcal{X}$-terms, whilst preserving the type. As was the case for systems with intersection types for the λ-calculus [16, 3], in order to get a notion of type assignment that is closed under η-reduction, we would need to introduce a $\leq$-relation on types which is contra-variant in the arrow; for simplicity, this is not part of the present system. The system presented here is also a natural extension of the system considered in [11], *i.e.* the basic implicative system for Classical Logic, but extended with intersection and union types and the type constants $\top$ and $\bot$. The main result of this paper is that this notion is closed under expansion, but needs to be restricted to become closed under reduction.

In [7] similar results have been obtained for $\overline{\lambda}\mu\tilde{\mu}$; the notions of context assignment considered there are not directly compatible to the one we consider here, so the results presented here are not a consequence of those in [7].

[2]But not all proof contractions represented in $\mathcal{X}$ are represented in $\overline{\lambda}\mu\tilde{\mu}$.

Contents of this paper

This paper is constructed as follows. Section 1 presents the syntax and reduction system of the calculus $\mathcal{X}$, and the basic system of context assignment for $\mathcal{X}$; in Section 2 we will embed the λ-calculus into $\mathcal{X}$, preserving types, and present *Système* $D\top$, Krivine's Intersection System for the λ-calculus. Then, in Section 3, we will define a notion of type assignment on $\mathcal{X}$ that uses intersection and union types, and show that type assignment in *Système* $D\top$ is preserved by the translation of the λ-calculus into $\mathcal{X}$.

This is followed in Section 4 where we show that the system of Section 3 is closed under witness expansion; as for witness reduction, in Section 5 we will see that this collapses. We will partially overcome these shortcomings in Section 6 where we define a number of subsystems, restrictions of the system defined in Section 3, that are either closed under Call-By-Name or Call-By-Value reduction. Being restrictions, for these systems now witness expansion collapses. We draw our conclusions in Section 7.

This paper corrects [5], and is an extension of [6]; the system now types all normal forms, and the subject-reduction problem is caught and treated.

1 The calculus $\mathcal{X}$

In this section we will give the definition of the $\mathcal{X}$-calculus which has been proven to be a fine-grained implementation model for various well-known calculi [11], like the λ-calculus, $\lambda\mathbf{x}$, $\lambda\mu$, and $\overline{\lambda}\mu\tilde{\mu}$. As discussed in the introduction, the calculus $\mathcal{X}$ is inspired by the sequent calculus LK; the fragment of LK we will consider in this section has only implication, and no structural rules. $\mathcal{X}$ features two separate categories of 'connectors', *plugs* and *sockets*, that act as input and output channels, respectively, and is defined without any notion of substitution or application.

Definition 1.1 (SYNTAX) The terms of the $\mathcal{X}$-calculus are defined by the following syntax, where x, y range over the infinite set of *sockets*, and α, β over the infinite set of *plugs*.

$$\begin{array}{rcccccc} P,Q & ::= & \langle x\cdot\alpha\rangle & \mid & \widehat{y}P\widehat{\beta}\cdot\alpha & \mid & P\widehat{\beta}[y]\,\widehat{x}Q \;\mid\; P\widehat{\alpha}\dagger\widehat{x}Q \\ & & \textit{capsule} & & \textit{export} & & \textit{import} \qquad \textit{cut} \end{array}$$

The $\hat{\cdot}$ symbolises that the *socket* or plug underneath is bound in the term. The notion of bound and free connector is defined as usual, and we will identify terms that only differ in the names of bound connectors, as usual.

Definition 1.2 The *free sockets* and *free plugs* in a term are defined through:

$$\begin{array}{lcl@{\qquad}lcl} fs(\langle x\cdot\alpha\rangle) & = & \{x\} & fp(\langle x\cdot\alpha\rangle) & = & \{\alpha\} \\ fs(\widehat{x}P\widehat{\alpha}\cdot\beta) & = & fs(P)\backslash\{x\} & fp(\widehat{x}P\widehat{\alpha}\cdot\beta) & = & (fp(P)\backslash\{\alpha\})\cup\{\beta\} \\ fs(P\widehat{\alpha}[y]\,\widehat{x}Q) & = & fs(P)\cup\{y\}\cup(fs(Q)\backslash\{x\}) & fp(P\widehat{\alpha}[y]\,\widehat{x}Q) & = & (fp(P)\backslash\{\alpha\})\cup fp(Q) \\ fs(P\widehat{\alpha}\dagger\widehat{x}Q) & = & fs(P)\cup(fs(Q)\backslash\{x\}) & fp(P\widehat{\alpha}\dagger\widehat{x}Q) & = & (fp(P)\backslash\{\alpha\})\cup fp(Q) \end{array}$$

A socket x or plug α occurring in P which is not free is called *bound*, and we write $x \in bs(P)$ and $\alpha \in bp(P)$. We will write $x \in fs(P,Q)$ for $x \in fs(P)$ & $x \in fs(Q)$.

We adopt Barendregt's convention (called convention on variables by Barendregt, but here it will be a convention on names), in that free and bound names of terms will be different.

The calculus, defined by the reduction rules below, explains in detail how cuts are propagated through terms to be eventually evaluated at the level of *capsules*, where the renaming

takes place. Reduction is defined by specifying both the interaction between well-connected basic syntactic structures, and how to deal with propagating active nodes to points in the term where they can interact.

It is important to know when a connector is introduced, *i.e.* is connectable, *i.e.* is exposed and unique; this will play a crucial role in the reduction rules.

Definition 1.3 (INTRODUCTION) *P introduces x* : : Either $P = Q\widehat{\beta}[x]\widehat{y}R$ with $x \notin fs(Q,R)$, or $P = \langle x{\cdot}\alpha\rangle$.
P introduces α : : Either $P = \widehat{x}Q\widehat{\beta}{\cdot}\alpha$ and $\alpha \notin fp(Q)$, or $P = \langle x{\cdot}\alpha\rangle$.

The principal reduction rules are:

Definition 1.4 (LOGICAL RULES) Let α and x be introduced in, respectively, the left and right-hand side of the main cuts below.

$$\begin{array}{rrcl}
(cap): & \langle y{\cdot}\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x{\cdot}\beta\rangle & \to & \langle y{\cdot}\beta\rangle \\
(exp): & (\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x{\cdot}\gamma\rangle & \to & \widehat{y}P\widehat{\beta}{\cdot}\gamma \\
(imp): & \langle y{\cdot}\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\beta}[x]\widehat{z}R) & \to & Q\widehat{\beta}[y]\widehat{z}R \\
(exp\text{-}imp): & (\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) & \to & \begin{cases} Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta} \dagger \widehat{z}R) \\ (Q\widehat{\gamma} \dagger \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R \end{cases}
\end{array}$$

The first three logical rules above specify a renaming procedure, whereas the last rule specifies the basic computational step: it links the *export* of a function, available on the plug α, to an adjacent *import* via the *socket* x. The effect of the reduction will be that the exported function is placed in-between the two sub-terms of the *import*, acting as interface. Notice that two cuts are created in the result, that can be grouped in two ways; these alternatives do not necessarily share all normal forms (reduction is non-confluent, so normal forms are not unique). And in fact, this rule presents a *critical pair*.

In $\mathcal{X}$ there are in fact two kinds of reduction, the one above, and the one which defines how to reduce a cut when one of its sub-terms does not introduce a connector mentioned in the cut. This will involve moving the cut inwards, towards a position where the connector *is* introduced. In case both connectors are not introduced this search can start in either direction, indicated by the tilting of the dagger, via the *activation* of the cut.

Definition 1.5 (ACTIVE CUTS) The syntax is extended with two *flagged* or *active* cuts:

$$P ::= \ldots \mid P_1\widehat{\alpha} \dagger\!\!\!\!\nearrow \widehat{x}P_2 \mid P_1\widehat{\alpha} \dagger\!\!\!\!\nwarrow \widehat{x}P_2$$

Terms constructed without these flagged cuts are called *pure*.

We define two cut-activation rules.

$$\begin{array}{rrcll}
(a\nearrow): & P\widehat{\alpha} \dagger \widehat{x}Q & \to & P\widehat{\alpha} \dagger\!\!\!\!\nearrow \widehat{x}Q & \textit{if } P \textit{ does not introduce } \alpha \\
(\nwarrow a): & P\widehat{\alpha} \dagger \widehat{x}Q & \to & P\widehat{\alpha} \dagger\!\!\!\!\nwarrow \widehat{x}Q & \textit{if } Q \textit{ does not introduce } x
\end{array}$$

Notice that, by these rules, in case both α is not introduced in P and x is not introduced in Q, activation can take place in both directions, so the cut $P\widehat{\alpha} \dagger \widehat{x}Q$ forms another critical pair, which is a second source of non-confluence.

The next rules define how to move an activated dagger inwards.

Definition 1.6 (PROPAGATION RULES) Left propagation rules:

$$
\begin{array}{rrcll}
(d\not{\prime}): & \langle y\cdot\alpha\rangle\widehat{\alpha}\not{\prime}\widehat{x}P & \to & \langle y\cdot\alpha\rangle\widehat{\alpha}\dagger\widehat{x}P & \\
(cap\not{\prime}): & \langle y\cdot\beta\rangle\widehat{\alpha}\not{\prime}\widehat{x}P & \to & \langle y\cdot\beta\rangle, & \beta\neq\alpha \\
(exp\text{-}outs\not{\prime}): & (\widehat{y}Q\widehat{\beta}\cdot\alpha)\widehat{\alpha}\not{\prime}\widehat{x}P & \to & (\widehat{y}(Q\widehat{\alpha}\not{\prime}\widehat{x}P)\widehat{\beta}\cdot\gamma)\widehat{\gamma}\dagger\widehat{x}P, & \gamma\ \textit{fresh} \\
(exp\text{-}ins\not{\prime}): & (\widehat{y}Q\widehat{\beta}\cdot\gamma)\widehat{\alpha}\not{\prime}\widehat{x}P & \to & \widehat{y}(Q\widehat{\alpha}\not{\prime}\widehat{x}P)\widehat{\beta}\cdot\gamma, & \gamma\neq\alpha \\
(imp\not{\prime}): & (Q\widehat{\beta}[z]\widehat{y}R)\widehat{\alpha}\not{\prime}\widehat{x}P & \to & (Q\widehat{\alpha}\not{\prime}\widehat{x}P)\widehat{\beta}\,[z]\,\widehat{y}(R\widehat{\alpha}\not{\prime}\widehat{x}P) & \\
(cut\not{\prime}): & (Q\widehat{\beta}\dagger\widehat{y}R)\widehat{\alpha}\not{\prime}\widehat{x}P & \to & (Q\widehat{\alpha}\not{\prime}\widehat{x}P)\widehat{\beta}\dagger\widehat{y}(R\widehat{\alpha}\not{\prime}\widehat{x}P) &
\end{array}
$$

Right propagation rules:

$$
\begin{array}{rlcll}
(\not{\prime}d): & P\widehat{\alpha}\not{\prime}\widehat{x}\langle x\cdot\beta\rangle & \to & P\widehat{\alpha}\dagger\widehat{x}\langle x\cdot\beta\rangle & \\
(\not{\prime}cap): & P\widehat{\alpha}\not{\prime}\widehat{x}\langle y\cdot\beta\rangle & \to & \langle y\cdot\beta\rangle, & y\neq x \\
(\not{\prime}exp): & P\widehat{\alpha}\not{\prime}\widehat{x}(\widehat{y}Q\widehat{\beta}\cdot\gamma) & \to & \widehat{y}(P\widehat{\alpha}\not{\prime}\widehat{x}Q)\widehat{\beta}\cdot\gamma & \\
(\not{\prime}imp\text{-}outs): & P\widehat{\alpha}\not{\prime}\widehat{x}(Q\widehat{\beta}[x]\widehat{y}R) & \to & P\widehat{\alpha}\dagger\widehat{z}((P\widehat{\alpha}\not{\prime}\widehat{x}Q)\widehat{\beta}\,[z]\,\widehat{y}(P\widehat{\alpha}\not{\prime}\widehat{x}R)), & z\ \textit{fresh} \\
(\not{\prime}imp\text{-}ins): & P\widehat{\alpha}\not{\prime}\widehat{x}(Q\widehat{\beta}[z]\widehat{y}R) & \to & (P\widehat{\alpha}\not{\prime}\widehat{x}Q)\widehat{\beta}\,[z]\,\widehat{y}(P\widehat{\alpha}\not{\prime}\widehat{x}R), & z\neq x \\
(\not{\prime}cut): & P\widehat{\alpha}\not{\prime}\widehat{x}(Q\widehat{\beta}\dagger\widehat{y}R) & \to & (P\widehat{\alpha}\not{\prime}\widehat{x}Q)\widehat{\beta}\dagger\widehat{y}(P\widehat{\alpha}\not{\prime}\widehat{x}R) &
\end{array}
$$

We write $P\to Q$ for the (reflexive, transitive, compatible) reduction relation generated by the logical, propagation and activation rules.

It is possible to define reduction on terms in $\mathcal{X}$ without using activation, allowing *cuts* to propagate over *cuts* at will; Urban and Bierman introduced the activated cuts to obtain a more controlled (limited) reduction, making a proof of strong normalisation possible [44].

Summarising, reduction brings all *cuts* down to *logical cuts* where both connectors are single and introduced, or *elimination cuts* that are cutting towards a *capsule* that does not contain the relevant connector, as in $P\widehat{\alpha}\not{\prime}\widehat{x}\langle z\cdot\beta\rangle$ or $\langle z\cdot\beta\rangle\widehat{\alpha}\not{\prime}\widehat{x}P$; performing the elimination cuts, via $(\not{\prime}cap)$ or $(cap\not{\prime})$, will then remove the term P.

In [12], two sub-reduction systems were introduced which explicitly favour one kind of activation whenever a critical pair occurs:

Definition 1.7 (CALL BY NAME AND CALL BY VALUE) We define Call By Name (CBN) and Call By Value (CBV) reduction by:

- If a cut can be activated in two ways, CBV only allows to activate it via $(a\not{\prime})$; we write $P\to_{\mathrm{V}} Q$ in that case. This is obtained by replacing rule $(\not{\prime}a)$ with:

$$(\not{\prime}a_{\mathrm{V}}):\ P\widehat{\alpha}\dagger\widehat{x}Q\ \to_{\mathrm{V}}\ P\widehat{\alpha}\not{\prime}\widehat{x}Q,\ \textit{if } P \textit{ introduces } \alpha \textit{ and } Q \textit{ does not introduce } x.$$

- CBN can only activate such a cut via $(\not{\prime}a)$ and we write $P\to_{\mathrm{N}} Q$. Likewise, we can reformulate this as the reduction system obtained by replacing rule $(a\not{\prime})$ with:

$$(a\not{\prime}_{\mathrm{N}}):\ P\widehat{\alpha}\dagger\widehat{x}Q\ \to_{\mathrm{N}}\ P\widehat{\alpha}\not{\prime}\widehat{x}Q,\ \textit{if } P \textit{ does not introduce } \alpha \textit{ and } Q \textit{ introduces } x.$$

- As in [36], we split the two variants of $(exp\text{-}imp)$ over the two notions of reduction:

$$
\begin{array}{lcl}
(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha}\dagger\widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) & \to_{\mathrm{V}} & Q\widehat{\gamma}\dagger\widehat{y}(P\widehat{\beta}\dagger\widehat{z}R) \\
(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha}\dagger\widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) & \to_{\mathrm{N}} & (Q\widehat{\gamma}\dagger\widehat{y}P)\widehat{\beta}\dagger\widehat{z}R
\end{array}
$$

This way, we obtain two notions of reduction that are clearly confluent: all reduction rules are left-linear and non-overlapping.

Notice that the full reduction relation $\rightarrow$ is not confluent. In fact, assuming α does not occur in P and x does not occur in Q, then $P\widehat{\alpha} \dagger \widehat{x}Q$ reduces to both P and Q. The first reduction takes place in CBV, the second in CBN.

$\mathcal{X}$ offers a natural presentation of the classical propositional calculus with implication, and can be seen as a variant of the G_3 system for LK [34].

We first define (simple) types and contexts.

Definition 1.8 (TYPES AND CONTEXTS [11, 12]) *i*) Let Φ be a countable (infinite) set of type-variables, ranged over by φ. The set of types is defined by the grammar

$$A, B \ ::= \ \varphi \mid (A \rightarrow B)$$

ii) A *context of sockets* Γ is a mapping from *sockets* to types, denoted as a finite set of *statements* $x{:}A$, such that the *subject* of the statements (x) are distinct. We write Γ_1, Γ_2 to mean the union of Γ_1 and Γ_2, provided Γ_1 and Γ_2 are compatible (*i.e.* identical on the intersection of their domains: if Γ_1 contains $x{:}A_1$ and Γ_2 contains $x{:}A_2$ then $A_1 = A_2$), and write $\Gamma, x{:}A$ for $\Gamma, \{x{:}A\}$.

iii) Contexts of *plugs* Δ are defined in a similar way.

When building witnesses for proofs, propositions receive names; in the style of $\lambda\mu$ and $\overline{\lambda}\mu\tilde{\mu}$, those that appear in the left part of a sequent receive *socket* names like x, y, z, etc, and those that appear in the right part of a sequent receive *plug* names like α, β, γ, etc. When in applying a rule a formula disappears from the sequent, the corresponding connector will get bound in the term that is constructed, and when a formula gets created, a connector will be associated to it.

Definition 1.9 (TYPING FOR $\mathcal{X}$ [11]) *i*) *Type judgements* are expressed via the ternary relation $P :\cdot \Gamma \vdash \Delta$, where Γ is a context of *sockets* and Δ is a context of *plugs*, and P is a term. We say that P is the *witness* of this judgement.

ii) *Context assignment for* $\mathcal{X}$ is defined by the following rules:

$$(cap): \ \frac{}{\langle y{\cdot}\alpha\rangle :\cdot \Gamma, y{:}A \vdash \alpha{:}A, \Delta} \qquad (imp): \ \frac{P :\cdot \Gamma \vdash \alpha{:}A, \Delta \quad Q :\cdot \Gamma, x{:}B \vdash \Delta}{P\widehat{\alpha}\,[y]\,\widehat{x}Q :\cdot \Gamma, y{:}A \rightarrow B \vdash \Delta}$$

$$(exp): \ \frac{P :\cdot \Gamma, x{:}A \vdash \alpha{:}B, \Delta}{\widehat{x}P\widehat{\alpha}{\cdot}\beta :\cdot \Gamma \vdash \beta{:}A \rightarrow B, \Delta} \qquad (cut): \ \frac{P :\cdot \Gamma \vdash \alpha{:}A, \Delta \quad Q :\cdot \Gamma, x{:}A \vdash \Delta}{P\widehat{\alpha} \dagger \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(*)$$

$(*)$: this rule is also applicable to activated cuts.

We write $P :\cdot \Gamma \vdash \Delta$ if there exists a derivation that has this judgement in the bottom line, and write $\mathcal{D} :: P :\cdot \Gamma \vdash \Delta$ if we want to name that derivation.

The Curry-Howard property for LK is easily achieved by erasing all term information.

Notice that Γ and Δ carry the types of the free connectors in P, as unordered sets. There is no notion of type for P itself, instead the derivable statement shows how P is connectable.

Example 1.10 (A PROOF OF PEIRCE'S LAW) The following is a proof for Peirce's Law in LK:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A, B}\,(Ax)}{\vdash A \Rightarrow B, A}\,(\Rightarrow R) \qquad \dfrac{}{A \vdash A}\,(Ax)}{(A \Rightarrow B) \Rightarrow A \vdash A}\,(\Rightarrow L)}{\vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A}\,(\Rightarrow R)$$

Inhabiting this proof in $\mathcal{X}$ gives the derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle y{\cdot}\sigma\rangle :\cdot\ y{:}A \vdash \sigma{:}A,\eta{:}B}\,(cap)}{\widehat{y}\langle y{\cdot}\sigma\rangle\widehat{\eta}{\cdot}\alpha :\cdot\ \vdash \alpha{:}A{\to}B,\sigma{:}A}\,(exp) \qquad \dfrac{}{\langle w{\cdot}\sigma\rangle :\cdot\ w{:}A \vdash \sigma{:}A}\,(cap)}{(\widehat{y}\langle y{\cdot}\sigma\rangle\widehat{\eta}{\cdot}\alpha)\widehat{\alpha}\ [z]\ \widehat{w}\langle w{\cdot}\sigma\rangle :\cdot\ z{:}(A{\to}B){\to}A \vdash \sigma{:}A}\,(imp)}{\widehat{z}((\widehat{y}\langle y{\cdot}\sigma\rangle\widehat{\eta}{\cdot}\alpha)\widehat{\alpha}\ [z]\ \widehat{w}\langle w{\cdot}\sigma\rangle)\widehat{\sigma}{\cdot}\gamma :\cdot\ \vdash \gamma{:}((A{\to}B){\to}A){\to}A}\,(exp)$$

The soundness result of simple type assignment with respect to reduction is stated as usual:

Theorem 1.11 (WITNESS REDUCTION [12]) *If $P :\cdot\ \Gamma \vdash \Delta$, and $P \to Q$, then $Q :\cdot\ \Gamma \vdash \Delta$.*

2 The relation with the Lambda Calculus

The remainder of this paper will be dedicated to the definition of a notion of intersection type assignment on $\mathcal{X}$. The definition will be such that it will be a natural extension of a system with intersection types for the λ-calculus; we will start by briefly summarising the latter. We assume the reader to be familiar with the λ-calculus [15]; we just recall the definition of lambda terms and β-contraction.

Definition 2.1 (LAMBDA TERMS AND β-CONTRACTION [15]) *i*) The set Λ of *lambda terms* is defined by the syntax:

$$M,N \ ::=\ x \mid \lambda x.M \mid MN$$

ii) The reduction relation $\to_\beta$ is defined as the compatible [15] closure of the rule:

$$(\lambda x.M)N \ \to_\beta\ M[N/x]$$

iii) The relation $\twoheadrightarrow_\beta$ is the reflexive and transitive closure of $\to_\beta$, and the $=_\beta$ is the equivalence relation generated by $\twoheadrightarrow_\beta$.

