On the Values of Reducibility Candidates

Colin Riba

Laboratoire de l'Informatique du Parallélisme*
ENS Lyon – Université de Lyon
colin.riba@ens-lyon.fr
http://perso.ens-lyon.fr/colin.riba/

Abstract. The straightforward elimination of union types is known to break subject reduction, and for some extensions of the lambda-calculus, to break strong normalization as well. Similarly, the straightforward elimination of implicit existential types breaks subject reduction.

We propose elimination rules for union types and implicit existential quantification which use a form call-by-value issued from Girard's reducibility candidates. We show that these rules remedy the above mentioned difficulties, for strong normalization and, for the existential quantification, for subject reduction as well.

Moreover, for extensions of the lambda-calculus based on intuitionistic logic, we show that the obtained existential quantification is equivalent to its usual impredicative encoding w.r.t. provability in realizability models built from reducibility candidates and biorthogonals.

1 Introduction

Although useful in some type systems [2, 12], implicit existential types can be problematic because of their elimination rule. Some of these problems seem related to similar issues with union types.

The straightforward elimination of union types is known to break subject reduction [1], and for some extensions of the λ-calculus, to break strong normalization as well [14]. The counter example for subject reduction given in [1] has been adapted to implicit existential types in [18]. Concerning strong normalization, the difficulties are related to type interpretations in reducibility. Usually, types are interpreted by closure operators. Union types and implicit existential types are both interpreted by the closure of the union. When no further assumption is made, the validity of their elimination rules follows from the closure under unions of the interpretations (i.e. when closure preserves unions). But results of [14] on failures of strong normalization with elimination of union types show that some rewrite systems are not compatible with type interpretations which are closed under unions.

These problems with elimination are caused by possibly bad cohabitations between the rules of elimination and the reductions of some calculi. This suggests

[*] UMR 5668 CNRS ENS-Lyon UCBL INRIA, 46 allée d'Italie, 69364 Lyon Cedex 7, France

either to adapt the elimination rules to the calculus, or to adapt the calculus to the elimination rules. In call-by-value settings, the first solution is studied in [6, 19] for systems with union and implicit existential quantification. Both works propose elimination rules restricted to call-by-value evaluation contexts. The second solution is studied in [8] for a call-by-value λ-calculus. On a related subject, a reducibility interpretation of Moggi's computational calculus based on a combination of reducibility candidates and biorthogonals is given in [11].

In this paper, working on rewriting-based extensions of the λ-calculus, we propose to eliminate union and implicit existential types by a $(\mathsf{let}\ x = t\ \mathsf{in}\ c)$ which is reduced according to a form call-by-value issued from Girard's reducibility candidates [7]. In contrast with [8, 6, 19], this does not force the whole calculus to be call-by-value. We build on [15, 16], where a general notion of Girard's candidates is proposed. This framework assumes only a rewrite relation and a set of contexts, called elimination contexts, subject to some axioms. The basic ingredient is an interaction property between terms and elimination contexts. Terms which interact with elimination contexts are called values since they are observable. They are the values that we use in the reduction of the let. Our notion of call-by-value is not the usual one (considered for instance in [8]) since in our case variables are not values. From a theoretical point of view, this allows to define the reduction of $(\mathsf{let}\ x = t\ \mathsf{in}\ c)$ by a rewrite relation. This would have been impossible if variables were values.

We present the basic tools in Sect. 2. The axiomatization of reducibility candidates is presented in Sect. 3. In Sect. 4, we show how to extend modularly our framework with the let and prove that the axioms for reducibility are preserved. Section 5 presents three applications. First, strong normalization with union types and possibly non-deterministic simple rewriting in Sect. 5.1. Second, strong normalization and subject reduction with existential, product and sum types in Sect. 5.2. Third, concerning extensions of the λ-calculus based on intuitionistic logic, we show in Sect. 5.3 that the obtained existential quantification is equivalent to the usual impredicative encoding w.r.t. provability in realizability models built from reducibility candidates and biorthogonals.

A version of this paper with full proofs is available from the webpage of the author.

2 Preliminaries

Terms. A *signature* Σ is a family of sets $(\Sigma_n)_{n\in\mathbb{N}}$ such that Σ_n contains algebraic symbols of arity n. We consider λ-terms with uncurryied symbols f in a signature Σ and variables $x \in \mathcal{X}$:

$$t, u \in \Lambda(\Sigma) \quad ::= \quad x \quad | \quad \lambda x.t \quad | \quad t\,u \quad | \quad \mathsf{f}(t_1, \dots, t_n)\ ,$$

where $\mathsf{f} \in \Sigma_n$. Let Λ be the set of *pure* λ*-terms* $\Lambda(\emptyset)$. A *substitution* is a function $\sigma : \mathcal{X} \to \Lambda(\Sigma)$ of finite domain. The capture-avoiding application of σ to a term t is written $t\sigma$ or $t[\sigma(x_1)/x_1, \dots, \sigma(x_n)/x_n]$ if $\mathcal{D}om(\sigma) = \{x_1, \dots, x_n\}$. We let $\sigma[t/x]$ be the substitution which maps x to t and is equal to σ on $\mathcal{D}om(\sigma) \setminus \{x\}$.