We can define the direct encoding of the λ-calculus into $\mathcal{X}$:

Definition 2.2 ([11]) The interpretation[3] of lambda terms into terms of $\mathcal{X}$ via the *plug* α, $\llbracket M \rrbracket^\lambda_\alpha$, is defined by:

$$\begin{array}{rcl}
\llbracket x \rrbracket^\lambda_\alpha & = & \langle x{\cdot}\alpha\rangle \\
\llbracket \lambda x.M \rrbracket^\lambda_\alpha & = & \widehat{x}\llbracket M \rrbracket^\lambda_\beta \widehat{\beta}{\cdot}\alpha \\
\llbracket MN \rrbracket^\lambda_\alpha & = & \llbracket M \rrbracket^\lambda_\gamma \widehat{\gamma} \dagger \widehat{x}(\llbracket N \rrbracket^\lambda_\beta \widehat{\beta}\ [x]\ \widehat{y}\langle y{\cdot}\alpha\rangle),\ \textit{with } x \textit{ fresh}
\end{array}$$

We can even represent substitution explicitly (so interpret $\lambda\mathbf{x}$), by adding

$$\llbracket M\langle < x{:=}N\rangle > \rrbracket^\lambda_\alpha \ =\ \llbracket N \rrbracket^\lambda_\gamma \widehat{\gamma} \nwarrow \widehat{x}\llbracket M \rrbracket^\lambda_\alpha,\ \ \gamma \textit{ fresh}$$

Notice that every sub-term of $\llbracket M \rrbracket^\lambda_\alpha$ has exactly one free *plug*, which corresponds to the name of hole of the present context in which M appears, *i.e.* its continuation.

As shown in [11], the notion of Curry type assignment for the λ-calculus, $\Gamma \vdash_\lambda M{:}A$, is strongly linked to the one defined for $\mathcal{X}$.

[3]This encoding corresponds to Gentzen's encoding of natural deduction into LK.

Definition 2.3 (CURRY TYPE ASSIGNMENT FOR λ-CALCULUS) The type assignment rules for the Curry type assignment system for the λ-calculus are:

$$(Ax) : \frac{}{\Gamma, x{:}A \vdash_\lambda x{:}A} \qquad (\to I) : \frac{\Gamma, x{:}A \vdash_\lambda M{:}B}{\Gamma \vdash_\lambda \lambda x.M{:}A\to B} \qquad (\to E) : \frac{\Gamma \vdash_\lambda M{:}A\to B \quad \Gamma \vdash_\lambda N{:}A}{\Gamma \vdash_\lambda MN{:}B}$$

In [11], the following relation is shown between the λ-calculus and $\mathcal{X}$:

Theorem 2.4 ([11]) *i) If $M \to_\beta N$, then $\llbracket M \rrbracket^\lambda_\alpha \to \llbracket N \rrbracket^\lambda_\alpha$.*
ii) If $\Gamma \vdash_\lambda M{:}A$, then $\llbracket M \rrbracket^\lambda_\alpha : \cdot\, \Gamma \vdash \alpha{:}A$.

It is worthwhile to notice that the image of Λ in to $\mathcal{X}$ is not extensional:

$$\begin{array}{rcl} \llbracket \lambda x.yx \rrbracket^\lambda_\alpha \triangleq \widehat{x}\llbracket yx \rrbracket^\lambda_\beta \widehat{\beta}\cdot\alpha \triangleq & \widehat{x}(\llbracket y \rrbracket^\lambda_\gamma \widehat{\gamma} \dagger \widehat{z}(\llbracket x \rrbracket^\lambda_\delta \widehat{\delta}\,[z]\,\widehat{w}\langle w\cdot\beta\rangle))\,\widehat{\beta}\cdot\alpha & \triangleq \\ & \widehat{x}(\langle y\cdot\gamma\rangle \widehat{\gamma} \dagger \widehat{z}(\langle x\cdot\delta\rangle \widehat{\delta}\,[z]\,\widehat{w}\langle w\cdot\beta\rangle))\,\widehat{\beta}\cdot\alpha & \to \\ & \widehat{x}(\langle x\cdot\delta\rangle \widehat{\delta}\,[y]\,\widehat{w}\langle w\cdot\beta\rangle)\,\widehat{\beta}\cdot\alpha & \neq \langle y\cdot\alpha\rangle = \llbracket y \rrbracket^\lambda_\alpha \end{array}$$

This means that, below, it is not necessary to get our notion of type assignment to be closed under extensionality as well.

The notion of intersection type assignment for $\mathcal{X}$ as defined in the next section is a conservative extension of the Intersection Type Assignment System of [16], in that we can translate lambda terms typeable in that system to $\mathcal{X}$ terms whilst preserving types.

The main concept covered by adding intersection types is that terms (and variables) are allowed to have any number of types, including zero; that a term has *both* types A and B is expressed by the type $A \cap B$, and that it has zero types by the type constant $\top$[4], which is the universal type, *i.e.* a type for all terms. There are many different notions of intersection type assignment in existence (see also [16, 35, 2, 3]), that in the context of the λ-calculus more or less coincide; the most important difference is normally the language of types (full BCD [16, 35] or strict [2, 3]), and the availability of a contra-variant $\leq$-relation [16, 3] or not [35, 2]. As we will see in Example 5.1, the natural system with intersection and union types for $\mathcal{X}$ has intersection on the right of the arrow, so is based on BCD types.

The type assignment system presented here is based on Krivine's *Sytème D* [35], which in itself is a restriction of the BCD-system defined by Barendregt, Coppo and Dezani-Ciancaglini in [16]. The BCD-system treats the two type constructors '$\to$' and '$\cap$' the same, allowing, in particular, intersection to occur at the right of arrow types. It also introduced a contra-variant partial order relation '$\leq$' on types, adds the type assignment rule $(\leq)$, and introduced a more general form of the rules concerning intersection. The strict system of [2] essentially allows intersection only on the left of the arrow type constructor (which corresponds to the approach of [20]), and only considers a $\leq$ relation induced by the associativity and commutativity of intersection; the system of [3] generalises this by making $\leq$ contra-variant over arrow types. Krivine's system does not have the contra-variant $\leq$-relation, and also omits the type constant $\top$.

The system we will use here sits in-between Krivine's and the BCD-system: we will use $\top$, but not allow a contra-variant $\leq$-relation; this system is known as *Système D*$\top$[5].

Definition 2.5 (INTERSECTION TYPES, STATEMENTS, AND CONTEXTS) *i)* $\mathcal{T}$, the set of *intersection types*, ranged over by $A, B, \ldots$, is defined through:

$$A, B \ ::= \ \varphi \mid \top \mid (A\to B) \mid (A \cap B)\ .$$

[4]In [16], Greek characters are used to represent types, and ω is used for $\top$; here we use Greek characters for *plugs*.

[5]Better known as *Système D*ω.

$\top$ is pronounced "*top*". In the notation of types, as usual, right-most outer-most brackets will be omitted, and $\cap$ binds more strongly than $\rightarrow$.

ii) A *statement* is an expression of the form $M : A$, with $M \in \Lambda$, and $A \in \mathcal{T}$. M is the *subject* and A the *predicate* of $M : A$.

iii) A *context* Γ is a partial mapping from term variables to intersection types, and we write $x{:}A \in \Gamma$ if $\Gamma x = A$, *i.e.* if A is the type stored for x in Γ. We will write $x \notin \Gamma$ if Γ is not defined on x, and $\Gamma \backslash x$ when we remove x from the domain of Γ.

iv) We write $\Gamma \cap x{:}A$ for the context defined by:

$$\begin{array}{rcll} \Gamma \cap x{:}A & = & \Gamma \cup \{x{:}A\}, & \text{if } x \notin \Gamma \\ & = & \Gamma \backslash x \cup \{x{:}A \cap B\}, & \text{if } x{:}B \in \Gamma \end{array}$$

We will often write $\Gamma, x{:}A$ for $\Gamma \cap x{:}A$ when $x \notin \Gamma$.

We will consider a pre-order (*i.e.* reflexive and transitive relation) on types which takes into account the idem-potence, commutativity and associativity of the intersection type constructor, and defines $\top$ to be the maximal element.

Definition 2.6 (Relations on types) *i*) On $\mathcal{T}$, the type inclusion relation $\leq$ is defined as the smallest pre-order such that:

$$A \leq \top \qquad A \cap B \leq A \qquad A \cap B \leq B \qquad C \leq A \;\&\; C \leq B \Rightarrow C \leq A \cap B$$

ii) The relation $\sim$ is defined by:

$$\begin{array}{rcl} A \leq B \leq A & \Rightarrow & A \sim B \\ A \sim C \;\&\; B \sim D & \Rightarrow & A \rightarrow B \sim C \rightarrow D. \end{array}$$

$\mathcal{T}$ can be considered modulo $\sim$; then $\leq$ becomes a partial order.

We need to point out that the $\leq$ relation as defined in [16] is different. It also contains the cases

$$\begin{array}{rcl} (A \rightarrow B) \cap (A \rightarrow C) & \leq & A \rightarrow (B \cap C) \\ C \leq A \;\&\; B \leq D & \Rightarrow & A \rightarrow B \leq C \rightarrow D \\ \top & \leq & \top \rightarrow \top \end{array}$$

These were mainly added to obtain a system closed under η-reduction (see also [3]), which is not an issue in this paper. Also, we could add these cases to Definition 3.7, but would then have to add a type assignment rule dealing explicitly with $\leq$.

It is easy to show that both $(A \cap B) \cap C \sim B \cap (A \cap C)$ and $A \cap B \sim B \cap A$, so the type constructor $\cap$ is associative and commutative, and we will write $A \cap B \cap C$ rather than $(A \cap B) \cap C$. We will write $\underline{n}$ for the set $\{1, \ldots, n\}$ with $n \geq 0$, and often write $\cap_{\underline{n}} A_i$ for $A_1 \cap \cdots \cap A_n$, and consider $\top$ to be the empty intersection: $\top = \cap_{\underline{0}} A_i$.

Definition 2.7 (*Système D*) *Intersection type assignment* and *derivations* are defined by the following natural deduction system.

$$(Ax) : \frac{}{\Gamma, x{:}A \vdash_D x : A} \quad (\cap I) : \frac{\Gamma \vdash_D M{:}A_j \quad (\forall j \in \underline{n})}{\Gamma \vdash_D M{:}\cap_{\underline{n}} A_i}\,(n \geq 0) \quad (\cap E) : \frac{\Gamma \vdash_D M{:}\cap_{\underline{n}} A_i}{\Gamma \vdash_D M{:}A_j}\,(j \in \underline{n})$$

$$(\rightarrow I) : \frac{\Gamma, x{:}A \vdash_D M{:}B}{\Gamma \vdash_D \lambda x.M{:}A \rightarrow B} \quad (\rightarrow E) : \frac{\Gamma \vdash_D M{:}A \rightarrow B \quad \Gamma \vdash_D N{:}A}{\Gamma \vdash_D MN{:}B}$$

We will write $\Gamma \vdash_D M{:}A$ for statements that are derived using these rules.

Again, notice that the BCD-system contains also the rule $(\leq)$,

$$(\leq):\ \frac{\Gamma \vdash_D M : A}{\Gamma \vdash_D M : B}\ (A \leq B)$$

which supersedes rule $(\cap E)$, and is added to be able to express contra-variance of the (original) $\leq$-relation over arrow types. The system as set up here does not need this rule.

Notice that the above notion of type assignment is not closed under η-reduction: take

$$\dfrac{\dfrac{\dfrac{}{y{:}A{\to}B \vdash_D y : A{\to}B} \qquad \dfrac{\dfrac{}{x{:}A{\cap}C \vdash_D x : A{\cap}C}}{x{:}A{\cap}C \vdash_D x : A}\ (\cap E)}{y{:}A{\to}B, x{:}A{\cap}C \vdash_D yx : B}\ (\to E)}{y{:}A{\to}B \vdash_D \lambda x.yx : (A{\cap}C){\to}B}\ (\to I)$$

Now $\lambda x.yx \to_\eta y$, but we cannot derive $y{:}A{\to}B \vdash_D y : (A{\cap}C){\to}B$.

It is easy to check that the type assignment system is closed under $=_\beta$. First, that the system is closed under subject reduction can be illustrated by the following 'Cut and Paste' proof:

Example 2.8 Suppose first that $\Gamma \vdash_D (\lambda x.M)N : A$ is derived by $(\to E)$, then there exists B such that $\Gamma \vdash_D \lambda x.M : B{\to}A$ and $\Gamma \vdash_D N : B$. Then (without loss of generality) $(\to I)$ is the last step performed for the first result, and there are sub-derivations for $\Gamma, x{:}B \vdash_D M : A$ and $\Gamma \vdash_D N : B$.

$$\dfrac{\dfrac{\boxed{\phantom{\Gamma, x{:}B \vdash_D M : A}}}{\Gamma, x{:}B \vdash_D M : A}\ (\to I) \qquad \boxed{}}{\Gamma \vdash_D (\lambda x.M)N : A}\ (\to E)$$

Then a derivation for $\Gamma \vdash_D M[N/x] : A$ can be obtained by replacing in the derivation for $\Gamma, x{:}B \vdash_D M : A$, the sub-derivation $\Gamma, x{:}B \vdash_D x : B$ by the derivation for $\Gamma \vdash_D N : B$.

$$\begin{array}{c} \boxed{} \\ \Gamma \vdash_D N : B \\ \boxed{} \\ \Gamma \vdash_D M[N/x] : A \end{array}$$

In fact, this reasoning is applicable to many notions of type assignment, and does not depend at all on the presence of either $\cap$ or $\top$.

The second problem to solve in a proof for closure under β-equality is that of β-expansion:

Example 2.9 In order to show 'if $\Gamma \vdash_D M[N/x] : A$, then $\Gamma \vdash_D (\lambda x.M)N : A$', we consider two cases:

- Assume that the term-variable x occurs in M and so the term N is a sub-term of $M[N/x]$; then N is typed in the derivation for $\mathcal{D} :: \Gamma \vdash_D M[N/x] : A$, probably with several different types $B_1, \ldots, B_n$.

$$\begin{array}{c} \boxed{} \quad \cdots \quad \boxed{} \\ \boxed{} \\ \Gamma \vdash_D M[N/x] : A \end{array}$$

A derivation for $\Gamma, x{:}\cap_{\underline{n}} B_i \vdash_D M : A$ can be obtained by replacing all derivations for $\Gamma \vdash_D N : B_j$ in $\mathcal{D}$ by the derivation for

$$\dfrac{\dfrac{}{\Gamma, x{:}\cap_{\underline{n}} B_i \vdash_D x : \cap_{\underline{n}} B_i}\,(Ax)}{\Gamma, x{:}\cap_{\underline{n}} B_i \vdash_D x : B_j}\,(\cap E)$$

Then, using $(\to I)$, we can derive $\Gamma \vdash_D \lambda x.M : (\cap_{\underline{n}} B_i) \to A$, and, using $(\cap I)$ on the collection of removed sub-derivations, derive $\Gamma \vdash_D N : \cap_{\underline{n}} B_i$. Then, using $(\to E)$, the redex can be typed:

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{}{\Gamma, x{:}\cap_{\underline{n}} B_i \vdash_D x : \cap_{\underline{n}} B_i}\,(Ax)}{\Gamma, x{:}\cap_{\underline{n}} B_i \vdash_D x : B_j}\,(\cap E) \quad \cdots \quad \dfrac{\dfrac{}{\Gamma, x{:}\cap_{\underline{n}} B_i \vdash_D x : \cap_{\underline{n}} B_i}\,(Ax)}{\Gamma, x{:}\cap_{\underline{n}} B_i \vdash_D x : B_j}\,(\cap E)\\ \Gamma, x{:}\cap_{\underline{n}} B_i \vdash_D M : A\end{array}}{\Gamma \vdash_D \lambda x.M : (\cap_{\underline{n}} B_i) \to A}\,(\to I) \quad \dfrac{\begin{array}{c}\Gamma \vdash_D N : B_j \quad (\forall j \in \underline{n})\end{array}}{\Gamma \vdash_D N : \cap_{\underline{n}} B_i}\,(\cap I)}{\Gamma \vdash_D (\lambda x.M)N : A}\,(\to E)$$

- When the term-variable x does not occur in M, the term N is a not a sub-term of $M[N/x]$ and $\Gamma \vdash_D M[N/x] : A$ stands for $\Gamma \vdash_D M : A$. In this case, the type $\top$ is used: since x does not occur in M, by weakening $x{:}\top$ can be assumed to appear in Γ, and applying rule $(\to I)$ gives $\Gamma \vdash_D \lambda x.M : \top \to A$. By $(\cap I)$, $\Gamma \vdash_D N : \top$, so, using $(\to E)$, the redex can be typed.

$$\dfrac{\dfrac{\begin{array}{c}\Gamma, x{:}\top \vdash_D M : A\end{array}}{\Gamma \vdash_D \lambda x.M : \top \to A}\,(\to I) \quad \dfrac{}{\Gamma \vdash_D N : \top}\,(\cap I)}{\Gamma \vdash_D (\lambda x.M)N : A}\,(\to E)$$

3 Intersection and Union Context Assignment for $\mathcal{X}$

The notion of intersection context assignment on $\mathcal{X}$ that we will present in this section is a natural extension of the system of Definition 1.9, *i.e.* the basic implicative system for Classical Logic, but extended with intersection and union types and the type constants $\top$ and $\bot$. The proofs we present here are corrections of those presented in [5]; it mistakenly claimed a witness reduction result.

Note that, as with intersection type assignment for the λ-calculus, the relation between the system we presented above and (classical) logic is lost. In fact, although the rules for dealing with intersection and union types are similar to the rules for the logical connectors '*and*' (&) and '*or*' ($\vee$), intersection and union have *no* logical content [31]: read as logical rules, $(\cap R)$ and $(\cup L)$ are only applicable when all sub-formulae are shown by proofs with *identical structure* - they have the same witness.

The system presented in this section is set up to be concise, and closed under reduction and expansion. As discussed in the previous section, the intersection type constructor is needed for expansion in the λ-calculus; in $\mathcal{X}$ we need intersection during expansion when, for example, considering the reduction step ($\nwarrow$*imp-outs*):

$$P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\gamma}\,[x]\,\widehat{y}R) \to P\widehat{\alpha} \dagger \widehat{v}((P\widehat{\alpha} \nwarrow \widehat{x}Q)\widehat{\gamma}\,[v]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x}R))$$

Assume we have a derivation for the right-hand side, constructed like this (notice that, by

Barendregt's convention, we can assume that y and γ are not free in P):

$$\dfrac{\boxed{\mathcal{D}_1}\quad \dfrac{\boxed{\mathcal{D}_2}\quad \boxed{\mathcal{D}_3}}{P\widehat{\alpha}\nwarrow\widehat{x}Q :\cdot \Gamma\vdash\gamma{:}A,\Delta}(cut)\quad \dfrac{\boxed{\mathcal{D}_4}\quad \boxed{\mathcal{D}_5}}{P\widehat{\alpha}\nwarrow\widehat{x}R :\cdot \Gamma,y{:}B,x{:}D\vdash\Delta}(cut)}{P :\cdot \Gamma\vdash\alpha{:}A\to B,\Delta\quad (P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\gamma}\,[v]\,\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}R) :\cdot \Gamma,v{:}A\to B\vdash\Delta}(imp)}{P\widehat{\alpha}\dagger\widehat{v}((P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\gamma}\,[v]\,\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}R)) :\cdot \Gamma\vdash\Delta}(cut)$$

Then, in order to construct a derivation for the left-hand side, we need to combine the derivations for P; as above for the λ-calculus, since this combines a number of types on the output α, the natural tool to use is intersection:

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}}{P :\cdot \Gamma\vdash\alpha{:}A\to B,\Delta}\quad \dfrac{\boxed{\mathcal{D}_2}}{P :\cdot \Gamma\vdash\alpha{:}C,\Delta}\quad \dfrac{\boxed{\mathcal{D}_4}}{P :\cdot \Gamma\vdash\alpha{:}D,\Delta}}{P :\cdot \Gamma\vdash\alpha{:}(A\to B)\cap C\cap D,\Delta}(\cap R)\quad \begin{array}{c}\dfrac{\boxed{\mathcal{D}_3}\quad \boxed{\mathcal{D}_5}}{Q\widehat{\gamma}\,[x]\,\widehat{y}R :\cdot \Gamma,x{:}(A\to B)\cap C\cap D\vdash\Delta}(imp)\\ \vdots\end{array}}{P\widehat{\alpha}\nwarrow\widehat{x}(Q\widehat{\gamma}\,[x]\,\widehat{y}R) :\cdot \Gamma\vdash\Delta}(cut)$$

Notice that we also naturally construct an intersection type for the *socket* x.

Similarly, we need union, for example, when dealing with rule (*exp-outs*$\nearrow$):

$$(\widehat{y}P\widehat{\epsilon}\cdot\alpha)\widehat{\alpha}\nearrow\widehat{x}Q \to (\widehat{y}(P\widehat{\alpha}\nearrow\widehat{x}Q)\widehat{\epsilon}\cdot\gamma)\widehat{\gamma}\dagger\widehat{x}Q$$

Assume we have a derivation for the right-hand side, shaped like

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad \boxed{\mathcal{D}_2}}{P :\cdot \Gamma,y{:}A\vdash\sigma{:}B,\alpha{:}C,\Delta\quad Q :\cdot \Gamma,x{:}C\vdash\Delta}(cut)}{P\widehat{\alpha}\nearrow\widehat{x}Q :\cdot \Gamma,y{:}A\vdash\sigma{:}B,\Delta}}{\widehat{y}(P\widehat{\alpha}\nearrow\widehat{x}Q)\widehat{\sigma}\cdot\gamma :\cdot \Gamma\vdash\gamma{:}A\to B,\Delta}(\to R)\quad \boxed{\mathcal{D}_3}}{(\widehat{y}(P\widehat{\alpha}\nearrow\widehat{x}Q)\widehat{\sigma}\cdot\gamma)\widehat{\gamma}\dagger\widehat{x}Q :\cdot \Gamma\vdash\Delta}(cut)$$

When trying to build a derivation for the left-hand side, we need a way to gather the derivations for Q using a new type constructor, dual to intersection; the natural choice is union:

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}}{P :\cdot \Gamma,y{:}A\vdash\sigma{:}B,\alpha{:}C,\Delta}(\to R)}{\widehat{y}P\widehat{\sigma}\cdot\alpha :\cdot \Gamma\vdash\alpha{:}C\cup(A\to B),\Delta}\quad \dfrac{\boxed{\mathcal{D}_2}\quad \boxed{\mathcal{D}_3}}{Q :\cdot \Gamma,x{:}C\vdash\Delta\quad Q :\cdot \Gamma,x{:}A\to B\vdash\Delta}(\cup L)}{(\widehat{y}P\widehat{\sigma}\cdot\alpha)\widehat{\alpha}\nearrow\widehat{x}Q :\cdot \Gamma\vdash\Delta}(cut)$$

Also here the building of the union type for x (as in the logical rule for $\vee$) is mirrored by the automatic construction of a union type for the *plug* α.

Using union and intersection is also supported by the normal view for statements like $\Gamma\vdash\Delta$, in that the formulae in the context Γ are all necessary for the result, and not all the formulae in Δ necessarily follow from Γ; in other words, the formulae in the context Γ are connected through the logical '&', whereas those in Δ are connected through the logical '$\vee$'. And although '$\cap$' is not '&', and '$\cup$' is not '$\vee$', the link between these concepts is strong enough to justify our choice.

The notion we define below builds intersection types for *sockets*, and union types for *plugs*. However, a union type like $A\cup B$ for *sockets* is allowed, but only if both A and B can be

justified (see rule $(\cup L)$); similarly, an intersection type like $A \cap B$ for *plugs* is only allowed if both A and B can be justified (see rule $(\cap R)$; this corresponds to rule $(\cap I)$ in $\vdash_D$).

The following definition of types is a natural extension of the notion of types of the previous section, by adding union as a type constructor.

Definition 3.1 (INTERSECTION AND UNION TYPES, CONTEXTS) *i*) The set $\mathcal{T}$ of *intersection-union types*, ranged over by $A, B, \ldots$ is defined by:

$$\mathcal{T} \ ::= \ \varphi \mid \top \mid \bot \mid (\mathcal{T} \to \mathcal{T}) \mid (\mathcal{T} \cap \mathcal{T}) \mid (\mathcal{T} \cup \mathcal{T})$$

The set $\mathcal{T}_p$ is the set of *proper types*, defined by:

$$\mathcal{T}_p \ ::= \ \varphi \mid (\mathcal{T} \to \mathcal{T})$$

ii) A *context* Γ of *sockets* (Δ of *plugs*) is a partial mapping from *sockets* (*plugs*) to types in $\mathcal{T}$, represented as a set of statements with only distinct connectors as subjects. We write $x \in \Gamma$ ($\alpha \in \Delta$) if $x \in dom(\Gamma)$ ($\alpha \in dom(\Delta)$).

As before, we will omit unnecessary brackets in types, and the type constructors '$\cap$' and '$\cup$' will bind more strongly than '$\to$', so $A \cap B \to C \cap D$ stands for $((A \cap B) \to (C \cap D))$, $A \to B \cap C \to D$ stands for $(A \to ((B \cap C) \to D))$, and $(A \to B) \cap C \to D$ stands for $(((A \to B) \cap C) \to D)$. We will sometimes write the omissible brackets to aid readability.

We will consider a pre-order on types which takes into account the idempotence, commutativity and associativity of the intersection and union type constructors, and defines $\top$ to be the maximal element, and $\bot$ to be the minimal.

Definition 3.2 (RELATIONS ON TYPES) *i*) The relation $\leq$ is defined as the least pre-order on $\mathcal{T}$ such that:

$$A \leq \top \qquad A \cap B \leq A \qquad A \cap B \leq B \qquad \bot \leq A \qquad A \leq A \cup B \qquad B \leq A \cup B$$
$$A \leq C \ \& \ A \leq B \ \Rightarrow \ A \leq B \cap C \qquad\qquad A \leq C \ \& \ B \leq C \ \Rightarrow \ A \cup B \leq C$$

ii) The equivalence relation $\sim$ on types is defined by:

$$A \leq B \ \& \ B \leq A \ \Rightarrow \ A \sim B$$
$$A \sim C \ \& \ B \sim D \ \Rightarrow \ A \to B \sim C \to D$$

Since the relation defined here is a natural extension of that in Definition 2.6, we are free to use the same symbol.

The following is easy to prove:

Proposition 3.3 $(A \cap C) \cap B \sim A \cap (B \cap C)$ *and* $(A \cup C) \cup B \sim A \cup (B \cup C)$.

As before, we will write $\cap_{i \in \underline{n}} A_i$ (or $\cap_{\underline{n}} A_i$) for $A_1 \cap \cdots \cap A_n$ and $\top$ (*top*) for the empty intersection type, as well as $\cup_{i \in \underline{n}} A_i$ (or $\cup_{\underline{n}} A_i$) for $A_1 \cup \cdots \cup A_n$ and $\bot$ (*bottom*) for the empty union.

As mentioned above, the relation $\leq$ is *not* defined over arrow types, as in the system of [16]. More pointedly, we do not consider the type $A \to (C \cap (C \to D))$ smaller than $(A \to C) \cap (A \to C \to D)$; the type assignment system would not be closed under the relation.

Proposition 3.4 i) If $\cap_{\underline{n}} A_i \leq \cap_{\underline{m}} B_j$ *then, for every* B_j*, there exists an* A_{i_j} *such that* $A_{i_j} \leq B_j$.
ii) If $\cup_{\underline{n}} A_i \leq \cup_{\underline{m}} B_j$ *then, for every* A_i*, there exists a* B_{j_i} *such that* $A_i \leq B_{j_i}$.
iii) $(A \cup C) \cap (B \cup C) \sim (A \cap B) \cup C$*, and* $(A \cup C) \cap (B \cup C) \sim (A \cap B) \cup C$.

iv) $\cap_{\underline{n}} A_i \leq \cup_{\underline{m}} B_j$*, then there exists* $k \in \underline{n}, l \in \underline{m}$ *such that* $A_k \leq B_l$*.*
v) If $A \leq B$*, then* $A \cap B \sim A$ *and* $A \cup B \sim B$*.*

Definition 3.5 *i)* For contexts of *sockets* $\Gamma_1, \ldots, \Gamma_n$, the context $\Gamma_1 \cap \cdots \cap \Gamma_n$ is defined by:

$$x{:}A_1 \cap \cdots \cap A_m \in \Gamma_1 \cap \cdots \cap \Gamma_n$$

if and only if $\{x{:}A_1, \ldots, x{:}A_m\}$ is the set of all statements about x that occur in $\bigcup_{\underline{n}} \Gamma_i$, where $\bigcup$ is set-union. We write $\Gamma \cap x{:}A$ for the context of *sockets* $\Gamma \cap \{x{:}A\}$; as before, we will write $\Gamma, x{:}A$ for $\Gamma \cap x{:}A$ when $x \notin \Gamma$, and we will write $\cap_{i \in \underline{n}} \Gamma_i$ (or $\cap_{\underline{n}} \Gamma_i$) for $\Gamma_1 \cap \cdots \cap \Gamma_n$.

ii) For contexts of *plugs*, $\Delta_1, \ldots, \Delta_n$, the context $\Delta_1 \cup \cdots \cup \Delta_n$ is defined by:

$$\alpha{:}A_1 \cup \cdots \cup A_m \in \Delta_1 \cup \cdots \cup \Delta_n$$

if and only if $\{\alpha{:}A_1, \ldots, \alpha{:}A_m\}$ is the set of all statements about α that occur in $\bigcup_{\underline{n}} \Delta_i$. We write $\alpha{:}A \cup \Delta$ for the context of *plugs* $\{\alpha{:}A\} \cup \Delta$, and write $\alpha{:}A, \Delta$ for $\alpha{:}A \cup \Delta$ when $\alpha \notin \Delta$, and write $\cup_{i \in \underline{n}} \Delta_i$ (or $\cup_{\underline{n}} \Delta_i$) for $\Delta_1 \cup \cdots \cup \Delta_n$.

iii) We extend $\leq$ to contexts by: $\Gamma \leq \Gamma'$ if for all $x{:}A' \in \Gamma'$ there exists $x{:}A \in \Gamma$ such that $A \leq A'$, and $\Delta \leq \Delta'$ if for all $\alpha{:}A \in \Gamma$ there exists $\alpha{:}A' \in \Gamma'$ such that $A \leq A'$.

Proposition 3.6 For all $k \in \underline{n}$*:* $\cap_{\underline{n}} \Gamma_i \leq \Gamma_k$*, and* $\Delta_k \leq \cup_{\underline{n}} \Delta_i$*.*

We will now define a notion of intersection-union context assignment for $\mathcal{X}$.

Definition 3.7 (Intersection and Union Typing for $\mathcal{X}$) *i)* $\cap\cup$*-type judgements* are expressed via a ternary relation $P :\cdot \Gamma \vdash \Delta$, where Γ is a context of *sockets* and Δ is a context of *plugs*, and P is a term. We say that P is the *witness* of this judgement.

ii) *Intersection and union context assignment for* $\mathcal{X}$ is defined by the following sequent system:

$$(Ax): \frac{}{\langle y{\cdot}\alpha \rangle :\cdot \Gamma \cap y{:}A \vdash \alpha{:}A \cup \Delta}\ (A \text{ proper}) \qquad (cut): \frac{P :\cdot \Gamma_1 \vdash \alpha{:}A, \Delta_1 \quad Q :\cdot \Gamma_2, x{:}A \vdash \Delta_2}{P\widehat{\alpha} \dagger \widehat{x}Q :\cdot \Gamma_1 \cap \Gamma_2 \vdash \Delta_1 \cup \Delta_2}$$

$$(\rightarrow R): \frac{P :\cdot \Gamma, x{:}A \vdash \alpha{:}B, \Delta}{\widehat{x}P\widehat{\alpha}{\cdot}\beta :\cdot \Gamma \vdash \beta{:}A \rightarrow B \cup \Delta} \qquad (\rightarrow L): \frac{P :\cdot \Gamma_1 \vdash \alpha{:}A, \Delta_1 \quad Q :\cdot \Gamma_2, x{:}B \vdash \Delta_2}{P\widehat{\alpha}\,[y]\,\widehat{x}Q :\cdot \Gamma_1 \cap \Gamma_2 \cap y{:}A \rightarrow B \vdash \Delta_1 \cup \Delta_2}$$

$$(\cap R): \frac{P :\cdot \Gamma_1 \vdash \alpha{:}A_1, \Delta_1 \quad \cdots \quad P :\cdot \Gamma_n \vdash \alpha{:}A_n, \Delta_n}{P :\cdot \cap_{\underline{n}} \Gamma_i \vdash \alpha{:}\cap_{\underline{n}} A_i, \cup_{\underline{n}} \Delta_i}\ (n \geq 2) \qquad (\top): \frac{}{P :\cdot \Gamma \vdash \alpha{:}\top, \Delta}$$

$$(\bot): \frac{}{P :\cdot \Gamma, x{:}\bot \vdash \Delta} \qquad (\cup L): \frac{P :\cdot \Gamma_1, x{:}A_1 \vdash \Delta_1 \quad \cdots \quad P :\cdot \Gamma_n, x{:}A_n \vdash \Delta_n}{P :\cdot \cap_{\underline{n}} \Gamma_i, x{:}\cup_{\underline{n}} A_i \vdash \cup_{\underline{n}} \Delta_i}\ (n \geq 2)$$

NB: the rule (cut) is also used for the activated cuts. We will say that rule $(\cap R)$ and $(\top)$ *are applied to* α, and $(\cup L)$ and $(\bot)$ *are applied to* x, respectively.

iii) We will write $P :\cdot \Gamma \vdash \Delta$ if there exists a derivation that has this judgement in the bottom line, and write $\mathcal{D} :: P :\cdot \Gamma \vdash \Delta$ if we want to name that derivation.

The most important thing to notice is that, via the rules, *intersection types* are built of contexts of *sockets*, and *union types* are built for contexts of *plugs*. However, a union type can appear in a contexts of *sockets*, but only via rule $(\cup L)$; similarly, an intersection type can appear in a contexts of *plugs*, but only via rule $(\cap R)$. Moreover, the last two rules $(\bot)$ and $(\top)$ are almost special cases of rules $(\cup L)$ and $(\cap R)$, but for the fact that the intersection (or union) over zero sets would give the empty set; since weakening should be an admissible rule, we need to state

the zero cases as separate rules. This could be avoided by forcing the rules to agree on the contexts, as in, for example,

$$(\cap R):\ \frac{P :\cdot \Gamma \vdash \alpha{:}A_1,\Delta \quad \ldots \quad P :\cdot \Gamma \vdash \alpha{:}A_n,\Delta}{P :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} A_i,\Delta}\ (n \geq 0)$$

$$(\cup L):\ \frac{P :\cdot \Gamma,x{:}A_1 \vdash \Delta \quad \ldots \quad P :\cdot \Gamma,x{:}A_n \vdash \Delta}{P :\cdot \Gamma,x{:}\cup_{\underline{n}} A_i \vdash \Delta}\ (n \geq 0)$$

We have chosen for the present variant that allows for the combination of contexts since this is convenient when constructing derivations where it avoids additional weakening; this does not exclude the normal approach, since the contexts can be equal, whereby the alternative rules become derivable. Whenever possible, we will treat the rules $(\bot)$ and $(\top)$ as empty cases of $(\cup L)$ and $(\cap R)$, respectively, and will use a short-hand notation for $(\cap R)$ and $(\cup L)$:

$$(\cap R):\ \frac{P :\cdot \Gamma_i \vdash \alpha{:}A_i,\Delta_i \quad (\forall i \in \underline{n})}{P :\cdot \cap_{\underline{n}}\Gamma_i \vdash \alpha{:}\cap_{\underline{n}} A_i,\cup_{\underline{n}}\Delta_i}\ (n \geq 0)$$

$$(\cup L):\ \frac{P :\cdot \Gamma_i,x{:}A_i \vdash \Delta_i \quad (\forall i \in \underline{n})}{P :\cdot \cap_{\underline{n}}\Gamma_i,x{:}\cup_{\underline{n}} A_i \vdash \cup_{\underline{n}}\Delta_i}\ (n \geq 0)$$

so will allow the counter be zero as well in these rules, conveniently ignoring the discrepancy.

We consider the system to have a limited applicability for certain rules:

Definition 3.8 (PROPER DERIVATIONS) We call a derivation *proper* if it ends with either rule (Ax), $(\rightarrow R)$, $(\rightarrow L)$, or (cut) (we call those rules proper as well), and limit the applicability of the non-proper rules in that rules $(\cap R)$, $(\top)$, $(\cup L)$, or $(\bot)$ are followed by a proper rule that binds the connector to which they are applied. Moreover, at least *one* of the two sub-derivations of rule (cut) has to be proper[6].

This implies that in proper derivations rules $(\cap R)$ and $(\cup L)$ cannot be repeated, and that we can assume that immediate sub-derivations of proper derivations have a precise shape, as for example for $(\rightarrow L)$:

$$\frac{\dfrac{\boxed{\phantom{P :\cdot \Gamma_i \vdash \beta{:}C_i,\Delta_i \quad (\forall i \in \underline{n})}}}{P :\cdot \cap_{\underline{n}}\Gamma_i \vdash \beta{:}\cap_{\underline{n}} C_i,\cup_{\underline{n}}\Delta_i}\ (\cap R) \qquad \dfrac{\boxed{\phantom{P :\cdot \Gamma'_j,z{:}D_j \vdash \Delta'_j \quad (\forall j \in \underline{m})}}}{P :\cdot \cap_{\underline{m}}\Gamma'_j,z{:}\cup_{\underline{m}} D_j \vdash \cup_{\underline{m}}\Delta'_j}\ (\cup L)}{P\,\widehat{\beta}\,[x]\,\widehat{z}Q :\cdot \cap_{\underline{n}}\Gamma_i \cap \cap_{\underline{m}}\Gamma'_j \cap z{:}\cap_{\underline{n}} C_i \rightarrow \cup_{\underline{m}} D_j \vdash \cup_{\underline{n}}\Delta_i \cup \cup_{\underline{m}}\Delta'_j}\ (\rightarrow L)$$

where n and m can be 1 or even 0.

In the proof of Theorem 4.1, we will allow the construction of derivations using rules $(\cap R)$ and $(\cup L)$ from collections of derivations that not necessarily are proper, in fact constructing derivations shaped like

$$\frac{\dfrac{\boxed{\phantom{P :\cdot \Gamma^j_i \vdash \beta{:}C^j_i,\Delta^j_i \quad (\forall j \in \underline{m})}}}{P :\cdot \cap_{j\in\underline{m}}\Gamma^j_i \vdash \beta{:}\cap_{j\in\underline{m}} C^j_i,\cup_{j\in\underline{m}}\Delta^j_i}\ (\cap R) \qquad (\forall i \in \underline{n})}{P :\cdot \cap_{i\in\underline{n}}\cap_{j\in\underline{m}}\Gamma^j_i \vdash \beta{:}\cap_{i\in\underline{n}}\cap_{j\in\underline{m}} C^j_i,\cup_{i\in\underline{n}}\cup_{j\in\underline{m}}\Delta^j_i}\ (\cap R)$$

[6]This is much in the spirit of the notion of strict intersection type assignment [2], where rule $(\cap I)$ can only be used for the right-hand sub-derivation for rule $(\rightarrow E)$.

which seems to violate the restriction put on those rules. However, notice that then we can construct:

$$\dfrac{\boxed{\phantom{P :\cdot \Gamma_i^j \vdash \beta{:}C_i^j, \Delta_i^j}} \quad (\forall j \in \underline{m}, \forall i \in \underline{n})}{P :\cdot \cap_{i\in\underline{n}}\cap_{j\in\underline{m}}\Gamma_i^j \vdash \beta{:}\cap_{i\in\underline{n}}\cap_{j\in\underline{m}}C_i^j, \cup_{i\in\underline{n}}\cup_{j\in\underline{m}}\Delta_i^j}\ (\cap R)$$

so can remove (collapse) a cascade of $(\cap R)$; we can reason similarly for derivations ending with $(\cup L)$.

Example 3.9 Since $\bot$ and $\top$ are union/intersection types, they can be used on the right and on the left, respectively.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle x{\cdot}\alpha\rangle :\cdot x{:}\bot \vdash \alpha{:}\bot}\ (\bot) \quad \dfrac{}{\langle z{\cdot}\beta\rangle :\cdot z{:}A \vdash \beta{:}A}}{\langle x{\cdot}\alpha\rangle\widehat{\alpha}\ [y]\ \widehat{z}\langle z{\cdot}\beta\rangle :\cdot x{:}\bot, y{:}\bot{\to}A \vdash \beta{:}A}\ (\to L)}{\widehat{x}(\langle x{\cdot}\alpha\rangle\widehat{\alpha}\ [y]\ \widehat{z}\langle z{\cdot}\beta\rangle)\widehat{\beta}{\cdot}\gamma :\cdot y{:}\bot{\to}A \vdash \gamma{:}\bot{\to}A}\ (\to R)}{\widehat{y}(\widehat{x}(\langle x{\cdot}\alpha\rangle\widehat{\alpha}\ [y]\ \widehat{z}\langle z{\cdot}\beta\rangle)\widehat{\beta}{\cdot}\gamma)\widehat{\gamma}{\cdot}\delta :\cdot \vdash \delta{:}(\bot{\to}A){\to}\bot{\to}A}\ (\to R) \qquad \dfrac{\dfrac{\dfrac{}{\langle x{\cdot}\gamma\rangle :\cdot x{:}A, y{:}\top \vdash \gamma{:}A}}{\widehat{y}\langle x{\cdot}\gamma\rangle\widehat{\gamma}{\cdot}\beta :\cdot x{:}A \vdash \beta{:}\top{\to}A}\ (\to R)}{\widehat{x}(\widehat{y}\langle x{\cdot}\gamma\rangle\widehat{\gamma}{\cdot}\beta)\widehat{\beta}{\cdot}\alpha :\cdot \emptyset \vdash \alpha{:}A{\to}\top{\to}A}\ (\to R)$$

Notice that these terms correspond to $\llbracket \lambda yx.yx \rrbracket^{\lambda}_{\delta}$ and $\llbracket \lambda xy.x \rrbracket^{\lambda}_{\alpha}$, respectively.

For the *capsule*, we can show:

Lemma 3.10 $\langle x{\cdot}\alpha\rangle :\cdot \Gamma, x{:}A \vdash \alpha{:}B, \Delta$, *if and only if* $A \leq B$.

Proof: By easy induction. □

We can type all terms in normal form (*i.e.* that are *cut*-free).

Lemma 3.11 *If P is in normal form, then there exist Γ and Δ such that $P :\cdot \Gamma \vdash \Delta$.*

Proof: By induction on the structure of terms; if P is in normal form, then P is either:

$\langle x{\cdot}\alpha\rangle$: : Take any proper A, then $\langle x{\cdot}\alpha\rangle :\cdot x{:}A \vdash \alpha{:}A$ by rule (Ax).

$\widehat{x}Q\widehat{\alpha}{\cdot}\beta$: : Then by induction, there exist Γ, Δ such that $Q :\cdot \Gamma \vdash \Delta$; we can assume that x has a type A in Γ, and α a type B in Δ, added by weakening if necessary. Then, by rule $(\to R)$ we can construct $\widehat{x}Q\widehat{\alpha}{\cdot}\beta :\cdot \Gamma\backslash x \vdash \beta{:}A{\to}B \cup \Delta\backslash\alpha$.

$Q\widehat{\alpha}\,[x]\,\widehat{y}R$: : Then by induction, there exist $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2$ such that $Q :\cdot \Gamma_1 \vdash \Delta_1$ and $R :\cdot \Gamma_2 \vdash \Delta_2$; we can assume that α has a type A in Δ_1, and y a type B in Γ_2, added by weakening if necessary. Then, by rule $(\to L)$ we can construct $Q\widehat{\alpha}\,[x]\,\widehat{y}R :\cdot \Gamma_1 \cap \Gamma_2\backslash y \cap x{:}A{\to}B \vdash \Delta_1\backslash\alpha \cup \Delta_2$. □

Notice the role of intersection and union in the last part. We cannot show that the term $Q\widehat{\alpha} \dagger \widehat{y}R$ is typeable this way: we can assume that there exist $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2$ such that $Q :\cdot \Gamma_1 \vdash \alpha{:}A, \Delta_1$ and $R :\cdot \Gamma_2, x{:}B \vdash \Delta_2$, but have no way of knowing if A and B are the same type; we would normally need unification to establish this, which will likely be undecidable.

As will be argued below, this notion of type assignment is too liberal to obtain preservation of types under conversion. As illustrated above, the system is constructed to satisfy preservation of types under expansion (see Theorem 4.1), but we will see that it is not closed under reduction (Section 5). We will partly recover from this in Section 6, where we define restrictions of the system above that satisfy preservation of types under, respectively, CBN and CBV reduction. However, a natural consequence of these restrictions is that the systems no longer will be closed under expansion.

First we show some basic properties:

Proposition 3.12 (WEAKENING) *The following rules are admissible:*

$$(W):\ \frac{P :\cdot \Gamma \vdash \Delta}{P :\cdot \Gamma' \vdash \Delta'}\ (\Gamma' \leq \Gamma, \Delta \leq \Delta') \qquad (\cap L):\ \frac{P :\cdot \Gamma, x{:}A \vdash \Delta}{P :\cdot \Gamma, x{:}A \cap B \vdash \Delta} \qquad (\cup R):\ \frac{P :\cdot \Gamma \vdash \alpha{:}A, \Delta}{P :\cdot \Gamma \vdash \alpha{:}A \cup B, \Delta}$$

Notice that rules $(\cap L)$ and $(\cup R)$ are special cases of (W).

Since weakening is an admissible rule, we will, whenever convenient, assume that contexts are equal in sub-derivations, rather than being combined using intersection and union by application of the rule.