Reductions. A *rewrite relation* is a binary relation $\to_R \subseteq (\Lambda(\Sigma) \setminus \mathcal{X}) \times \Lambda(\Sigma)$ which is closed under contexts and substitutions. Write $\to_{RS}$ for $\to_R \cup \to_S$ and let $(t)_R =_{\text{def}} \{v \mid t \to_R v\}$. Define the *product extension* of $\to_R$ as $(t_1, \dots, t_n) \to_R (u_1, \dots, u_n)$ when there is $k \in \{1, \dots, n\}$ such that $t_k \to_R u_k$ and $t_i = u_i$ for all $i \neq k$. We denote by $\mathcal{SN}_R$ the set of *strongly normalizing* terms for $\to_R$, which is the smallest set such that $\forall t. (\forall u.\ t \to_R u \implies u \in \mathcal{SN}_R) \Rightarrow t \in \mathcal{SN}_R$. Let $\to_\beta$ be the smallest rewrite relation on $\Lambda(\Sigma)$ such that $(\lambda x.t)u \to_\beta t[u/x]$.

The Polymorphic λ-Calculus $\boldsymbol{\lambda_\forall}$. Our core language is the *Curry-style* polymorphic λ-calculus $\boldsymbol{\lambda_\forall}$. Its types are the formulas of second-order minimal logic, with variables $X \in \mathcal{X}_\mathcal{T}$:

$$T, U \in \mathcal{T}_\forall \quad ::= \quad X \in \mathcal{X}_\mathcal{T} \quad | \quad U \Rightarrow T \quad | \quad \forall X.T \ .$$

Its typing rules are the following:

$$(\mathrm{Ax})\ \frac{}{\Gamma, x : T \vdash x : T} \qquad (\Rightarrow \mathrm{I})\ \frac{\Gamma, x : U \vdash t : T}{\Gamma \vdash \lambda x.t : U \Rightarrow T} \qquad (\Rightarrow \mathrm{E})\ \frac{\Gamma \vdash t : U \Rightarrow T \qquad \Gamma \vdash u : U}{\Gamma \vdash tu : T}$$

$$(\forall\,\mathrm{I})\ \frac{\Gamma \vdash t : T}{\Gamma \vdash t : \forall X.T}\ (X \notin \Gamma) \qquad (\forall\,\mathrm{E})\ \frac{\Gamma \vdash t : \forall X.T}{\Gamma \vdash t : T[U/X]}$$

Types without $\forall X.T$ are called *simple types*, denoted by $\mathcal{T}$. The *simply typed λ-calculus* $\boldsymbol{\lambda}$ is $\boldsymbol{\lambda_\forall}$ restricted to $\mathcal{T}$ (hence without the rules $(\forall\,\mathrm{I})$ and $(\forall\,\mathrm{E})$).

Implicit Existential Types. We add to $\boldsymbol{\lambda_\forall}$ *implicit existential types.* Let $\mathcal{T}_{\forall\exists}$ be the extension of $\mathcal{T}_\forall$ with the second-order existential quantification $\exists X.T$. The usual implicit rules for $\exists X.T$ are

$$(\exists\,\mathrm{I})\ \frac{\Gamma \vdash t : T[U/X]}{\Gamma \vdash t : \exists X.T} \qquad (\exists\,\mathrm{E})\ \frac{\Gamma \vdash t : \exists X.T \qquad \Gamma, x : T \vdash c : C}{\Gamma \vdash c[t/x] : C}\ X \notin \Gamma, C$$

As for universal quantification, they are not reflected at the term level by corresponding constructors and eliminators. The rule $(\exists\,\mathrm{E})$ does not satisfy subject reduction [18].

Example 2.1 ([18]). Let $\mathrm{I} = \lambda x.x$, $u_1 = z(\mathrm{I}xy)(\mathrm{I}xy)$, $u_2 = z(xy)(\mathrm{I}xy)$ and let Γ be the context $x : Y \Rightarrow \exists X.X \Rightarrow X, y : Y, z : \forall X.(X \Rightarrow X) \Rightarrow (X \Rightarrow X) \Rightarrow Z$. Using $(\exists\,\mathrm{E})$ we have $\Gamma \vdash u_1 : Z$. But $u_1 \to_\beta u_2$ and u_2 is not of type Z in Γ.