We can also show the following (standard) result:

Proposition 3.13 (THINNING) *The following rules are admissible:*

$$(Th\text{-}L):\ \frac{P :\cdot \Gamma \vdash \Delta}{P :\cdot \Gamma \backslash x \vdash \Delta}\ (x \notin fs(P)) \qquad (Th\text{-}R):\ \frac{P :\cdot \Gamma \vdash \Delta}{P :\cdot \Gamma \vdash \Delta \backslash \alpha}\ (\alpha \notin fp(P))$$

Applying these repeatedly gives that the following rule is admissible:

$$(T):\ \frac{P :\cdot \Gamma \vdash \Delta}{P :\cdot \{x{:}A \in \Gamma \mid x \in fs(P)\} \vdash \{\alpha{:}A \in \Delta \mid \alpha \in fp(P)\}}$$

Lemma 3.14 (ELIMINATION) *The following rules are admissible:*

$$(\cap E):\ \frac{P :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} A_i, \Delta}{P :\cdot \Gamma \vdash \alpha{:}A_j, \Delta}\ (j \in \underline{n}) \qquad (\cup E):\ \frac{P :\cdot \Gamma, x{:}\cup_{\underline{n}} A_i \vdash \Delta}{P :\cdot \Gamma, x{:}A_j \vdash \Delta}\ (j \in \underline{n})$$

Proof: By easy induction on the structure of derivations. □

The converse of the elimination rules does not hold. In particular, the system is truly classical in that it does not have *choice* (*i.e.*, we cannot show that, if $P :\cdot \Gamma \vdash \alpha{:}A \cup B, \Delta$ then either $P :\cdot \Gamma \vdash \alpha{:}A, \Delta$ or $P :\cdot \Gamma \vdash \alpha{:}B, \Delta$).

Example 3.15 Notice that

$$\frac{\overline{\langle x{\cdot}\delta\rangle :\cdot x{:}A \vdash \delta{:}A, \beta{:}B}}{\widehat{x}\langle x{\cdot}\delta\rangle\, \widehat{\beta}{\cdot}\delta :\cdot \vdash \delta{:}A \cup (A \to B)}\ (\to R) \qquad \frac{\overline{\langle x{\cdot}\gamma\rangle :\cdot x{:}A \vdash \gamma{:}A} \quad \overline{\langle v{\cdot}\alpha\rangle :\cdot v{:}C \vdash \alpha{:}C}}{\langle x{\cdot}\gamma\rangle\, \widehat{\gamma}\, [x]\, \widehat{v}\langle v{\cdot}\alpha\rangle :\cdot x{:}A \cap (A \to C) \vdash \alpha{:}C}\ (\to L)$$

but we cannot derive $\widehat{x}\langle x{\cdot}\delta\rangle\, \widehat{\beta}{\cdot}\delta :\cdot \vdash \delta{:}A$, nor $\widehat{x}\langle x{\cdot}\delta\rangle\, \widehat{\beta}{\cdot}\delta :\cdot \vdash \delta{:}A \to B$. As for the second derivation, neither can we derive $\langle x{\cdot}\gamma\rangle\, \widehat{\gamma}\, [x]\, \widehat{v}\langle v{\cdot}\alpha\rangle :\cdot x{:}A \vdash \alpha{:}C$, nor $\langle x{\cdot}\gamma\rangle\, \widehat{\gamma}\, [x]\, \widehat{v}\langle v{\cdot}\alpha\rangle :\cdot x{:}A \to C \vdash \alpha{:}C$.

To link our notion of type assignment to *Système* $D\top$, we can show that typeability is preserved by $\llbracket \cdot \rrbracket^{\lambda}_{\alpha}$:

Theorem 3.16 *If* $\Gamma \vdash_{\mathrm{D}} M{:}A$, *then* $\llbracket M \rrbracket^{\lambda}_{\alpha} :\cdot \Gamma \vdash \alpha{:}A$.

Proof: By induction on the structure of derivations in $\vdash_{D}$.
(Ax) : : Then $M \equiv x$, and $\Gamma = \Gamma', x{:}A$, so $x \notin \Gamma'$; let $A = \cap_{\underline{n}} A_i$. We can construct

$$\frac{\overline{\llbracket x \rrbracket^{\lambda}_{\alpha} :\cdot \Gamma', x{:}A_i \vdash \alpha{:}A_i}\ (Ax) \quad (\forall i \in \underline{n})}{\llbracket x \rrbracket^{\lambda}_{\alpha} :\cdot \Gamma', x{:}\cap_{\underline{n}} A_i \vdash \alpha{:}\cap_{\underline{n}} A_i}\ (\cap R)$$

$(\to I)$: : Then $M \equiv \lambda x.N$, $A = C \to D$, and $\Gamma, x{:}C \vdash_D N : D$. Then $\mathcal{D} :: \llbracket N \rrbracket^{\lambda}_{\beta} :\cdot\ \Gamma, x{:}C \vdash \beta{:}D$ exists by induction, and we can construct:

$$\dfrac{\boxed{\mathcal{D}}\ \llbracket N \rrbracket^{\lambda}_{\beta} :\cdot\ \Gamma, x{:}C \vdash \beta{:}D}{\widehat{x}\,\llbracket N \rrbracket^{\lambda}_{\beta}\,\widehat{\beta}\cdot\alpha :\cdot\ \Gamma \vdash \alpha{:}C \to D}\ (\to R)$$

Notice that $\widehat{x}\,\llbracket N \rrbracket^{\lambda}_{\beta}\,\widehat{\beta}\cdot\alpha = \llbracket \lambda x.N \rrbracket^{\lambda}_{\alpha}$.

$(\to E)$: : Then $M \equiv M_1 M_2$, and there exists B such that both $\Gamma \vdash_D M_1 : B \to A$ and $\Gamma \vdash_D M_2 : B$. By induction, both $\mathcal{D}_1 :: \llbracket M_1 \rrbracket^{\lambda}_{\gamma} :\cdot\ \Gamma \vdash \gamma{:}B \to A$, and $\mathcal{D}_2 :: \llbracket M_2 \rrbracket^{\lambda}_{\beta} :\cdot\ \Gamma \vdash \beta{:}B$ exist, and we can construct:

$$\dfrac{\boxed{\mathcal{D}_1}\ \llbracket M_1 \rrbracket^{\lambda}_{\gamma} :\cdot\ \Gamma \vdash \gamma{:}B \to A \qquad \dfrac{\boxed{\mathcal{D}_2}\ \llbracket M_2 \rrbracket^{\lambda}_{\beta} :\cdot\ \Gamma \vdash \beta{:}B \qquad \dfrac{}{\langle y\cdot\alpha \rangle :\cdot\ y{:}A \vdash \alpha{:}A}\ (Ax)}{\llbracket M_2 \rrbracket^{\lambda}_{\beta}\,\widehat{\beta}\,[x]\ \widehat{y}\langle y\cdot\alpha \rangle :\cdot\ \Gamma, x{:}B \to A \vdash \alpha{:}A}\ (\to L)}{\llbracket M_1 \rrbracket^{\lambda}_{\gamma}\,\widehat{\gamma} \dagger \widehat{x}(\llbracket M_2 \rrbracket^{\lambda}_{\beta}\,\widehat{\beta}\,[x]\ \widehat{y}\langle y\cdot\alpha \rangle) :\cdot\ \Gamma \vdash \alpha{:}A}\ (cut)$$

Notice that $\llbracket M_1 M_2 \rrbracket^{\lambda}_{\alpha} = \llbracket M_1 \rrbracket^{\lambda}_{\gamma}\,\widehat{\gamma} \dagger \widehat{x}(\llbracket M_2 \rrbracket^{\lambda}_{\beta}\,\widehat{\beta}\,[x]\ \widehat{y}\langle y\cdot\alpha \rangle)$, and that, by construction, $x, y \notin \Gamma$.

$(\cap I)$: : Then $A = \cap_{\underline{n}} A_i$, and we have $\Gamma \vdash_D M : A_i$, for all $i \in \underline{n}$. By induction, we have $\llbracket M \rrbracket^{\lambda}_{\alpha} :\cdot\ \Gamma \vdash \alpha{:}A_i$, for all $i \in \underline{n}$, so, by rule $(\cap R)$, also $\llbracket M \rrbracket^{\lambda}_{\alpha} :\cdot\ \Gamma \vdash \alpha{:}\cap_{\underline{n}} A_i$.

$(\cap E)$: : Then $A = A_i$, and we have $\Gamma \vdash_D M : \cap_{\underline{n}} A_i$. By induction, $\llbracket M \rrbracket^{\lambda}_{\alpha} :\cdot\ \Gamma \vdash \alpha{:}\cap_{\underline{n}} A_i$ for all $i \in \underline{n}$, so, by $(\cap E)$ (see Lemma 3.14), also $\llbracket M \rrbracket^{\lambda}_{\alpha} :\cdot\ \Gamma \vdash \alpha{:}A_i$.

$(\top)$: : Then $A = \top$; notice that $\llbracket P \rrbracket^{\lambda}_{\alpha} :\cdot\ \Gamma \vdash \alpha{:}\top$ by rule $(\top)$. □

4 Preservation of types under expansion and (some) reduction

One of the main properties of the intersection type assignment system is the preservation of types under both subject reduction and subject expansion. We will show the same results for our system for $\mathcal{X}$, but with restrictions. We are able to show the witness expansion result for the notion of context assignment of Definition 3.7, but for witness reduction, we will have to limit that notion.

We start by showing that witness expansion follows relatively easily.

Theorem 4.1 (WITNESS EXPANSION) *Let $P \to Q$: if $Q :\cdot\ \Gamma \vdash \Delta$ then $P :\cdot\ \Gamma \vdash \Delta$.*

Proof: By induction on the definition of $\to$, where we focus on the rules; we will only show some of the more interesting cases.

$$(\textit{exp-imp}) :: \ (\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) \to \begin{cases} (Q\widehat{\gamma} \dagger \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R \\ Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta} \dagger \widehat{z}R) \end{cases}, \text{ with } \alpha \notin fp(R),\ x \notin fs(Q,R).$$

We have two shapes for the (proper) derivation; first (with $n \geq 0$):

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1^i}\ Q :\cdot\ \Gamma \vdash \gamma{:}A_i, \Delta \qquad \boxed{\mathcal{D}_2^i}\ P :\cdot\ \Gamma, y{:}A_i \vdash \beta{:}B_i, \Delta}{Q\widehat{\gamma} \dagger \widehat{y}P :\cdot\ \Gamma \vdash \beta{:}B_i, \Delta \quad (\forall i \in \underline{n})}\ (cut)}{Q\widehat{\gamma} \dagger \widehat{y}P :\cdot\ \Gamma \vdash \beta{:}\cap_{\underline{n}} B_i, \Delta}\ (\cap R) \qquad \boxed{\mathcal{D}_3}\ R :\cdot\ \Gamma, z{:}\cap_{\underline{n}} B_i \vdash \Delta}{(Q\widehat{\gamma} \dagger \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R :\cdot\ \Gamma \vdash \Delta}\ (cut)$$

and we can construct:

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_2^i}\quad P :\cdot \Gamma, y{:}A_i \vdash \beta{:}B_i, \Delta \ \ (\forall i \in \underline{n})}{P :\cdot \Gamma, y{:}\cap_{\underline{n}} A_i \vdash \beta{:}\cap_{\underline{n}} B_i, \Delta}\,(\cap R)}{\widehat{y} P \widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} A_i \to \cap_{\underline{n}} B_i, \Delta}\,(\to R) \qquad \dfrac{\dfrac{\boxed{\mathcal{D}_1^i}\quad Q :\cdot \Gamma \vdash \gamma{:}A_i, \Delta \ \ (\forall i \in \underline{n})}{Q :\cdot \Gamma \vdash \gamma{:}\cap_{\underline{n}} A_i, \Delta}\,(\cap R) \qquad \boxed{\mathcal{D}_3}\quad R :\cdot \Gamma, z{:}\cap_{\underline{n}} B_i \vdash \Delta}{Q\widehat{\gamma}[x]\widehat{z}R :\cdot \Gamma, x{:}\cap_{\underline{n}} A_i \to \cap_{\underline{n}} B_i \vdash \Delta}\,(\to L)}{(\widehat{y} P \widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) :\cdot \Gamma \vdash \Delta}\,(cut)$$

or it is shaped as:

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad Q :\cdot \Gamma \vdash \gamma{:}A, \Delta \qquad \boxed{\mathcal{D}_2}\quad P :\cdot \Gamma, y{:}A \vdash \beta{:}\cup_{\underline{m}} C_j, \Delta}{Q\widehat{\gamma} \dagger \widehat{y}P :\cdot \Gamma \vdash \beta{:}\cup_{\underline{m}} C_j, \Delta}\,(cut) \qquad \dfrac{\boxed{\mathcal{D}_3^j}\quad R :\cdot \Gamma, z{:}C_j \vdash \Delta \ (\forall j \in \underline{m})}{R :\cdot \Gamma, z{:}\cup_{\underline{m}} C_j \vdash \Delta}\,(\cup L)}{(Q\widehat{\gamma} \dagger \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R :\cdot \Gamma \vdash \Delta}\,(cut)$$

(notice that $\beta \notin fp(Q)$) and then we can construct:

$$\dfrac{\dfrac{\boxed{\mathcal{D}_2}\quad P :\cdot \Gamma, y{:}A \vdash \beta{:}\cup_{\underline{m}} C_j, \Delta}{\widehat{x} P \widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash \alpha{:}A \to \cup_{\underline{m}} C_j, \Delta}\,(\to R) \qquad \dfrac{\boxed{\mathcal{D}_1}\quad Q :\cdot \Gamma \vdash \gamma{:}A, \Delta \qquad \dfrac{\boxed{\mathcal{D}_3^j}\quad R :\cdot \Gamma, z{:}C_j \vdash \Delta \ (\forall j \in \underline{m})}{R :\cdot \Gamma, z{:}\cup_{\underline{m}} C_j \vdash \Delta}\,(\cup L)}{Q\widehat{\gamma}[x]\widehat{z}R :\cdot \Gamma, x{:}A \to \cup_{\underline{m}} C_j \vdash \Delta}\,(\to L)}{(\widehat{y} P \widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) :\cdot \Gamma \vdash \Delta}\,(cut)$$

The case of $Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta} \dagger \widehat{z}R)$ is similar.

$(cap\not\nearrow) :: \langle y\cdot\beta\rangle \widehat{\alpha} \not\nearrow \widehat{x}P \to \langle y\cdot\beta\rangle$, with $\beta \neq \alpha$. By Lemma 3.10, there exists $A \leq B$ such that

$$\boxed{\mathcal{D}}\quad \langle y\cdot\beta\rangle :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta$$

and, using weakening, we can construct

$$\dfrac{\dfrac{\boxed{\mathcal{D}}\quad \langle y\cdot\beta\rangle :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta}{\langle y\cdot\beta\rangle :\cdot \Gamma, y{:}A \vdash \alpha{:}\bot, \beta{:}B, \Delta}\,(W) \qquad \dfrac{}{P :\cdot \Gamma, x{:}\bot \vdash \Delta}\,(\bot)}{\langle y\cdot\beta\rangle \widehat{\alpha} \not\nearrow \widehat{x}P :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta}\,(cut)$$

$(exp\text{-}outs\not\nearrow) :: (\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \not\nearrow \widehat{x}Q \to (\widehat{y}(P\widehat{\alpha} \not\nearrow \widehat{x}Q)\widehat{\beta}\cdot\gamma)\widehat{\gamma} \dagger \widehat{x}Q$, with γ fresh. We have two cases: first (with $n \geq 0$),

$$\dfrac{\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1^i}\quad P :\cdot \Gamma, y{:}C_i \vdash \alpha{:}E_i, \beta{:}D_i, \Delta \qquad \boxed{\mathcal{D}_2^i}\quad Q :\cdot \Gamma, x{:}E_i \vdash \Delta}{P\widehat{\alpha} \not\nearrow \widehat{x}Q :\cdot \Gamma, y{:}C_i \vdash \beta{:}D_i, \Delta}\,(cut)}{\widehat{y}(P\widehat{\alpha} \not\nearrow \widehat{x}Q)\widehat{\beta}\cdot\gamma :\cdot \Gamma \vdash \gamma{:}C_i \to D_i, \Delta}\,(\to R)\ (\forall i \in \underline{n})}{\widehat{y}(P\widehat{\alpha} \not\nearrow \widehat{x}Q)\widehat{\beta}\cdot\gamma :\cdot \Gamma \vdash \gamma{:}\cap_{\underline{n}}(C_i \to D_i), \Delta}\,(\cap R) \qquad \boxed{\mathcal{D}_3}\quad Q :\cdot \Gamma, x{:}\cap_{\underline{n}}(C_i \to D_i) \vdash \Delta}{(\widehat{y}(P\widehat{\alpha} \not\nearrow \widehat{x}Q)\widehat{\beta}\cdot\gamma)\widehat{\gamma} \dagger \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$

and we can then construct:

$$\dfrac{\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1^i}}{P :\cdot \Gamma, y{:}C_i \vdash \alpha{:}E_i, \beta{:}D_i, \Delta}}{P :\cdot \Gamma, y{:}C_i \vdash \alpha{:}\cup_{\underline{n}} E_i, \beta{:}D_i, \Delta}\,(W)}{\widehat{y}P\widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash \alpha{:}(C_i{\to}D_i) \cup \cup_{\underline{n}} E_i, \Delta}\,(\to R)\quad (\forall i{\in}\underline{n})}{\widehat{y}P\widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}}(C_i{\to}D_i) \cup \cup_{\underline{n}} E_i, \Delta}\,(\cap R) \qquad \dfrac{\begin{array}{c}\boxed{\mathcal{D}_3}\\ Q :\cdot \Gamma, x{:}\cap_{\underline{n}}(C_i{\to}D_i) \vdash \Delta\\ \vdots \qquad \boxed{\mathcal{D}_2^i}\\ Q :\cdot \Gamma, x{:}E_i \vdash \Delta\ (\forall i{\in}\underline{n})\end{array}}{Q :\cdot \Gamma, x{:}\cap_{\underline{n}}(C_i{\to}D_i) \cup \cup_{\underline{n}} E_i \vdash \Delta}\,(\cup L)}{(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \not\prec \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$

(notice that we use $\cap_{\underline{n}}(A_i{\cup}B) \sim \cap_{\underline{n}} A_i{\cup}B$ - see Lemma 3.4 ((iii))). For the second case ($m \geq 0$):

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1}}{P :\cdot \Gamma, y{:}C \vdash \alpha{:}E, \beta{:}D, \gamma{:}F, \Delta} \quad \dfrac{\boxed{\mathcal{D}_2}}{Q :\cdot \Gamma, x{:}E \vdash \Delta}}{P\widehat{\alpha} \not\prec \widehat{x}Q :\cdot \Gamma, y{:}C \vdash \beta{:}D, \gamma{:}F, \Delta}\,(cut)}{\widehat{y}(P\widehat{\alpha} \not\prec \widehat{x}Q)\widehat{\beta}\cdot\gamma :\cdot \Gamma \vdash \gamma{:}C{\to}D{\cup}F, \Delta}\,(\to R) \qquad \dfrac{\dfrac{\boxed{\mathcal{D}_3^j}}{Q :\cdot \Gamma, x{:}B_j \vdash \Delta}\quad (\forall j{\in}\underline{m})}{Q :\cdot \Gamma, x{:}\cup_{\underline{m}} B_j \vdash \Delta}\,(\cup L)}{(\widehat{y}(P\widehat{\alpha} \not\prec \widehat{x}Q)\widehat{\beta}\cdot\gamma)\widehat{\gamma} \dagger \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$

so $C{\to}D{\cup}F = \cup_{\underline{m}} B_j$, and $C{\to}D = B_h$ for some $h \in \underline{m}$, and we can then construct:

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1}}{P :\cdot \Gamma, y{:}C \vdash \alpha{:}E, \beta{:}D, \gamma{:}F, \Delta}\,(T)}{P :\cdot \Gamma, y{:}C \vdash \alpha{:}E, \beta{:}D, \Delta}}{\widehat{y}P\widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash \alpha{:}(C{\to}D){\cup}E, \Delta}\,(\to R) \qquad \dfrac{\boxed{\mathcal{D}_3^h} \quad \boxed{\mathcal{D}_2}}{Q :\cdot \Gamma, x{:}C{\to}D \vdash \Delta \quad Q :\cdot \Gamma, x{:}E \vdash \Delta}\,(\cup L)}{(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \not\prec \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$

Notice that we can argue that $(\cup L)$ is obsolete here; below we will ignore these cases.

$(imp\not\prec) :: (P\widehat{\beta}[z]\widehat{y}Q)\widehat{\alpha} \not\prec \widehat{z}R \to (P\widehat{\alpha} \not\prec \widehat{x}R)\widehat{\beta}\,[z]\,\widehat{y}(Q\widehat{\alpha} \not\prec \widehat{x}R)$. We can assume that y, β are not free in R; the derivation for the right-hand side is shaped as follows (with $n, m \geq 0$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1^i}}{P :\cdot \Gamma \vdash \alpha{:}C_i, \beta{:}A_i, \Delta} \quad \dfrac{\boxed{\mathcal{D}_2^i}}{R :\cdot \Gamma, x{:}C_i \vdash \Delta}}{P\widehat{\alpha} \not\prec \widehat{x}R :\cdot \Gamma \vdash \beta{:}A_i, \Delta}\,(cut)\quad (\forall i{\in}\underline{n})}{P\widehat{\alpha} \not\prec \widehat{x}R :\cdot \Gamma \vdash \beta{:}\cap_{\underline{n}} A_i, \Delta}\,(\cap R) \qquad \dfrac{\dfrac{\boxed{\mathcal{D}_3^j} \quad \boxed{\mathcal{D}_4^j}}{Q :\cdot \Gamma, y{:}B_j \vdash \alpha{:}D_j, \Delta \quad R :\cdot \Gamma, x{:}D_j \vdash \Delta}\,(cut)}{Q\widehat{\alpha} \not\prec \widehat{x}R :\cdot \Gamma, y{:}B_j \vdash \Delta}\quad (\forall j{\in}\underline{m})}{Q\widehat{\alpha} \not\prec \widehat{x}R :\cdot \Gamma, y{:}\cup_{\underline{m}} B_j \vdash \Delta}\,(\cup L)}{(P\widehat{\alpha} \not\prec \widehat{x}R)\widehat{\beta}\,[z]\,\widehat{y}(Q\widehat{\alpha} \not\prec \widehat{x}R) :\cdot \Gamma{\cap}z{:}\cap_{\underline{n}} A_i{\to}\cup_{\underline{m}} B_j \vdash \Delta}\,(\to L)$$

and we can construct:

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1^i}}{P :\cdot \Gamma \vdash \alpha{:}C_i, \beta{:}A_i, \Delta\ (\forall i{\in}\underline{n})}}{P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{n}} C_i, \beta{:}\cap_{\underline{n}} A_i, \Delta}\,(\cap R) \quad \begin{array}{c}\dfrac{\boxed{\mathcal{D}_3^j}}{Q :\cdot \Gamma, y{:}B_j \vdash \alpha{:}D_j, \Delta}\,(\cup L)\\ Q :\cdot \Gamma, y{:}\cup_{\underline{m}} B_j \vdash \alpha{:}\cup_{\underline{m}} D_j, \Delta\\ \vdots\end{array}}{P\widehat{\beta}[z]\widehat{y}Q :\cdot \Gamma{\cap}z{:}\cap_{\underline{n}} A_i{\to}\cup_{\underline{m}} B_j \vdash \alpha{:}\cup_{\underline{n}} C_i \cup \cup_{\underline{m}} D_j, \Delta}\,(\to L) \qquad \dfrac{\begin{array}{c}\boxed{\mathcal{D}_4^j}\\ R :\cdot \Gamma, x{:}D_j \vdash \Delta\ (\forall j{\in}\underline{m})\\ \vdots \qquad \boxed{\mathcal{D}_2^i}\\ R :\cdot \Gamma, x{:}C_i \vdash \Delta\ (\forall i{\in}\underline{n})\end{array}}{R :\cdot \Gamma, x{:}\cup_{\underline{n}} C_i \cup \cup_{\underline{m}} D_j \vdash \Delta}\,(\cup L)}{(P\widehat{\beta}[z]\widehat{y}Q)\widehat{\alpha} \not\prec \widehat{x}R :\cdot \Gamma{\cap}z{:}\cap_{\underline{n}} A_i{\to}\cup_{\underline{m}} B_j \vdash \Delta}\,(cut)$$

$(\nwarrow cap) :: P\widehat{\alpha} \nwarrow \widehat{x}\langle y{\cdot}\beta\rangle \to \langle y{\cdot}\beta\rangle$, $y \neq x$. By Lemma 3.10, $\mathcal{D} :: \langle y{\cdot}\beta\rangle :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta$ for some

$B \geq A$ and, using weakening, we can construct

$$\dfrac{\dfrac{}{P :\cdot \Gamma \vdash \beta{:}\top, \Delta}\,(\top) \qquad \dfrac{\boxed{\mathcal{D}}\;\; \langle y{\cdot}\beta\rangle :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta}{\langle y{\cdot}\beta\rangle :\cdot \Gamma, y{:}A, x{:}\top \vdash \beta{:}B, \Delta}\,(W)}{P\widehat{\alpha}\nwarrow\widehat{x}\langle y{\cdot}\beta\rangle :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta}\,(cut)$$

($\nwarrow$*imp-outs*) :: $P\widehat{\alpha}\nwarrow\widehat{x}(Q\widehat{\beta}[x]\,\widehat{y}R) \to P\widehat{\alpha}\dagger\widehat{v}((P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\,[v]\;\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}R))$, with v fresh. We have two cases; the first, with $(\cap R)$ applied to the left-most P, as argued above for the case ($exp\text{-}outs\nearrow$), will imply that the number of types in the intersection is 1. The second case is (with $l, m, n \geq 0$):

$$\dfrac{\boxed{\mathcal{D}_1}\;\; P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{n}}(\cap_{\underline{m}} D^i_j \to \cup_{\underline{l}} F^i_k), \Delta \qquad \dfrac{\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}^{ij}_2}}{P :\cdot \Gamma \vdash \alpha{:}C^i_j, \Delta} \quad \dfrac{\boxed{\mathcal{D}^{ij}_3}}{Q :\cdot \Gamma, x{:}C^i_j \vdash \beta{:}D^i_j, \Delta}}{P\widehat{\alpha}\nwarrow\widehat{x}Q :\cdot \Gamma \vdash \beta{:}D^i_j, \Delta \quad (\forall j{\in}\underline{m})}\,(cut)}{P\widehat{\alpha}\nwarrow\widehat{x}Q :\cdot \Gamma \vdash \beta{:}\cap_{j\in\underline{m}} D^i_j, \Delta}\,(\cap R) \quad \dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}^{ik}_4}}{P :\cdot \Gamma \vdash \alpha{:}E^i_k, \Delta} \quad \dfrac{\boxed{\mathcal{D}^{ik}_5}}{R :\cdot \Gamma, y{:}F^i_k, x{:}E^i_k \vdash \Delta}}{P\widehat{\alpha}\nwarrow\widehat{x}R :\cdot \Gamma, y{:}F^i_k \vdash \Delta \quad (\forall k{\in}\underline{l})}\,(cut)}{P\widehat{\alpha}\nwarrow\widehat{x}R :\cdot \Gamma, y{:}\cup_{k\in\underline{l}} F^i_k \vdash \Delta}\,(\cup L)}{(P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\,[v]\;\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}R) :\cdot \Gamma, v{:}\cap_{\underline{m}} D^i_j \to \cup_{\underline{l}} F^i_k \vdash \Delta \quad (\forall i{\in}\underline{n})}\,(imp)}{(P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\,[v]\;\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}R) :\cdot \Gamma, v{:}\cup_{\underline{n}}(\cap_{\underline{m}} D^i_j \to \cup_{\underline{l}} F^i_k) \vdash \Delta}\,(\cup L)}{P\widehat{\alpha}\dagger\widehat{v}((P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\,[v]\;\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}R)) :\cdot \Gamma \vdash \Delta}\,(cut)$$

and we can construct:

$$\dfrac{\dfrac{\boxed{\mathcal{D}^{ij}_2}\;\; P :\cdot \Gamma \vdash \alpha{:}C^i_j, \Delta\;(\forall j{\in}\underline{m}) \qquad \dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}^{ij}_3}}{Q :\cdot \Gamma, x{:}C^i_j \vdash \beta{:}D^i_j, \Delta\;(\forall j{\in}\underline{m})}\,(\cap R)}{Q :\cdot \Gamma, x{:}\cap_{\underline{m}} C^i_j \vdash \beta{:}\cap_{\underline{m}} D^i_j, \Delta}}{Q :\cdot \Gamma, x{:}\cap_{\underline{n}}\cap_{\underline{m}} C^i_j \vdash \beta{:}\cap_{\underline{m}} D^i_j, \Delta}\,(W) \quad \dfrac{\dfrac{\boxed{\mathcal{D}^{ik}_5}}{R :\cdot \Gamma, y{:}F^i_k, x{:}E^i_k \vdash \Delta\;(\forall k{\in}\underline{l})}}{R :\cdot \Gamma, y{:}\cup_{\underline{l}} F^i_k, x{:}\cap_{\underline{l}} E^i_k \vdash \Delta}\,(\cup L)}{R :\cdot \Gamma, y{:}\cup_{\underline{l}} F^i_k, x{:}\cap_{\underline{n}}\cap_{\underline{l}} E^i_k \vdash \Delta}\,(W)}{Q\widehat{\beta}[x]\,\widehat{y}R :\cdot \Gamma, x{:}(\cap_{\underline{n}}\cap_{\underline{m}} C^i_j) \cap (\cap_{\underline{n}}\cap_{\underline{l}} E^i_k) \cap (\cap_{\underline{m}} D^i_j \to \cup_{\underline{l}} F^i_k) \vdash \Delta \quad (\forall i{\in}\underline{n})}\,(imp)}{P :\cdot \Gamma \vdash \alpha{:}C^i_j, \Delta\;(\forall j{\in}\underline{m})\,Q\widehat{\beta}[x]\,\widehat{y}R :\cdot \Gamma, x{:}(\cap_{\underline{n}}\cap_{\underline{m}} C^i_j) \cap (\cap_{\underline{n}}\cap_{\underline{l}} E^i_k) \cap (\cup_{\underline{n}}(\cap_{\underline{m}} D^i_j \to \cup_{\underline{l}} F^i_k)) \vdash \Delta}\,(\cup L)$$

$$\dfrac{\begin{array}{c}\vdots\\ P :\cdot \Gamma \vdash \alpha{:}C^i_j, \Delta\;(\forall j{\in}\underline{m})\,Q\widehat{\beta}[x]\,\widehat{y}R :\cdot \Gamma, x{:}(\cap_{\underline{n}}\cap_{\underline{m}} C^i_j) \cap (\cap_{\underline{n}}\cap_{\underline{l}} E^i_k) \cap (\cup_{\underline{n}}(\cap_{\underline{m}} D^i_j \to \cup_{\underline{l}} F^i_k)) \vdash \Delta\end{array} \qquad \dfrac{\boxed{\mathcal{D}^{ik}_4}\;\; P :\cdot \Gamma \vdash \alpha{:}E^i_k, \Delta\;(\forall k{\in}\underline{l}) \quad \boxed{\mathcal{D}_1}\;\; P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{n}}(\cap_{\underline{m}} D^i_j \to \cup_{\underline{l}} F^i_k), \Delta}{P :\cdot \Gamma \vdash \alpha{:}(\cap_{\underline{n}}\cap_{\underline{m}} C^i_j) \cap (\cap_{\underline{n}}\cap_{\underline{k}} E^i_k) \cap (\cup_{\underline{n}}(\cap_{\underline{m}} D^i_j \to \cup_{\underline{l}} F^i_k)), \Delta}\,(\cap R)}{P\widehat{\alpha}\nwarrow\widehat{x}(Q\widehat{\beta}[x]\,\widehat{y}R) :\cdot \Gamma \vdash \Delta}\,(cut)$$

(here we use that $\cup_{\underline{n}}(A \cap C_i) \sim A \cap \cup_{\underline{n}} C_i$, again by Lemma 3.4 ((*iii*))).

($\nwarrow$*imp-ins*) :: $P\widehat{\alpha}\nwarrow\widehat{x}(Q\widehat{\beta}[z]\,\widehat{y}R) \to (P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\,[z]\;\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{z}R)$, $x \neq z$. We have:

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}^i_1}}{P :\cdot \Gamma \vdash \alpha{:}C_i, \Delta} \quad \dfrac{\boxed{\mathcal{D}^i_2}}{Q :\cdot \Gamma, x{:}C_i \vdash \beta{:}A_i, \Delta}}{P\widehat{\alpha}\nwarrow\widehat{x}Q :\cdot \Gamma \vdash \beta{:}A_i, \Delta \quad (\forall i{\in}\underline{n})}\,(cut)}{P\widehat{\alpha}\nwarrow\widehat{x}Q :\cdot \Gamma \vdash \beta{:}\cap_{\underline{n}} A_i, \Delta}\,(\cap R) \quad \dfrac{\dfrac{\boxed{\mathcal{D}^j_3}}{P :\cdot \Gamma \vdash \alpha{:}D_j, \Delta} \quad \dfrac{\boxed{\mathcal{D}^j_4}}{R :\cdot \Gamma, y{:}B_j, x{:}D_j \vdash \Delta}}{P\widehat{\alpha}\nwarrow\widehat{x}R :\cdot \Gamma, y{:}B_j \vdash \Delta \quad (\forall j{\in}\underline{m})}\,(cut)}{P\widehat{\alpha}\nwarrow\widehat{x}R :\cdot \Gamma, y{:}\cup_{\underline{m}} B_j \vdash \Delta}\,(\cup L)}{(P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\,[z]\;\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}R) :\cdot \Gamma, z{:}\cap_{\underline{n}} A_i \to \cup_{\underline{m}} B_j \vdash \Delta}\,(imp)$$

and can construct

$$\dfrac{\dfrac{\begin{array}{c}\boxed{\mathcal{D}_1^i}\\ P :\cdot \Gamma \vdash \alpha{:}C_i,\Delta\ (\forall i{\in}\underline{n})\\ \vdots\\ \boxed{\mathcal{D}_3^j}\\ P :\cdot \Gamma \vdash \alpha{:}D_j,\Delta\ (\forall j{\in}\underline{m})\end{array}}{P :\cdot \Gamma \vdash \alpha{:}(\cap_{\underline{m}} C_j) \cap (\cap_{\underline{m}} D_j),\Delta}\,(\cap R) \qquad \begin{array}{c}\dfrac{\dfrac{\boxed{\mathcal{D}_2^i}}{Q :\cdot \Gamma, x{:}C_i \vdash \beta{:}A_i,\Delta\ (\forall i{\in}\underline{n})}}{Q :\cdot \Gamma, x{:}\cap_{\underline{n}} C_i \vdash \beta{:}\cap_{\underline{n}} A_i,\Delta}\,(\cap R) \quad \dfrac{\boxed{\mathcal{D}_4^j}}{R :\cdot \Gamma, y{:}B_j, x{:}D_j \vdash \Delta\ (\forall j{\in}\underline{m})}\,(\cup L)}{Q\widehat{\beta}[z]\,\widehat{y}R :\cdot \Gamma, x{:}(\cap_{\underline{n}} C_i) \cap (\cap_{\underline{m}} D_j), z{:}\cap_{\underline{n}} A_i \to \cup_{\underline{m}} B_j \vdash \Delta}\,(imp)\\ \vdots\end{array}}{P\widehat{\alpha}\,\text{\rotatebox{90}{$\in$}}\,\widehat{x}(Q\widehat{\beta}[z]\,\widehat{y}R) :\cdot \Gamma, z{:}\cap_{\underline{n}} A_i \to \cup_{\underline{m}} B_j \vdash \Delta}\,(cut)$$

□

As for reduction, we can only show a partial result, as we will do in the following sections. Before we come to that, we show the (unrestricted) witness reduction results we *can* prove. We start by showing that the logical rules are sound.

Theorem 4.2 *If $P\widehat{\alpha} \dagger \widehat{x}Q \to R$ via a logical rule, and $P\widehat{\alpha} \dagger \widehat{x}Q :\cdot \Gamma \vdash \Delta$, then also $R :\cdot \Gamma \vdash \Delta$.*

Proof: We consider the four cases.

$(Ax) :: \langle y{\cdot}\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x{\cdot}\beta\rangle \to \langle y{\cdot}\alpha\rangle$. Then, by Lemma 3.10 there are $A \leq B \leq C$ such that:

$$\dfrac{\boxed{\phantom{\langle y{\cdot}\alpha\rangle :\cdot \Gamma, y{:}A \vdash \alpha{:}B,\Delta}} \qquad \boxed{\phantom{\langle x{\cdot}\beta\rangle :\cdot \Gamma, x{:}B \vdash \beta{:}C,\Delta}}}{\langle y{\cdot}\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x{\cdot}\beta\rangle :\cdot \Gamma, x{:}A \vdash \beta{:}C,\Delta}\,(cut)$$

Since then also $A \leq C$, by Lemma 3.10 also $\langle y{\cdot}\beta\rangle :\cdot \Gamma, y{:}A \vdash \beta{:}C,\Delta$.

$(exp) :: (\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x{\cdot}\gamma\rangle \to \widehat{y}P\widehat{\beta}{\cdot}\gamma$, with $\alpha \notin fp(P)$; we can assume that α does not occur in Δ, but cannot assume that for γ. The derivation is constructed either as (with $n \geq 0$):

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_i}}{P :\cdot \Gamma, y{:}B_i \vdash \beta{:}C_i, \gamma{:}A_i,\Delta}}{\widehat{y}P\widehat{\beta}{\cdot}\alpha :\cdot \Gamma \vdash \alpha{:}B_i{\to}C_i, \gamma{:}A_i,\Delta}\,(\to R)\ (\forall i{\in}\underline{n})}{\widehat{y}P\widehat{\beta}{\cdot}\alpha :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}}(B_i{\to}C_i), \gamma{:}\cup_{\underline{n}} A_i,\Delta}\,(\cap R) \qquad \boxed{\phantom{\langle x{\cdot}\gamma\rangle :\cdot \Gamma, x{:}\cap_{\underline{n}}(B_i{\to}C_i) \vdash \gamma{:}E,\Delta}}}{(\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x{\cdot}\gamma\rangle :\cdot \Gamma \vdash \gamma{:}E \cup \cup_{\underline{n}} A_i,\Delta'}\,(cut)$$

with $\cap_{\underline{n}}(B_i{\to}C_i) \leq E$ by Lemma 3.10, so there exists $h \in \underline{n}$ such that $B_h{\to}C_h \leq E$. For the right-hand side we can construct:

$$\dfrac{\dfrac{\boxed{\mathcal{D}_h}}{P :\cdot \Gamma, y{:}B_h \vdash \beta{:}C_h, \gamma{:}A_h,\Delta}}{\widehat{y}P\widehat{\beta}{\cdot}\gamma :\cdot \Gamma \vdash \gamma{:}(B_h{\to}C_h) \cup A_h,\Delta}\,(\to R)}{\widehat{y}P\widehat{\beta}{\cdot}\gamma :\cdot \Gamma \vdash \gamma{:}E \cup \cup_{\underline{n}} A_i,\Delta}\,(W)$$

Else, the derivation is constructed using $(\cup L)$ for $\langle x{\cdot}\gamma\rangle$, but, as above, we can then argue that there is only one type in the union.

$(imp) :: \langle y{\cdot}\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(P\widehat{\beta}[x]\,\widehat{z}Q) \to P\widehat{\beta}[y]\,\widehat{y}Q$, with $x \notin fs(P,Q)$. We have only one case to con-

sider:

$$\dfrac{\dfrac{}{\langle y{\cdot}\alpha\rangle :\cdot\ \Gamma,y{:}A \vdash \alpha{:}\cup_{\underline{m}}(C_j{\to}D_j),\Delta} \qquad \dfrac{\dfrac{\begin{array}{c}\mathcal{D}_2^j\\ P :\cdot\ \Gamma,y{:}A \vdash \beta{:}C_j,\Delta\end{array} \quad \begin{array}{c}\mathcal{D}_3^j\\ Q :\cdot\ \Gamma,y{:}A,z{:}D_j \vdash \Delta\end{array}}{P\widehat{\beta}[x]\widehat{z}Q :\cdot\ \Gamma,y{:}A,x{:}C_j{\to}D_j \vdash \Delta}\,(\to L) \quad (\forall j{\in}\underline{m})}{P\widehat{\beta}[x]\widehat{z}Q :\cdot\ \Gamma,y{:}A,x{:}\cup_{\underline{m}}(C_j{\to}D_j) \vdash \Delta}\,(\cup L)}{\langle y{\cdot}\alpha\rangle\widehat{\alpha}\dagger\widehat{x}(P\widehat{\beta}[x]\widehat{z}Q) :\cdot\ \Gamma,y{:}A \vdash \Delta}\,(cut)$$

Notice that, by Lemma 3.10, $A \leq \cup_{\underline{m}}(C_j{\to}D_j)$, so $A{\cap}\cup_{\underline{m}}(C_j{\to}D_j) = A$; we can derive:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_2^j\\ P :\cdot\ \Gamma,y{:}A \vdash \beta{:}C_j,\Delta\end{array} \quad \begin{array}{c}\mathcal{D}_3^j\\ Q :\cdot\ \Gamma,y{:}A,z{:}D_j \vdash \Delta\end{array}}{P\widehat{\beta}[y]\widehat{z}Q :\cdot\ \Gamma,y{:}A{\cap}(C_j{\to}D_j) \vdash \Delta}\,(\to L) \quad (\forall j{\in}\underline{m})}{P\widehat{\beta}[y]\widehat{z}Q :\cdot\ \Gamma,y{:}A{\cap}\cup_{\underline{m}}(C_j{\to}D_j) \vdash \Delta}\,(\cup L)$$

$$(\textit{exp-imp}) :: \ (\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha}\dagger\widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) \ \to \ \begin{cases}(Q\widehat{\gamma}\dagger\widehat{y}P)\widehat{\beta}\dagger\widehat{z}R\\ Q\widehat{\gamma}\dagger\widehat{y}(P\widehat{\beta}\dagger\widehat{z}R)\end{cases}, \text{ with } \alpha \notin fp(R), x \notin fs(Q,R).$$

We have either the following structure of the derivation:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1^i\\ P :\cdot\ \Gamma,y{:}A_i \vdash \beta{:}B_i,\Delta\end{array}}{\widehat{y}P\widehat{\beta}{\cdot}\alpha :\cdot\ \Gamma \vdash \alpha{:}A_i{\to}B_i,\Delta}\,(\to R) \quad (\forall i{\in}\underline{n})}{\widehat{y}P\widehat{\beta}{\cdot}\alpha :\cdot\ \Gamma \vdash \alpha{:}\cap_{\underline{n}}(A_i{\to}B_i),\Delta}\,(\cap R) \quad \dfrac{\begin{array}{c}\mathcal{D}_2\\ Q :\cdot\ \Gamma \vdash \gamma{:}C,\Delta\end{array} \quad \begin{array}{c}\mathcal{D}_3\\ R :\cdot\ \Gamma,z{:}D \vdash \Delta\end{array}}{Q\widehat{\gamma}[x]\widehat{z}R :\cdot\ \Gamma{\cap}x{:}C{\to}D \vdash \Delta}\,(\to L)}{(\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha}\dagger\widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) :\cdot\ \Gamma \vdash \Delta}\,(cut)$$

so there exists $x{:}E \in \Gamma$ such that $\cap_{\underline{n}}(A_i{\to}B_i) = E{\cap}(C{\to}D)$, so $A_h{\to}B_h = C{\to}D$ for some $h \in \underline{n}$. Then we can construct:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_2\\ Q :\cdot\ \Gamma \vdash \gamma{:}C,\Delta\end{array} \quad \begin{array}{c}\mathcal{D}_1^h\\ P :\cdot\ \Gamma,y{:}A_h \vdash \beta{:}B_h,\Delta\end{array}}{Q\widehat{\gamma}\dagger\widehat{y}P :\cdot\ \Gamma \vdash \beta{:}B_h,\Delta}\,(cut) \quad \begin{array}{c}\mathcal{D}_3\\ R :\cdot\ \Gamma,z{:}D \vdash \Delta\end{array}}{(Q\widehat{\gamma}\dagger\widehat{y}P)\widehat{\beta}\dagger\widehat{z}R :\cdot\ \Gamma \vdash \Delta}\,(cut)$$

and

$$\dfrac{\begin{array}{c}\mathcal{D}_2\\ Q :\cdot\ \Gamma \vdash \gamma{:}C,\Delta\end{array} \quad \dfrac{\begin{array}{c}\mathcal{D}_1^h\\ P :\cdot\ \Gamma,y{:}A_h \vdash \beta{:}B_h,\Delta\end{array} \quad \begin{array}{c}\mathcal{D}_3\\ R :\cdot\ \Gamma,z{:}D \vdash \Delta\end{array}}{P\widehat{\beta}\dagger\widehat{z}R :\cdot\ \Gamma,y{:}A_h \vdash \Delta}\,(cut)}{Q\widehat{\gamma}\dagger\widehat{y}(P\widehat{\beta}\dagger\widehat{z}R) :\cdot\ \Gamma \vdash \Delta}\,(cut)$$

or we have the structure:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ P :\cdot\ \Gamma,y{:}A \vdash \beta{:}B,\Delta\end{array}}{\widehat{y}P\widehat{\beta}{\cdot}\alpha :\cdot\ \Gamma \vdash \alpha{:}A{\to}B{\cup}\Delta}\,(\to R) \quad \dfrac{\dfrac{\begin{array}{c}\mathcal{D}_2^j\\ Q :\cdot\ \Gamma \vdash \gamma{:}C_j,\Delta\end{array} \quad \begin{array}{c}\mathcal{D}_3^j\\ R :\cdot\ \Gamma,z{:}D_j \vdash \Delta\end{array}}{Q\widehat{\gamma}[x]\widehat{z}R :\cdot\ \Gamma,x{:}C_j{\to}D_j \vdash \Delta}\,(\to L) \quad (\forall j{\in}\underline{m})}{Q\widehat{\gamma}[x]\widehat{z}R :\cdot\ \Gamma,x{:}\cup_{\underline{m}}(C_j{\to}D_j) \vdash \Delta}\,(\cup L)}{(\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha}\dagger\widehat{x}(Q\widehat{\gamma}[x]\widehat{z}R) :\cdot\ \Gamma \vdash \Delta}\,(cut)$$

and there exists $\alpha{:}E \in \Delta$ such that $(A{\to}B) \cup E = \cup_{\underline{m}}(C_j{\to}D_j)$, so there exist $h \in \underline{m}$ such that $A{\to}B = C_h{\to}D_h$, and we can construct both

$$\dfrac{\dfrac{\begin{array}{c}\boxed{\mathcal{D}_2^h}\\ Q :\cdot \Gamma \vdash \gamma{:}C_h, \Delta\end{array} \quad \begin{array}{c}\boxed{\mathcal{D}_1}\\ P :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta\end{array}}{Q\widehat{\gamma} \dagger \widehat{y}P :\cdot \Gamma \vdash \beta{:}B, \Delta}\,(cut) \quad \begin{array}{c}\boxed{\mathcal{D}_3^h}\\ R :\cdot \Gamma, z{:}D_h \vdash \Delta\end{array}}{(Q\widehat{\gamma} \dagger \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R :\cdot \Gamma \vdash \Delta}\,(cut)$$

and

$$\dfrac{\begin{array}{c}\boxed{\mathcal{D}_2^h}\\ Q :\cdot \Gamma \vdash \gamma{:}C_h, \Delta\end{array} \quad \dfrac{\begin{array}{c}\boxed{\mathcal{D}_1}\\ P :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta\end{array} \quad \begin{array}{c}\boxed{\mathcal{D}_3^h}\\ R :\cdot \Gamma, z{:}D_h \vdash \Delta\end{array}}{P\widehat{\beta} \dagger \widehat{z}R :\cdot \Gamma, y{:}A \vdash \Delta}\,(cut)}{Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta} \dagger \widehat{z}R) :\cdot \Gamma \vdash \Delta}\,(cut)$$

In fact, we have shown in this last case that the application of $(\cap R)$ (or $(\cup L)$) is obsolete, in that the 'other' connector x (α) occurs only once, so does not carry an intersection (union) type. $\square$

We now come to the soundness results we can show for the propagation rules. These basically show that both intersection and union distribute properly for, respectively, right and left propagation. We will see that it is the propagation in the 'other' direction that creates problems.

First, we will now show that left-propagation into a union type is safe (sound).

Theorem 4.3 *If $P\widehat{\alpha} \not\leftarrow \widehat{x}Q$ is typed as follows:*

$$\dfrac{\begin{array}{c}\boxed{\phantom{\mathcal{D}_1}}\\ P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{n}} A_i, \Delta\end{array} \quad \begin{array}{c}\boxed{\phantom{\mathcal{D}_1}}\\ Q :\cdot \Gamma, x{:}\cup_{\underline{n}} A_i \vdash \Delta\end{array}\,(\cup L)}{P\widehat{\alpha} \not\leftarrow \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$

and $P\widehat{\alpha} \not\leftarrow \widehat{x}Q \to R$ (in one step), then $R :\cdot \Gamma \vdash \Delta$.

Proof: We check the various possibilities; we only show the more illustrative cases:

$(\textit{exp-outs}\not\leftarrow) :: (\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha} \not\leftarrow \widehat{x}Q \to (\widehat{y}(P\widehat{\alpha} \not\leftarrow \widehat{x}Q)\widehat{\beta}{\cdot}\gamma)\widehat{\gamma} \dagger \widehat{x}Q$, with γ fresh. We have for the left-hand term:

$$\dfrac{\dfrac{\begin{array}{c}\boxed{\mathcal{D}_1}\\ P :\cdot \Gamma, y{:}A \vdash \beta{:}B, \alpha{:}\cup_{\underline{m}} C_j, \Delta\end{array}}{\widehat{y}P\widehat{\beta}{\cdot}\alpha :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{m}} C_j \cup (A{\to}B), \Delta}\,(\to R) \quad \dfrac{\begin{array}{c}\boxed{\mathcal{D}_2^j}\\ Q :\cdot \Gamma, x{:}C_j \vdash \Delta\ (\forall j{\in}\underline{m})\end{array} \quad \begin{array}{c}\boxed{\mathcal{D}_3}\\ Q :\cdot \Gamma, x{:}A{\to}B \vdash \Delta\end{array}}{Q :\cdot \Gamma, x{:}\cup_{\underline{m}} C_j \cup (A{\to}B) \vdash \Delta}\,(\cup L)}{(\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha} \not\leftarrow \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$

We can construct for the right-hand term:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\boxed{\mathcal{D}_1}\\ P :\cdot \Gamma, y{:}A \vdash \beta{:}B, \alpha{:}\cup_{\underline{m}} C_j, \Delta\end{array} \quad \dfrac{\begin{array}{c}\boxed{\mathcal{D}_2^j}\\ Q :\cdot \Gamma, x{:}C_j \vdash \Delta\ (\forall j{\in}\underline{m})\end{array}}{Q :\cdot \Gamma, x{:}\cup_{\underline{m}} C_j \vdash \Delta}\,(\cup L)}{P\widehat{\alpha} \not\leftarrow \widehat{x}Q :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta}\,(cut)}{\widehat{y}(P\widehat{\alpha} \not\leftarrow \widehat{x}Q)\widehat{\beta}{\cdot}\gamma :\cdot \Gamma \vdash \gamma{:}A{\to}B, \Delta}\,(\to R) \quad \begin{array}{c}\boxed{\mathcal{D}_3}\\ Q :\cdot \Gamma, x{:}A{\to}B \vdash \Delta\end{array}}{(\widehat{y}(P\widehat{\alpha} \not\leftarrow \widehat{x}Q)\widehat{\beta}{\cdot}\gamma)\widehat{\gamma} \dagger \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$

$(imp\nearrow) :: (P\widehat{\beta}[z]\widehat{y}Q)\widehat{\alpha}\nearrow\widehat{z}R \to (P\widehat{\alpha}\nearrow\widehat{x}R)\widehat{\beta}\ [z]\ \widehat{y}(Q\widehat{\alpha}\nearrow\widehat{x}R)$. Let $n = k + m$.

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{k}} C_l, \beta{:}A, \Delta \qquad \boxed{\mathcal{D}_2}\quad Q :\cdot \Gamma, y{:}B \vdash \alpha{:}\cup_{\underline{m}} C_j, \Delta}{P\widehat{\beta}[z]\widehat{y}Q :\cdot \Gamma\cap z{:}A\to B \vdash \alpha{:}\cup_{\underline{n}} C_i, \Delta}\,(\to L) \qquad \dfrac{\boxed{\mathcal{D}_3^i}\quad R :\cdot \Gamma, x{:}C_i \vdash \Delta \quad (\forall i \in \underline{n})}{R :\cdot \Gamma, x{:}\cup_{\underline{n}} C_i \vdash \Delta}\,(\cup L)}{(P\widehat{\beta}[z]\widehat{y}Q)\widehat{\alpha}\nearrow\widehat{x}R :\cdot \Gamma\cap z{:}A\to B \vdash \Delta}\,(cut)$$

We can now construct:

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{k}} C_l, \beta{:}A, \Delta \qquad \dfrac{\boxed{\mathcal{D}_3^l}\quad R :\cdot \Gamma, x{:}C_l \vdash \Delta \quad (\forall l \in \underline{k})}{R :\cdot \Gamma, x{:}\cup_{\underline{k}} C_l \vdash \Delta}\,(\cup L)}{P\widehat{\alpha}\nearrow\widehat{x}R :\cdot \Gamma \vdash \beta{:}A, \Delta}\,(cut) \qquad \dfrac{\boxed{\mathcal{D}_2}\quad Q :\cdot \Gamma, y{:}B \vdash \alpha{:}\cup_{\underline{m}} C_j, \Delta \qquad \dfrac{\boxed{\mathcal{D}_3^j}\quad R :\cdot \Gamma, x{:}C_j \vdash \Delta \quad (\forall j \in \underline{m})}{R :\cdot \Gamma, x{:}\cup_{\underline{m}} C_j \vdash \Delta}\,(\cup L)}{Q\widehat{\alpha}\nearrow\widehat{x}R :\cdot \Gamma, y{:}B \vdash \Delta}\,(cut)}{(P\widehat{\alpha}\nearrow\widehat{x}R)\widehat{\beta}\ [z]\ \widehat{y}(Q\widehat{\alpha}\nearrow\widehat{x}R) :\cdot \Gamma\cap z{:}A\to B \vdash \Delta}\,(\to L)$$

□

Similarly, we can show that right-propagation into an intersection type is safe (sound).

Theorem 4.4 *If $P\widehat{\alpha}\nwarrow\widehat{x}Q$ is typed as follows:*

$$\dfrac{\boxed{\phantom{\mathcal{D}}}\quad Q :\cdot \Gamma, x{:}\cap_{\underline{n}} A_i \vdash \Delta\,(\cap R) \qquad \boxed{\phantom{\mathcal{D}}}\quad P :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} A_i, \Delta}{P\widehat{\alpha}\nwarrow\widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$

and $P\widehat{\alpha}\nwarrow\widehat{x}Q \to R$ (in one step), then $R :\cdot \Gamma \vdash \Delta$.

Proof: We check some more illustrative cases:
$(\nwarrow exp) :: P\widehat{\alpha}\nwarrow\widehat{x}(\widehat{y}Q\widehat{\beta}\cdot\gamma) \to \widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\cdot\gamma$. For the left-hand term we can derive:

$$\dfrac{\boxed{\mathcal{D}_1}\quad P :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} C_i, \Delta \qquad \dfrac{\boxed{\mathcal{D}_2}\quad Q :\cdot \Gamma, x{:}\cap_{\underline{n}} C_i, y{:}A \vdash \beta{:}B, \Delta}{\widehat{y}Q\widehat{\beta}\cdot\gamma :\cdot \Gamma, x{:}\cap_{\underline{n}} C_i \vdash \gamma{:}A\to B\cup\Delta}\,(\to R)}{P\widehat{\alpha}\nwarrow\widehat{x}(\widehat{y}Q\widehat{\beta}\cdot\gamma) :\cdot \Gamma \vdash \gamma{:}A\to B\cup\Delta}\,(cut)$$

Then we can construct:

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad P :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} C_i, \Delta \qquad \boxed{\mathcal{D}_2}\quad Q :\cdot \Gamma, x{:}\cap_{\underline{n}} C_i, y{:}A \vdash \beta{:}B, \Delta}{P\widehat{\alpha}\nwarrow\widehat{x}Q :\cdot \Gamma, y{:}A \vdash \beta{:}B, \Delta}\,(cut)}{\widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\cdot\gamma :\cdot \Gamma \vdash \gamma{:}A\to B\cup\Delta}\,(\to R)$$

$(\nwarrow imp\text{-}outs) :: P\widehat{\alpha}\nwarrow\widehat{x}(Q\widehat{\beta}[x]\widehat{y}R) \to P\widehat{\alpha}\dagger\widehat{v}((P\widehat{\alpha}\nwarrow\widehat{x}Q)\widehat{\beta}\ [v]\ \widehat{y}(P\widehat{\alpha}\nwarrow\widehat{x}R))$, with v fresh. We have:

$$\dfrac{\boxed{\mathcal{D}_1}\quad P :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} C_i, \Delta\,(\cap R) \qquad \dfrac{\boxed{\mathcal{D}_2}\quad Q :\cdot \Gamma, x{:}B \vdash \beta{:}C, \Delta \qquad \boxed{\mathcal{D}_3}\quad R :\cdot \Gamma, y{:}D, x{:}B \vdash \Delta}{Q\widehat{\beta}[x]\widehat{y}R :\cdot \Gamma, x{:}B\cap(C\to D) \vdash \Delta}\,(\to L)}{P\widehat{\alpha}\nwarrow\widehat{x}(Q\widehat{\beta}[x]\widehat{y}R) :\cdot \Gamma \vdash \Delta}\,(cut)$$

so $\cap_{\underline{n}} A_i = B \cap (C \to D)$; let $A_h = C \to D$. Then we can construct:

$$\dfrac{\dfrac{\mathcal{D}_1^h}{P :\cdot \Gamma \vdash \alpha{:}C \to D, \Delta} \qquad \dfrac{\dfrac{\dfrac{\mathcal{D}_1^i}{P :\cdot \Gamma \vdash \alpha{:}A_i, \Delta\ (\forall i \neq h \in \underline{n})}}{P :\cdot \Gamma \vdash \alpha{:}B, \Delta} (\cap R) \quad \begin{array}{c}\mathcal{D}_2\\ Q :\cdot \Gamma, x{:}B \vdash \beta{:}C, \Delta\\ \vdots\end{array}}{P\widehat{\alpha} \nwarrow \widehat{x} Q :\cdot \Gamma \vdash \beta{:}C, \Delta}(cut) \quad \dfrac{\dfrac{\mathcal{D}_1^i}{P :\cdot \Gamma \vdash \alpha{:}A_i, \Delta\ (\forall i \neq h \in \underline{n})}}{P :\cdot \Gamma \vdash \alpha{:}B, \Delta} (\cap R) \quad \begin{array}{c}\mathcal{D}_3\\ R :\cdot \Gamma, y{:}D, x{:}B \vdash \Delta\\ \vdots\end{array}}{P\widehat{\alpha} \nwarrow \widehat{x} R :\cdot \Gamma, y{:}D \vdash \Delta}(cut)}{(P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta}\,[v]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x} R) :\cdot \Gamma, v{:}C \to D \vdash \Delta}(imp)}{P\widehat{\alpha} \dagger \widehat{v}((P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta}\,[v]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x} R)) :\cdot \Gamma \vdash \Delta}(cut)$$

□

5 The problem with witness reduction

As in the system of [14] defined for the λ-calculus, we suffer loss of the subject reduction property (here called witness reduction), as will be shown by the four examples in this section. Analysis of these examples will lead to the definition of *two* restrictions on the notions of type assignment, $\vdash_{\mathrm{N}}$ and $\vdash_{\mathrm{V}}$, that we will show to be closed under reduction for, respectively, Call-By-Name and Call-By-Value reduction.

First we give a counter example for the full witness reduction result by presenting a reducible, typeable term, for which we cannot type the CBV-contractum using the same contexts:

Example 5.1 Take

$$\begin{array}{rcl} P & = & (\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}\langle v{\cdot}\alpha\rangle)\widehat{\alpha} \dagger \widehat{y}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle), \\ Q & = & \langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}(\langle v{\cdot}\delta\rangle\,\widehat{\delta}\,[v]\,\widehat{w}\langle w{\cdot}\beta\rangle) \end{array}$$

We can type P as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\langle x{\cdot}\gamma\rangle :\cdot x{:}A \vdash \gamma{:}A \quad \langle v{\cdot}\alpha\rangle :\cdot v{:}C \vdash \alpha{:}C}{\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}\langle v{\cdot}\alpha\rangle :\cdot x{:}A \cap (A \to C) \vdash \alpha{:}C}(\to L) \quad \dfrac{\langle x{\cdot}\gamma\rangle :\cdot x{:}A \vdash \gamma{:}A \quad \langle v{\cdot}\alpha\rangle :\cdot v{:}C \to D \vdash \alpha{:}C \to D}{\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}\langle v{\cdot}\alpha\rangle :\cdot x{:}A \cap (A \to C \to D) \vdash \alpha{:}C \to D}(\to L)}{\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}\langle v{\cdot}\alpha\rangle :\cdot x{:}A \cap (A \to C) \cap (A \to C \to D) \vdash \alpha{:}C \cap (C \to D)}(\cap R) \quad \dfrac{\langle y{\cdot}\delta\rangle :\cdot y{:}C \vdash \delta{:}C \quad \langle w{\cdot}\beta\rangle :\cdot w{:}D \vdash \beta{:}D}{\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle :\cdot y{:}C \cap (C \to D) \vdash \beta{:}D}(\to L)}{(\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}\langle v{\cdot}\alpha\rangle)\widehat{\alpha} \dagger \widehat{y}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle) :\cdot x{:}A \cap (A \to C) \cap (A \to C \to D) \vdash \beta{:}D}(cut)$$

Notice that, in step $(\to L)$, the type $C \to D$ is added for y to the type C that was used to type the *capsule* $\langle y{\cdot}\delta\rangle$.

Now the cut in P can be activated in *two* directions, both to the left and to the right. The CBN-reduction creates no difficulties; we consider going left via a CBV-reduction, reducing P to Q:

$$\begin{array}{ll} (\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}\langle v{\cdot}\alpha\rangle)\widehat{\alpha} \dagger \widehat{y}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle) & \to\ (a\not\nwarrow) \\ (\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}\langle v{\cdot}\alpha\rangle)\widehat{\alpha} \not\nwarrow \widehat{y}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle) & \to\ (imp\not\nwarrow) \\ (\langle x{\cdot}\gamma\rangle\,\widehat{\alpha} \not\nwarrow \widehat{y}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle))\,\widehat{\gamma}\,[x]\,\widehat{v}(\langle v{\cdot}\alpha\rangle\,\widehat{\alpha} \not\nwarrow \widehat{y}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle)) & \to\ (cap\not\nwarrow) \\ \langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}(\langle v{\cdot}\alpha\rangle\,\widehat{\alpha} \not\nwarrow \widehat{y}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle)) & \to \\ & (d\not\nwarrow), (\nwarrow a), (\nwarrow imp\text{-}ins), (\nwarrow d), (cap), (\nwarrow cap), (med) \\ \langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}(\langle v{\cdot}\delta\rangle\,\widehat{\delta}\,[v]\,\widehat{w}\langle w{\cdot}\beta\rangle) & \end{array}$$

We cannot derive P's contexts for Q, so cannot derive:

$$\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[v]\,\widehat{w}\langle w{\cdot}\beta\rangle) :\cdot x{:}A \cap (A \to C) \cap (A \to C \to D) \vdash \beta{:}D$$

We can at most achieve:

$$\dfrac{\dfrac{}{\langle x{\cdot}\gamma\rangle :\cdot\ x{:}A \vdash \gamma{:}A} \qquad \dfrac{\dfrac{}{\langle y{\cdot}\delta\rangle :\cdot\ y{:}C \vdash \delta{:}C} \qquad \dfrac{}{\langle w{\cdot}\beta\rangle :\cdot\ w{:}D \vdash \beta{:}D}}{\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[y]\,\widehat{w}\langle w{\cdot}\beta\rangle :\cdot\ y{:}C{\cap}(C{\to}D) \vdash \beta{:}D}\,(\to L)}{\langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\,[x]\,\widehat{v}(\langle y{\cdot}\delta\rangle\,\widehat{\delta}\,[v]\,\widehat{w}\langle w{\cdot}\beta\rangle) :\cdot\ x{:}A{\cap}(A{\to}C{\cap}(C{\to}D)) \vdash \beta{:}D}\,(\to L)$$

Notice that the outermost *import* forces the construction of the type $A{\to}C{\cap}(C{\to}D)$[7], which should be amongst the types A, $A{\to}C$ or $A{\to}C{\to}D$; this is not the case.[8]

Also using union types we can create a counter example, but now for CBN-reduction.

Example 5.2 Similarly, take

$$\begin{array}{rcl} P & = & (\widehat{x}\langle x{\cdot}\delta\rangle\,\widehat{\beta}{\cdot}\delta)\,\widehat{\delta}\dagger\widehat{z}(\widehat{v}\langle z{\cdot}\alpha\rangle\,\widehat{\alpha}{\cdot}\gamma) \\ Q & = & \widehat{v}(\widehat{x}\langle x{\cdot}\alpha\rangle\,\widehat{\beta}{\cdot}\alpha)\,\widehat{\alpha}{\cdot}\gamma \end{array}$$

Then for P we can derive:

$$\dfrac{\dfrac{\dfrac{}{\langle x{\cdot}\delta\rangle :\cdot\ x{:}A \vdash \delta{:}A,\beta{:}B}}{\widehat{x}\langle x{\cdot}\delta\rangle\,\widehat{\beta}{\cdot}\delta :\cdot\ \vdash \delta{:}A{\cup}(A{\to}B)}\,(\to R) \qquad \dfrac{\dfrac{\dfrac{}{\langle z{\cdot}\alpha\rangle :\cdot\ z{:}A,v{:}C \vdash \alpha{:}A}}{\widehat{v}\langle z{\cdot}\alpha\rangle\,\widehat{\alpha}{\cdot}\gamma :\cdot\ z{:}A \vdash \gamma{:}C{\to}A}\,(\to R) \quad \dfrac{\dfrac{}{\langle z{\cdot}\alpha\rangle :\cdot\ z{:}A{\to}B,v{:}C \vdash \alpha{:}A{\to}B}}{\widehat{v}\langle z{\cdot}\alpha\rangle\,\widehat{\alpha}{\cdot}\gamma :\cdot\ z{:}A{\to}B \vdash \gamma{:}C{\to}A{\to}B}\,(\to R)}{\widehat{v}\langle z{\cdot}\alpha\rangle\,\widehat{\alpha}{\cdot}\gamma :\cdot\ z{:}A{\cup}(A{\to}B) \vdash \gamma{:}(C{\to}A){\cup}(C{\to}A{\to}B)}\,(\cup L)}{(\widehat{x}\langle x{\cdot}\delta\rangle\,\widehat{\beta}{\cdot}\delta)\,\widehat{\delta}\dagger\widehat{z}(\widehat{v}\langle z{\cdot}\alpha\rangle\,\widehat{\alpha}{\cdot}\gamma) :\cdot\ \vdash \gamma{:}(C{\to}A){\cup}(C{\to}A{\to}B)}\,(cut)$$

In CBN, P reduces to Q:

$$\begin{array}{lcl} (\widehat{x}\langle x{\cdot}\delta\rangle\,\widehat{\beta}{\cdot}\delta)\,\widehat{\delta}\dagger\widehat{z}(\widehat{v}\langle z{\cdot}\alpha\rangle\,\widehat{\alpha}{\cdot}\gamma) & \to & (\nwarrow a),(\nwarrow exp) \\ \widehat{v}((\widehat{x}\langle x{\cdot}\delta\rangle\,\widehat{\beta}{\cdot}\delta)\,\widehat{\delta}\nwarrow\widehat{z}\langle z{\cdot}\alpha\rangle)\,\widehat{\alpha}{\cdot}\gamma & \to & (\nwarrow d),(a\nearrow) \\ \widehat{v}((\widehat{x}\langle x{\cdot}\delta\rangle\,\widehat{\beta}{\cdot}\delta)\,\widehat{\delta}\nearrow\widehat{z}\langle z{\cdot}\alpha\rangle)\,\widehat{\alpha}{\cdot}\gamma & \to & (exp\text{-}outs\nearrow) \\ \widehat{v}((\widehat{x}(\langle x{\cdot}\delta\rangle\,\widehat{\delta}\nearrow\widehat{z}\langle z{\cdot}\alpha\rangle)\,\widehat{\beta}{\cdot}\epsilon)\,\widehat{\epsilon}\dagger\widehat{z}\langle z{\cdot}\alpha\rangle)\,\widehat{\alpha}{\cdot}\gamma & \to & (d\nearrow),(cap),(exp) \\ \widehat{v}(\widehat{x}\langle x{\cdot}\alpha\rangle\,\widehat{\beta}{\cdot}\alpha)\,\widehat{\alpha}{\cdot}\gamma & & \end{array}$$

We cannot derive P's typeing for Q, so cannot derive:

$$\widehat{v}(\widehat{x}\langle x{\cdot}\alpha\rangle\,\widehat{\beta}{\cdot}\alpha)\,\widehat{\alpha}{\cdot}\gamma :\cdot\ \vdash \gamma{:}(C{\to}A){\cup}(C{\to}A{\to}B)$$

We can only manage to show:

$$\dfrac{\dfrac{\dfrac{}{\langle x{\cdot}\delta\rangle :\cdot\ v{:}C,x{:}A \vdash \delta{:}A,\beta{:}B}}{\widehat{x}\langle x{\cdot}\alpha\rangle\,\widehat{\beta}{\cdot}\alpha :\cdot\ v{:}C \vdash \alpha{:}A{\cup}(A{\to}B)}\,(\to R)}{\widehat{v}(\widehat{x}\langle x{\cdot}\alpha\rangle\,\widehat{\beta}{\cdot}\alpha)\,\widehat{\alpha}{\cdot}\gamma :\cdot\ \vdash \gamma{:}C{\to}A{\cup}(A{\to}B)}\,(\to R)$$

Notice that we would need to transform (weaken) $C{\to}A{\cup}(A{\to}B)$ into $(C{\to}A){\cup}(C{\to}A{\to}B)$ to close this case, but cannot show that this step is admissible.

These two examples show that it is not possible to obtain a general witness reduction result; we can only partially recover from that, as we will show in the next section.

To highlight the problem, focussing on two of the propagation rules, it becomes evident that subject reduction cannot always be shown:

---

[7]This example also shows elegantly that using strict types (no intersection on the right-hand side of arrows) would not suffice for $\mathcal{X}$.

[8]Even if we extend the system with a contra-variant $\leq$, which would solve this case, we can still give counter examples for soundness. Since such a relation is not needed for completeness, we have not considered it here.

Example 5.3 Take rule $(cut\not{>}) : (P\widehat{\beta} \dagger \widehat{y}Q)\widehat{\alpha} \not{>} \widehat{x}R \to (P\widehat{\alpha} \not{>} \widehat{z}R)\widehat{\beta} \dagger \widehat{y}(Q\widehat{\alpha} \not{>} \widehat{x}R)$. Suppose the derivation the left-hand term is shaped like:

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1^i}}{P :\cdot \Gamma \vdash \alpha{:}C_i, \beta{:}A_i, \Delta} \quad \dfrac{\boxed{\mathcal{D}_2^i}}{Q :\cdot \Gamma, y{:}A_i \vdash \alpha{:}C_i, \Delta}}{P\widehat{\beta} \dagger \widehat{y}Q :\cdot \Gamma \vdash \alpha{:}C_i, \Delta \quad (\forall i \in \underline{n})}\,(cut)}{P\widehat{\beta} \dagger \widehat{y}Q :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} C_i, \Delta}\,(\cap R) \quad \dfrac{\boxed{\mathcal{D}_3}}{R :\cdot \Gamma, x{:}\cap_{\underline{n}} C_i \vdash \Delta}}{(P\widehat{\beta} \dagger \widehat{y}Q)\widehat{\alpha} \not{>} \widehat{x}R :\cdot \Gamma \vdash \Delta}\,(cut)$$

and assume that all C_i are needed to type R (as in Example 3.15). Then, for the reduct, given the present sub-derivations, we can at most derive:

$$\dfrac{\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1^i}}{P :\cdot \Gamma \vdash \alpha{:}C_i, \beta{:}A_i, \Delta\ (\forall i \in \underline{n})}}{P :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}} C_i, \beta{:}\cup_{\underline{n}} A_i, \Delta}\,(\cap R) \quad \begin{array}{c}\boxed{\mathcal{D}_3}\\ R :\cdot \Gamma, x{:}\cap_{\underline{n}} C_i \vdash \Delta\\ \vdots\end{array}}{P\widehat{\alpha} \not{>} \widehat{x}R :\cdot \Gamma \vdash \beta{:}\cup_{\underline{n}} A_i, \Delta}\,(cut) \quad \dfrac{\dfrac{\boxed{\mathcal{D}_2^i}}{Q :\cdot \Gamma, y{:}A_i \vdash \alpha{:}C_i, \Delta\ (\forall i \in \underline{n})}}{Q :\cdot \Gamma, y{:}\cap_{\underline{n}} A_i \vdash \alpha{:}\cap_{\underline{n}} C_i, \Delta}\,(\cap R) \quad \begin{array}{c}\boxed{\mathcal{D}_3}\\ R :\cdot \Gamma, x{:}\cap_{\underline{n}} C_i \vdash \Delta\\ \vdots\end{array}}{Q\widehat{\alpha} \not{>} \widehat{x}R :\cdot \Gamma, y{:}\cap_{\underline{n}} A_i \vdash \Delta}\,(cut)}{(P\widehat{\alpha} \not{>} \widehat{x}R)\widehat{\beta} \dagger \widehat{y}(Q\widehat{\alpha} \not{>} \widehat{x}R) :\cdot \Gamma \vdash \Delta}\,(?)$$

Notice again that we cannot weaken $\cup_{\underline{n}} A_i$ to $\cap_{\underline{n}} A_i$, so we cannot close the $(?)$ gap to make the last step to finish this derivation.

Example 5.4 Take the rule $(\not{<}cut) : P\widehat{\alpha} \not{<} \widehat{x}(Q\widehat{\beta} \dagger \widehat{y}R) \to (P\widehat{\alpha} \not{<} \widehat{x}Q)\widehat{\beta} \dagger \widehat{y}(P\widehat{\alpha} \not{<} \widehat{z}R)$. Suppose the derivation for the left-hand side is shaped like:

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}}{P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{n}} C_i, \Delta} \quad \dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_2^i}}{Q :\cdot \Gamma, x{:}C_i \vdash \beta{:}A_i, \Delta} \quad \dfrac{\boxed{\mathcal{D}_3^i}}{R :\cdot \Gamma, x{:}C_i, y{:}A_i \vdash \Delta}}{Q\widehat{\beta} \dagger \widehat{y}R :\cdot \Gamma, x{:}C_i \vdash \Delta \quad (\forall i \in \underline{n})}\,(cut)}{Q\widehat{\beta} \dagger \widehat{y}R :\cdot \Gamma, x{:}\cup_{\underline{n}} C_i \vdash \Delta}\,(\cup L)}{P\widehat{\alpha} \not{<} \widehat{x}(Q\widehat{\beta} \dagger \widehat{y}R) :\cdot \Gamma \vdash \Delta}\,(cut)$$

and assume that all C_i are needed to type P (as in Example 3.15). Then, for the reduct, given the present sub-derivations, we can at most derive:

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1}}{P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{n}} C_i, \Delta} \quad \dfrac{\dfrac{\boxed{\mathcal{D}_2^i}}{Q :\cdot \Gamma, x{:}C_i \vdash \beta{:}A_i, \Delta\ (\forall i \in \underline{n})}}{Q :\cdot \Gamma, x{:}\cup_{\underline{n}} C_i \vdash \beta{:}\cup_{\underline{n}} A_i, \Delta}\,(\cup L)}{P\widehat{\alpha} \not{<} \widehat{x}Q :\cdot \Gamma \vdash \beta{:}\cup_{\underline{n}} A_i, \Delta}\,(cut) \quad \dfrac{\begin{array}{c}\boxed{\mathcal{D}_1}\\ P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{n}} C_i, \Delta\\ \vdots\end{array} \quad \dfrac{\boxed{\mathcal{D}_3^i}}{R :\cdot \Gamma, x{:}C_i, y{:}A_i \vdash \Delta\ (\forall i \in \underline{n})}}{R :\cdot \Gamma, x{:}\cup_{\underline{n}} C_i, y{:}\cap_{\underline{n}} A_i \vdash \Delta}\,(\cup L)}{P\widehat{\alpha} \not{<} \widehat{z}R :\cdot \Gamma, y{:}\cap_{\underline{n}} A_i \vdash \Delta}\,(cut)}{(P\widehat{\alpha} \not{<} \widehat{x}Q)\widehat{\beta} \dagger \widehat{y}(P\widehat{\alpha} \not{<} \widehat{z}R) :\cdot \Gamma \vdash \Delta}\,(?)$$

Notice that we cannot weaken $\cup_{\underline{n}} A_i$ to $\cap_{\underline{n}} A_i$, so we again cannot fill the $(?)$ gap.

We could build similar examples also for the other propagation rules $(exp\text{-}outs\not{>})$, $(exp\text{-}ins\not{>})$, $(imp\not{>})$, $(\not{<}exp)$, $(\not{<}imp\text{-}outs)$, and $(\not{<}imp\text{-}ins)$.

So, this notion of type assignment is too liberal to obtain preservation of types under conversion: analysing the problems above, we can summarise them by: both (1) right-propagation into an introduced union, *i.e.* the result of rule $(\cup L)$, and (2) left-propagation into an introduced intersection, the result of rule $(\cap R)$, break the witness reduction property.

In this paper, the solution to the above problem is to, in Section 6, 'flatten' the context assignment rules of Definition 3.7 by defining *two* separate systems, $P :\cdot \Gamma \vdash_{\mathrm{N}} \Delta$ and $P :\cdot \Gamma \vdash_{\mathrm{V}} \Delta$, as in Definition 6.1 and Definition 6.5. The first system will give a notion of context assignment $P :\cdot \Gamma \vdash_{\mathrm{N}} \Delta$ which solves problem "no left propagation into intersection" restricting to CBN reduction, and solves problem "no right propagation into union" by restricting union types to terms that introduce *sockets*; the second system $P :\cdot \Gamma \vdash_{\mathrm{V}} \Delta$ for the CBV restriction limits intersections to terms that introduce *plugs*.

6 Systems with preservation of types under CBN or CBV reduction

In this section, we will try and retrieve the witness reduction property using a restriction of the system proposed in Definition 3.7. The approach we choose here is, in fact, partially inspired by [14], where union types can only be assigned to values. The solutions we present here are, however, very different: we do not need to limit the structure of types, and, for CBN, limit rule $(\cup L)$ to *names*, *i.e.* introduced *sockets*. To avoid left-propagating into an intersection type, first note that we only left-propagate if the *socket* x involved in the cut is introduced: then x occurs only once in the right-hand term, and an intersection for x should be obsolete; after all, the introduced occurrence only needs one type, and the other types can be safely removed. For CBV, we limit intersection types to *values*, *i.e.* terms that introduce *plugs*; this is reminiscent of the limitation in ML of quantification of types to terms that are values [29, 39], and is used also in [42].

We define $P :\cdot \Gamma \vdash_{\mathrm{N}} \Delta$ as in Definition 3.7, where we change the applicability of rule $(\cup L)$, and add a rule to treat left-activated cuts:

Definition 6.1 The context assignment rules for $\vdash_{\mathrm{N}}$ are as in Definition 3.7, but for:

$$(cut):\ \frac{P :\cdot \Gamma_1 \vdash_{\mathrm{N}} \alpha{:}A, \Delta_1 \quad Q :\cdot \Gamma_2, x{:}A \vdash_{\mathrm{N}} \Delta_2}{P\widehat{\alpha} \dagger \widehat{x}Q :\cdot \Gamma_1 \cap \Gamma_2 \vdash_{\mathrm{N}} \Delta_1 \cup \Delta_2}\ (\textit{for inactive and right-activated cuts})$$

$$(\not{\cdot}):\ \frac{P :\cdot \Gamma_1 \vdash_{\mathrm{N}} \alpha{:}A, \Delta_1 \quad Q :\cdot \Gamma_2, x{:}A \vdash_{\mathrm{N}} \Delta_2}{P\widehat{\alpha} \not{\cdot} \widehat{x}Q :\cdot \Gamma_1 \cap \Gamma_2 \vdash_{\mathrm{N}} \Delta_1 \cup \Delta_2}\ (\textit{A not an intersection type, x introduced})$$

$$(\cup L):\ \frac{P :\cdot \Gamma_i, x{:}A_i \vdash_{\mathrm{N}} \Delta_i \quad (\forall i \in \underline{n})}{P :\cdot \cap_{\underline{n}} \Gamma_i, x{:}\cup_{\underline{n}} A_i \vdash_{\mathrm{N}} \cup_{\underline{n}} \Delta_i}\ (n \geq 0,\ \textit{x introduced in P})$$

Remark 6.2 Before coming to the proof that the system is closed under reduction, we can easily verify that this notion of type assignment is *not closed* for witness expansion. This is clear from the fact that the side-condition of rule $(\cup L)$ is not preserved by witness expansion: take $(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x\cdot\gamma\rangle$ such that α is introduced, and γ does not appear in P, then γ is introduced in the term that is the result of contracting this cut, $\widehat{y}P\widehat{\beta}\cdot\gamma$, but not in the term $(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x\cdot\gamma\rangle$ itself.

We will see that the addition of rule $(\not{\cdot})$ solves the problem of 'left propagation into intersection' in the context of CBN reduction, and that the restriction on rule $(\cup L)$ solves 'right propagation into union'.

Below, we need the following property.

Lemma 6.3 *If $\mathcal{D} :: P :\cdot \Gamma, x{:}\cap_{\underline{n}} A_i \vdash_{\mathrm{N}} \Delta$ is a proper derivation, and x is introduced in P, then, for some $j \in \underline{n}$, $P :\cdot \Gamma, x{:}A_j \vdash_{\mathrm{N}} \Delta$.*

Proof: We look at the two cases:

$P = \langle x\cdot\alpha\rangle ::$ then there exists Γ', Δ' and a proper A such that $\Gamma' \cap x{:}A = \Gamma, x{:}\cap_{\underline{n}} A_i$ and $\Delta = \alpha{:}A \cup \Delta'$. But then $A = A_j$, for some $j \in \underline{n}$, and, by rule (Ax), also $\langle x\cdot\alpha\rangle :\cdot \Gamma', x{:}A \vdash_{\mathrm{N}} \alpha{:}A \cup \Delta'$.

$P = Q\widehat{\alpha}\,[x]\,\widehat{y}R$ :: there exists $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2, C, D$ such that $Q :\cdot \Gamma_1 \vdash_{\mathrm{N}} \alpha{:}C, \Delta_1$, $R :\cdot \Gamma_2, y{:}D \vdash_{\mathrm{N}} \Delta_2$ and $\Gamma_1 \cap \Gamma_2 \cap x{:}C{\to}D = \Gamma, x{:}\cap_{\underline{n}} A_i$, so there exists $x{:}B \in \Gamma_1 \cap \Gamma_2$ such that $B \cap (C{\to}D) = \cap_{\underline{n}} A_i$, so there is a $j \in \underline{n}$ such that $A_j = C{\to}D$, and $\Delta = \Delta_1 \cup \Delta_2$. By thinning, we have both $Q :\cdot \Gamma_1 \backslash x \vdash_{\mathrm{N}} \alpha{:}C, \Delta_1$ and $R :\cdot \Gamma_2 \backslash x, y{:}D \vdash_{\mathrm{N}} \Delta_2$, so also $P :\cdot \Gamma \backslash x, C{\to}D \vdash_{\mathrm{N}} \Delta$. □

Notice that we cannot generalise this by limiting to terms that are 'linear' in x (*i.e.* have only one occurrence of x), since x could then appear in more than one sub-derivation, e.g. in $(\cap R)$, so could have a true intersection type.

Using this lemma, we can now show a first soundness result.

Theorem 6.4 (WITNESS REDUCTION FOR $\vdash_{\mathrm{N}}$ WRT CBN) *If $P :\cdot \Gamma \vdash_{\mathrm{N}} \Delta$, and $P \to_{\mathrm{N}} Q$, then $Q :\cdot \Gamma \vdash_{\mathrm{N}} \Delta$.*

Proof: By induction on the definition of $\to_{\mathrm{N}}$, where we focus on the rules; since by Lemma 4.2, 4.3 and 4.4 the only remaining (problematic) cases are right-propagation into a union and left-propagation into an intersection, we will only focus on those.

$(a\not{\cdot}_{\mathrm{N}})$ :: $P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not{\cdot} \widehat{x}Q$, if P does not introduce α, and Q introduces x. Then there exists C such that both $P :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}C, \Delta$ and $Q :\cdot \Gamma, x{:}C \vdash_{\mathrm{N}} \Delta$. Now the (proper) derivation is shaped like:

$$\dfrac{\dfrac{\boxed{}}{P :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}C_i, \Delta}\ (\forall i \in \underline{n})}{P :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}\cap_{\underline{n}} C_i, \Delta}\ (\cap R) \qquad \dfrac{\boxed{\mathcal{D}}}{Q :\cdot \Gamma, x{:}\cap_{\underline{n}} C_i \vdash_{\mathrm{N}} \Delta}}{P\widehat{\alpha} \dagger \widehat{x}Q :\cdot \Gamma \vdash_{\mathrm{N}} \Delta}\ (cut)$$

(with $n \geq 0$) and $\mathcal{D}$ is proper. Since x is introduced in Q, by Lemma 6.3 $Q :\cdot \Gamma, x{:}C_j \vdash_{\mathrm{N}} \Delta$ for some $j \in \underline{n}$; since $P :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}C_j, \Delta$ is part of the derivation on the left-hand side, we can derive:

$$\dfrac{\boxed{} \qquad \boxed{}}{P :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}C_j, \Delta \qquad Q :\cdot \Gamma, x{:}C_j \vdash_{\mathrm{N}} \Delta}\ (\not{\cdot})$$

Notice that now the left-activated cut is typed correctly, since the side-condition to rule $(\not{\cdot})$ is satisfied.

$(exp\text{-}outs\not{\cdot})$ :: $(\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha} \not{\cdot} \widehat{x}Q \to (\widehat{y}(P\widehat{\alpha} \not{\cdot} \widehat{x}Q)\widehat{\beta}{\cdot}\gamma)\widehat{\gamma} \dagger \widehat{x}Q$, with γ fresh. Then the derivation for the left-hand side is shaped as:

$$\dfrac{\dfrac{\boxed{}}{P :\cdot \Gamma, y{:}A \vdash_{\mathrm{N}} \beta{:}B, \alpha{:}\cup_{\underline{n}} C_i, \Delta}\ (\to R)}{\widehat{y}P\widehat{\beta}{\cdot}\alpha :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}(A{\to}B) \cup \cup_{\underline{n}} C_i, \Delta} \qquad \dfrac{\boxed{} \qquad \boxed{}}{Q :\cdot \Gamma, x{:}(A{\to}B) \cup \cup_{\underline{n}} C_i \vdash_{\mathrm{N}} \Delta}\ (\forall i \in \underline{n})\ (\cup L)}{(\widehat{y}P\widehat{\beta}{\cdot}\alpha)\widehat{\alpha} \not{\cdot} \widehat{x}Q :\cdot \Gamma \vdash_{\mathrm{N}} \Delta}\ (\not{\cdot})$$

(Since the type involved in the cut is not an intersection type, the left-hand sub-derivation

does not end with $(\cap R)$.) For the right-hand side, we can now derive:

$$\dfrac{\dfrac{\dfrac{\boxed{\phantom{P :\cdot \Gamma, y{:}A \vdash_{\mathrm{N}} \beta{:}B, \alpha{:}\cup_{\underline{n}} C_i, \Delta}}}{P :\cdot \Gamma, y{:}A \vdash_{\mathrm{N}} \beta{:}B, \alpha{:}\cup_{\underline{n}} C_i, \Delta} \quad \dfrac{\boxed{\phantom{Q :\cdot \Gamma, x{:}C_i \vdash_{\mathrm{N}} \Delta \ (\forall i \in \underline{n})}}}{Q :\cdot \Gamma, x{:}\cup_{\underline{n}} C_i \vdash_{\mathrm{N}} \Delta}\,(\cup L)}{P\widehat{\alpha} \not\nearrow \widehat{x} Q :\cdot \Gamma, y{:}A \vdash_{\mathrm{N}} \beta{:}B, \Delta}\,(\not\nearrow)}{\widehat{y}(P\widehat{\alpha} \not\nearrow \widehat{x} Q)\widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}A \to B, \Delta}\,(\to R) \quad \boxed{\phantom{Q :\cdot \Gamma, x{:}A \to B \vdash_{\mathrm{N}} \Delta}}}{(\widehat{y}(P\widehat{\alpha} \not\nearrow \widehat{x} Q)\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x} Q :\cdot \Gamma \vdash_{\mathrm{N}} \Delta}\,(cut)$$

$(exp\text{-}ins\not\nearrow)$, $(imp\not\nearrow)$, $(cut\not\nearrow)$ :: Much as in the previous part.

$(\nwarrow exp)$ :: $P\widehat{\alpha} \nwarrow \widehat{x}(\widehat{y} Q \widehat{\beta}\cdot\gamma) \to \widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta}\cdot\gamma$. Notice that $\widehat{y} Q \widehat{\beta}\cdot\gamma$ cannot be typed with a union type, since it does not introduce x; this part concludes much in the style of the previous parts.

$(\nwarrow imp\text{-}outs)$ :: $P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta}[x]\widehat{y} R) \to P\widehat{\alpha} \dagger \widehat{v}((P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta}\,[v]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x} R))$, with v fresh. Assume $Q\widehat{\beta}[x]\widehat{y} R$ is typed with a union type; then x is introduced, so does not occur in either Q or R, and we have (with $n \geq 0$):

$$\dfrac{\boxed{\phantom{\mathcal{D}_1}}\ P :\cdot \Gamma \vdash \alpha{:}\cup_{\underline{n}}(A_i \to B_i), \Delta \quad \dfrac{\dfrac{\boxed{\mathcal{D}_2^i}\ Q :\cdot \Gamma \vdash_{\mathrm{N}} \beta{:}A_i, \Delta \quad \boxed{\mathcal{D}_3^i}\ R :\cdot \Gamma, y{:}B_i \vdash_{\mathrm{N}} \Delta}{Q\widehat{\beta}[x]\widehat{y} R :\cdot \Gamma, x{:}A_i \to B_i \vdash \Delta}\,(imp) \quad (\forall i \in \underline{n})}{Q\widehat{\beta}[x]\widehat{y} R :\cdot \Gamma, x{:}\cup_{\underline{n}}(A_i \to B_i) \vdash_{\mathrm{N}} \Delta}\,(\cup L)}{P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta}[x]\widehat{y} R) :\cdot \Gamma \vdash \Delta}\,(cut)$$

By weakening, also $Q :\cdot \Gamma, x{:}\top \vdash_{\mathrm{N}} \beta{:}A_i, \Delta$ and $R :\cdot \Gamma, x{:}\top, y{:}B_i \vdash_{\mathrm{N}} \beta{:}A_i, \Delta$. We can then construct:

$$\dfrac{\boxed{\mathcal{D}_1}\ P :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}\cup_{\underline{n}}(A_i \to B_i), \Delta \quad \dfrac{\dfrac{\dfrac{}{P :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}\top, \Delta}\,(\top) \quad \boxed{\phantom{Q :\cdot \Gamma, x{:}\top \vdash_{\mathrm{N}} \beta{:}A_i, \Delta}}}{P\widehat{\alpha} \nwarrow \widehat{x} Q :\cdot \Gamma \vdash_{\mathrm{N}} \beta{:}A_i, \Delta}\,(cut) \quad \dfrac{\dfrac{\dfrac{}{P :\cdot \Gamma \vdash_{\mathrm{N}} \alpha{:}\top, \Delta}\,(\top) \quad \vdots \quad \boxed{\phantom{R :\cdot \Gamma, x{:}\top, y{:}B_i \vdash_{\mathrm{N}} \beta{:}A_i, \Delta}}}{P\widehat{\alpha} \nwarrow \widehat{x} R :\cdot \Gamma, y{:}B_i \vdash_{\mathrm{N}} \Delta}\,(cut)}{(P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta}\,[v]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x} R) :\cdot \Gamma, v{:}A_i \to B_i \vdash_{\mathrm{N}} \Delta}\,(imp) \quad (\forall i \in \underline{n})}{(P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta}\,[v]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x} R) :\cdot \Gamma, v{:}\cup_{\underline{n}}(A_i \to B_i) \vdash_{\mathrm{N}} \Delta}\,(\cup L)}{P\widehat{\alpha} \dagger \widehat{v}((P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta}\,[v]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x} R)) :\cdot \Gamma \vdash \Delta}$$

$(\nwarrow imp\text{-}ins)$ :: $P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta}[z]\widehat{y} R) \to (P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta}\,[z]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{z} R)$, $x \neq z$. Notice that $Q\widehat{\beta}[z]\widehat{y} R$ in the left-hand term is not typed using $(\cup L)$ towards x, since it does not introduce x. So the only possibility is that the cut propagates towards an intersection, which is safe.

$(\nwarrow cut)$ :: $P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta} \dagger \widehat{y} R) \to (P\widehat{\alpha} \nwarrow \widehat{x} Q)\widehat{\beta} \dagger \widehat{y}(P\widehat{\alpha} \nwarrow \widehat{z} R)$. As above. □

To conclude, we will now define a notion of context assignment that will prove to be closed to reduction with respect to CBV reduction. Since the definition is in idea and concept entirely dual to the restriction for CBN defined above, we will just focus on the differences.

Definition 6.5 The context assignment rules for $\vdash_{\mathrm{V}}$ are as those in Definition 3.7, except for:

$$(cut):\ \frac{P :\cdot \Gamma_1 \vdash_{\mathrm{V}} \alpha{:}A, \Delta_1 \quad Q :\cdot \Gamma_2, x{:}A \vdash_{\mathrm{V}} \Delta_2}{P\widehat{\alpha} \dagger \widehat{x}Q :\cdot \Gamma_1 \cap \Gamma_2 \vdash_{\mathrm{V}} \Delta_1 \cup \Delta_2}\ (\textit{for inactive and left-activated cuts})$$

$$(\nwarrow):\ \frac{P :\cdot \Gamma_1 \vdash_{\mathrm{V}} \alpha{:}A, \Delta_1 \quad Q :\cdot \Gamma_2, x{:}A \vdash_{\mathrm{V}} \Delta_2}{P\widehat{\alpha} \nwarrow \widehat{x}Q :\cdot \Gamma_1 \cap \Gamma_2 \vdash_{\mathrm{V}} \Delta_1 \cup \Delta_2}\ (A \textit{ not a union type, } \alpha \textit{ introduced})$$

$$(\cap R):\ \frac{P :\cdot \Gamma_i \vdash_{\mathrm{V}} \alpha{:}A_i, \Delta_i \quad (\forall i \in \underline{n})}{P :\cdot \cap_{\underline{n}} \Gamma_i \vdash_{\mathrm{V}} \alpha{:}\cap_{\underline{n}} A_i, \cup_{\underline{n}} \Delta_i}\ (n \geq 0,\ \alpha \textit{ introduced in } P)$$

From the fact that the side-condition of rule $(\cap R)$ is not preserved under witness expansion, we can easily verify that this notion of type assignment is not closed under that operation.

We will see that the addition of rule $(\nwarrow)$ solves the problem of 'right propagation into union' in the context of CBV reduction, and that the restriction on rule $(\cup L)$ solves 'left propagation into intersection'.

Below, we need the following property.

Lemma 6.6 If $\mathcal{D} :: P :\cdot \Gamma \vdash_{\mathrm{V}} \alpha{:}\cup_{\underline{n}} A_i, \Delta$ is proper and α is introduced in P, then $P :\cdot \Gamma \vdash_{\mathrm{V}} \alpha{:}A_j, \Delta$ for some $j \in \underline{n}$.

Proof: We look at the two cases:

$P = \langle x \cdot \alpha \rangle$ :: then there exists Γ', Δ' and proper A such that $\Gamma = \Gamma' \cap x{:}A$ and $\Delta = \alpha{:}A \cup \Delta'$. But then $A = A_j$ for some $j \in \underline{n}$, and, by rule (Ax), also $\langle x \cdot \alpha \rangle :\cdot \Gamma' \cap x{:}A \vdash_{\mathrm{V}} \alpha{:}A, \Delta'$.

$P = \widehat{x}Q\widehat{\beta} \cdot \alpha$ :: there exists Δ', C, D such that $Q :\cdot \Gamma, x{:}C \vdash_{\mathrm{V}} \beta{:}D, \Delta'$, with $C \to D = A_j$ for some $j \in \underline{n}$, and $\Delta' = \alpha{:}\cup_{i \neq j \in \underline{n}} A_i, \Delta$; by thinning, we have $Q :\cdot \Gamma, x{:}C \vdash_{\mathrm{V}} \beta{:}D, \Delta$, so also $\widehat{x}Q\widehat{\delta} \cdot \alpha :\cdot \Gamma \vdash_{\mathrm{V}} \alpha{:}C \to D, \Delta$. □

We conclude the results of this paper by showing the second soundness result.

Theorem 6.7 (WITNESS REDUCTION FOR $\vdash_{\mathrm{V}}$ WRT CBV) *If $P :\cdot \Gamma \vdash_{\mathrm{V}} \Delta$, and $P \to_{\mathrm{V}} Q$, then $Q :\cdot \Gamma \vdash_{\mathrm{V}} \Delta$.*

Proof: We only show the interesting cases.

$(\nwarrow a_{\mathrm{V}})$ :: $P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \nwarrow \widehat{x}Q$, if α introduced in P, and x not introduced in Q. The (proper) derivation for the left-hand term is shaped like:

$$\dfrac{\boxed{\quad \mathcal{D} \quad} \qquad \dfrac{\boxed{\quad\quad} \quad Q :\cdot \Gamma, x{:}C_i \vdash_{\mathrm{V}} \Delta \quad (\forall i \in \underline{n})}{Q :\cdot \Gamma, x{:}\cup_{\underline{n}} C_i \vdash_{\mathrm{V}} \Delta}\ (\cup L)}{P\widehat{\alpha} \dagger \widehat{x}Q :\cdot \Gamma \vdash_{\mathrm{V}} \Delta}\ (cut)$$

and $\mathcal{D}$ is proper. Since α is introduced in P, by Lemma 6.6, $P :\cdot \Gamma \vdash_{\mathrm{V}} \alpha{:}C_j, \Delta$ for some $j \in \underline{n}$; since $Q :\cdot \Gamma, x\alpha{:}C_j \vdash_{\mathrm{V}} \Delta$ is part of the derivation on the left-hand side, we can derive:

$$\dfrac{\boxed{\quad\quad} \quad P :\cdot \Gamma \vdash_{\mathrm{V}} \alpha{:}C_j, \Delta \qquad \boxed{\quad\quad} \quad Q :\cdot \Gamma, x{:}C_j \vdash_{\mathrm{V}} \Delta}{P\widehat{\alpha} \nwarrow \widehat{x}Q :\cdot \Gamma \vdash_{\mathrm{V}} \Delta}\ (\nwarrow)$$

Notice that the right-activated cut is typed correctly, since the side-condition to rule $(\nwarrow)$ is satisfied.

$(\textit{exp-outs}\not\nwarrow)$ :: $(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha}\not\nwarrow\widehat{x}Q \to (\widehat{y}(P\widehat{\alpha}\not\nwarrow\widehat{x}Q)\widehat{\beta}\cdot\gamma)\widehat{\gamma}\dagger\widehat{x}Q$, with γ fresh. Notice that $y,\beta \notin fc(Q)$. Assume $(\cap R)$ has been used for $\widehat{y}P\widehat{\beta}\cdot\alpha$:

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1^i}}{P :\cdot \Gamma, y{:}A_i \vdash \beta{:}B_i, \alpha{:}A_i{\to}B_i, \Delta}}{\widehat{y}P\widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash_{\mathrm{V}} \alpha{:}A_i{\to}B_i, \Delta}\,(\to R) \quad (\forall i{\in}\underline{n})}{\widehat{y}P\widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash \alpha{:}\cap_{\underline{n}}(A_i{\to}B_i), \Delta}\,(\cap R) \qquad \dfrac{\boxed{\mathcal{D}_2}}{Q :\cdot \Gamma, x{:}\cap_{\underline{n}}(A_i{\to}B_i) \vdash_{\mathrm{V}} \Delta}}{(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha}\not\nwarrow\widehat{x}Q :\cdot \Gamma \vdash_{\mathrm{V}} \Delta}\,(cut)$$

Notice that then α is introduced, so, in particular, does not occur in P. Then, by thinning and weakening we can derive $P :\cdot \Gamma, y{:}A_i \vdash \beta{:}B_i, \alpha{:}\bot, \Delta$, and we can construct:

$$\dfrac{\dfrac{\dfrac{\dfrac{\boxed{\phantom{\mathcal{D}_1}}}{P :\cdot \Gamma, y{:}A_i \vdash \beta{:}B_i, \alpha{:}\bot, \Delta} \qquad \dfrac{}{Q :\cdot \Gamma, x{:}\bot \vdash_{\mathrm{V}} \Delta}\,(\bot)}{P\widehat{\alpha}\not\nwarrow\widehat{x}Q :\cdot \Gamma, y{:}A_i \vdash_{\mathrm{V}} \beta{:}B_i, \Delta}\,(cut) \quad (\forall i{\in}\underline{n})}{\widehat{y}(P\widehat{\alpha}\not\nwarrow\widehat{x}Q)\widehat{\beta}\cdot\gamma :\cdot \Gamma \vdash \gamma{:}\cap_{\underline{n}}(A_i{\to}B_i), \Delta}\,(\cap R) \qquad \dfrac{\boxed{\mathcal{D}_2}}{Q :\cdot \Gamma, x{:}\cap_{\underline{n}}(A_i{\to}B_i) \vdash_{\mathrm{V}} \Delta}}{(\widehat{y}(P\widehat{\alpha}\not\nwarrow\widehat{x}Q)\widehat{\beta}\cdot\gamma)\widehat{\gamma}\dagger\widehat{x}Q :\cdot \Gamma \vdash_{\mathrm{V}} \Delta}$$

$(\textit{exp-ins}\not\nwarrow)$ :: $(\widehat{y}P\widehat{\beta}\cdot\gamma)\widehat{\alpha}\not\nwarrow\widehat{x}Q \to \widehat{y}(P\widehat{\alpha}\not\nwarrow\widehat{x}Q)\widehat{\beta}\cdot\gamma$, with $\gamma \neq \alpha$. Notice that $(\cap R)$ has not been used for $\widehat{y}P\widehat{\beta}\cdot\gamma$, since it does not introduce α. Similar for $(\textit{imp}\not\nwarrow)$, and $(\textit{cut}\not\nwarrow)$.

$(\nwarrow\textit{exp})$, $(\nwarrow\textit{imp-outs})$, $(\nwarrow\textit{imp-ins})$, $(\nwarrow\textit{cut})$ :: As above, rule $(\nwarrow)$ excludes that $(\cup L)$ has been used. □

7 Conclusions

We have seen that it is straightforward to define a natural notion of context assignment to the sequent calculus $\mathcal{X}$ that uses intersection and union types. This system was shown natural in that we were able to show that witness expansion - the main reason to use either intersection or union - follows easily, and that both intersection and union play their natural and crucial role in the proof of that property.

However, somewhat surprisingly, this ease is not paired with soundness with respect to witness reduction. As in similar notions for the λ-calculus, combining union and intersection types breaks the soundness of the system. We have isolated the problem cases, and seen that it is exactly the non-logical behaviour of both type constructors that causes the problem. We have looked at a number of restrictions for either CBN or CBV reduction that overcome this defect, but all with the loss of the witness expansion result.

This implies that it is impossible to define a semantics using types for $\mathcal{X}$, even for the two confluent sub-reduction systems.

The problem of strong normalisation is still open.

Acknowledgement

I would like to thank Philippe Audebaud and Mariangiola Dezani for fruitful discussions, and especially thank Vanessa Loprete for valuable support.

References

[1] Abadi, M., Gordon, A.: A Calculus for Cryptographic Protocols: The Spi Calculus, *Proceedings of the Fourth ACM Conference on Computer and Communications Security*, ACM Press, 1997.

[2] Bakel, S. van: Complete restrictions of the Intersection Type Discipline, *Theoretical Computer Science*, **102**(1), 1992, 135–163.

[3] Bakel, S. van: Intersection Type Assignment Systems, *Theoretical Computer Science*, **151**(2), 1995, 385–435.

[4] Bakel, S. van: Cut-Elimination in the Strict Intersection Type Assignment System is Strongly Normalising, *Notre Dame journal of Formal Logic*, **45**(1), 2004, 35–63.

[5] Bakel, S. van: Intersection and Union Types for $\mathcal{X}$, *Electronic Proceedings of 3rd International Workshop* Intersection Types and Related Systems *(ITRS'04), Turku, Finland*, 136, 2004.

[6] Bakel, S. van: Reduction in $\mathcal{X}$ does not agree with Intersection and Union Types, *Electronic Proceedings of 4th International Workshop* Intersection Types and Related Systems *(ITRS'08), Turin, Italy*, 2008.

[7] Bakel, S. van: Completeness and Partial Soundness Results for Intersection & Union Typing for $\overline{\lambda}\mu\tilde{\mu}$, *Annals of Pure and Applied Logic*, **161**, 2010, 1400–1430.

[8] Bakel, S. van, Cardelli, L., Vigliotti, M.: From $\mathcal{X}$ to π; Representing the Classical Sequent Calculus in the π-calculus, *Electronic Proceedings of International Workshop on Classical Logic and Computation 2008* (CL&C'08), *Reykjavik, Iceland*, 2008.

[9] Bakel, S. van, de'Liguoro, U.: Logical equivalence for subtyping object and recursive types, *Theory of Computing Systems*, **42**(3), 2008, 306–348.

[10] Bakel, S. van, Fernández, M.: Normalisation Results for Typeable Rewrite Systems, *Information and Computation*, **2**(133), 1997, 73–116.

[11] Bakel, S. van, Lengrand, S., Lescanne, P.: The language $\mathcal{X}$: Circuits, Computations and Classical Logic, *Proceedings of Ninth Italian Conference on Theoretical Computer Science* (ICTCS'05), *Siena, Italy* (M. Coppo, E. Lodi, G. M. Pinna, Eds.), 3701, Springer Verlag, 2005.

[12] Bakel, S. van, Lescanne, P.: Computation with Classical Sequents, *Mathematical Structures in Computer Science*, **18**, 2008, 555–609.

[13] Barbanera, F., Berardi, S.: A Symmetric Lambda Calculus for Classical Program Extraction, *Information and Computation*, **125**(2), 1996, 103–117.

[14] Barbanera, F., Dezani-Ciancaglini, M., de'Liguoro, U.: Intersection and Union Types: Syntax and Semantics, *Information and Computation*, **119(2)**, 1995, 202–230.

[15] Barendregt, H.: *The Lambda Calculus: its Syntax and Semantics*, Revised edition, North-Holland, Amsterdam, 1984.

[16] Barendregt, H., Coppo, M., Dezani-Ciancaglini, M.: A filter lambda model and the completeness of type assignment, *Journal of Symbolic Logic*, **48**(4), 1983, 931–940.

[17] Bloo, R., Rose, K.: Preservation of Strong Normalisation in Named Lambda Calculi with Explicit Substitution and Garbage Collection, *CSN'95 – Computer Science in the Netherlands*, 1995.

[18] Church, A.: A Note on the Entscheidungsproblem, *Journal of Symbolic Logic*, **1**(1), 1936, 40–41.

[19] Coppo, M., Dezani-Ciancaglini, M.: A New Type Assignment for Lambda-Terms, *Archive für Mathematischer Logic und Grundlagenforschung*, **19**, 1978, 139–156.

[20] Coppo, M., Dezani-Ciancaglini, M., Venneri, B.: Functional characters of solvable terms, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **27**, 1981, 45–58.

[21] Curien, P.-L., Herbelin, H.: The Duality of Computation, *Proceedings of the 5th ACM SIGPLAN International Conference on Functional Programming (ICFP'00)*, 35.9, ACM, 2000.

[22] Curry, H., Feys, R.: *Combinatory Logic*, vol. 1, North-Holland, Amsterdam, 1958.

[23] Davies, R., Pfenning, F.: A judgmental reconstruction of modal logic, *Mathematical Structures in Computer Science*, **11**(4), 2001, 511–540.

[24] Dougherty, D., Ghilezan, S., Lescanne, P.: Intersection and Union Types in the $\overline{\lambda}\mu\tilde{\mu}$-calculus, *Electronic Proceedings of 2nd International Workshop* Intersection Types and Related Systems *(ITRS'04), Turku, Finland*, 136, 2004.

[25] Dougherty, D., Ghilezan, S., Lescanne, P.: Characterizing strong normalization in the Curien-Herbelin symmetric lambda calculus: extending the Coppo-Dezani heritage, *Theoretical Computer Science*, **398**, 2008.

[26] Dougherty, D., Ghilezan, S., Lescanne, P., Likavec, S.: Strong Normalization of the Dual Classical Sequent Calculus, *Proceedings of 12th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'05)*, 3835, 2005.

[27] Dunfield, J., Pfenning, F.: Type Assignment for Intersections and Unions in Call-by-Value Languages, *Proceedings of 6th International Conference on* Foundations of Software Science and Computational Structures *(FOSSACS'03)* (A. D. Gordon, Ed.), 2003.

[28] Gentzen, G.: Investigations into logical deduction, *The Collected Papers of Gerhard Gentzen*, Ed M. E. Szabo, North Holland, 68ff (1969), 1935.

[29] Harper, B., Lillibridge, M.: ML with callcc is unsound, 1991, Post to TYPES mailing list, July 8.

[30] Herbelin, H.: *C'est maintenant qu'on calcule: au cœur de la dualité*, Mémoire d'habilitation, Université Paris 11, Décembre 2005.

[31] Hindley, J.: Coppo-Dezani Types do not Correspond to Propositional Logic, *Theoretical Computer Science*, **28**, 1984, 235–236.

[32] Hindley, J.: *Basic Simple Type Theory*, Cambridge University Press, 1997.

[33] Honda, K., Tokoro, M.: An object calculus for asynchronous communication, *Proceedings of ECOOP'91*, 512, Springer Verlag, 1991.

[34] Kleene, S.: *Introduction to Metamathematics*, North Holland, Amsterdam, 1952.

[35] Krivine, J.-L.: *Lambda calculus, types and models*, Ellis Horwood, 1993.

[36] Lengrand, S.: Call-by-value, call-by-name, and strong normalization for the classical sequent calculus, *3rd Workshop on Reduction Strategies in Rewriting and Programming (WRS 2003)* (B. Gramlich, S. Lucas, Eds.), 86, Elsevier, 2003.

[37] Milner, R.: Functions as processes, *Mathematical Structures in Computer Science*, **2**(2), 1992, 269–310.

[38] Milner, R.: *Communicating and Mobile Systems: the π-calculus*, Cambridge University Press, 1999.

[39] Milner, R., Tofte, M., Harper, R., MacQueen, D.: *The Definition of Standard ML*, MIT Press, 1990, Revised edition.

[40] Parigot, M.: An algorithmic interpretation of classical natural deduction, *Proceedings of 3rd International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'92)*, 624, Springer Verlag, 1992.

[41] Pierce, B.: *Programming with Intersection Types and Bounded Polymorphism*, Ph.D. Thesis, Carnegie Mellon University, School of Computer Science, Pitssburgh, 1991, CMU-CS-91-205.

[42] Summers, A.: *Curry-Howard Term Calculi for Gentzen-Style Classical Logic*, Ph.D. Thesis, Imperial College London, 2008.

[43] Urban, C.: *Classical Logic and Computation*, Ph.D. Thesis, University of Cambridge, October 2000.

[44] Urban, C., Bierman, G.: Strong normalisation of cut-elimination in classical logic, *Fundamenta Informaticae*, **45**(1,2), 2001, 123–155.

[45] Wadler, P.: Call-by-Value is Dual to Call-by-Name, *Proceedings of the eighth ACM SIGPLAN international conference on Functional programming*, 2003.

[46] Wright, A.: Simple imperative polymorphism, *Lisp and Symbolic Computation*, **8**(4), 1995, 343–355.