Moreover, $(\exists\,\mathrm{E})$ causes difficulties with strong normalization. To explain them, we introduce some basic notions on reducibility.

Reducibility. Let $\to_R$ be a rewrite relation on $\Lambda(\Sigma)$. In strong normalization proofs based on reducibility, types T are interpreted by sets of strongly normalizing terms $[\![T]\!]$. Strong normalization of typable terms follows from the *adequacy* of the interpretation: typable terms must belong to the interpretation of their type. However, not every type interpretation is adequate. Usually, an adequate interpretation can be described by a *closure operator* on $\mathcal{P}(\mathcal{SN}_{\mathsf{R}})$, i.e. by a function $\mathcal{R}ed(\_) : \mathcal{P}(\mathcal{SN}_{\mathsf{R}}) \to \mathcal{P}(\mathcal{SN}_{\mathsf{R}})$ which is idempotent ($\mathcal{R}ed(\mathcal{R}ed(A)) = \mathcal{R}ed(A)$), extensive ($A \subseteq \mathcal{R}ed(A)$) and monotone ($A \subseteq B \Rightarrow \mathcal{R}ed(A) \subseteq \mathcal{R}ed(B)$). It is well-known that the set of *closed elements* $\mathcal{R}ed =_{\text{def}} \{\mathcal{R}ed(A) \mid A \subseteq \mathcal{SN}_{\mathsf{R}}\}$ is a complete lattice whose g.l.b.'s are given by intersections. We say that $\mathcal{R}ed$ is a *reducibility family* if $\mathcal{X} \subseteq A$ for all $A \in \mathcal{R}ed$. There are essentially three kinds of reducibility families: Tait's saturated sets [17], Girard's reducibility candidates [7] and biorthogonals [13]. In this paper we focus on the last two.

Assume that $\mathcal{R}ed$ is closed under the *function space* $\_ \Rightarrow \_$, which is defined as $A \Rightarrow B \;=_{\text{def}}\; \{t \mid \forall u.\ u \in A \implies t\,u \in B\}$ where $A, B \subseteq \Lambda(\Sigma)$. Given an assignment $\rho : \mathcal{X}_{\mathcal{T}} \to \mathcal{R}ed$, the interpretation $[\![T]\!]^{\mathcal{R}ed}_{\rho} \in \mathcal{R}ed$ of polymorphic types $T \in \mathcal{T}_{\forall}$ is inductively defined as $[\![X]\!]^{\mathcal{R}ed}_{\rho} =_{\text{def}} \rho(X)$,

$$[\![U \Rightarrow T]\!]^{\mathcal{R}ed}_{\rho} \;=_{\text{def}}\; [\![U]\!]^{\mathcal{R}ed}_{\rho} \Rightarrow [\![T]\!]^{\mathcal{R}ed}_{\rho} \quad\text{and}\quad [\![\forall X.T]\!]^{\mathcal{R}ed}_{\rho} \;=_{\text{def}}\; \bigcap_{C \in \mathcal{R}ed} [\![T]\!]^{\mathcal{R}ed}_{\rho[C/X]} \;.$$

We say that $[\![\_]\!]^{\mathcal{R}ed}$ is *adequate* if ($\Gamma \vdash t : T$ and $(\rho, \sigma) \models_{\mathcal{R}ed} \Gamma$) implies $t\sigma \in [\![T]\!]^{\mathcal{R}ed}_{\rho}$, where $(\rho, \sigma) \models_{\mathcal{R}ed} \Gamma$ if $\sigma(x) \in [\![\Gamma(x)]\!]^{\mathcal{R}ed}_{\rho}$ for all $x \in \mathcal{D}om(\Gamma)$. If $[\![\_]\!]^{\mathcal{R}ed}$ is adequate then $\Gamma \vdash t : T$ implies $t \in \mathcal{SN}_{\mathsf{R}}$.

Adequacy and Existential Quantification. Adequacy w.r.t. the rules ($\forall$ I) and ($\forall$ E) is ensured by the definition of $[\![\forall X.T]\!]^{\mathcal{R}ed}_{\rho}$. Dually, one might expect adequacy w.r.t. ($\exists$ I) and ($\exists$ E) for the extension of $[\![\_]\!]^{\mathcal{R}ed}_{\rho}$ to $\mathcal{T}_{\forall\exists}$ defined as

$$[\![\exists X.T]\!]^{\mathcal{R}ed}_{\rho} \quad=_{\text{def}}\quad \mathcal{R}ed\left(\textstyle\bigcup_{C \in \mathcal{R}ed} [\![T]\!]^{\mathcal{R}ed}_{\rho[C/X]}\right) \;.$$

Since $[\![T]\!]^{\mathcal{R}ed}_{\rho[C/X]} \subseteq \mathcal{R}ed(\bigcup_{C \in \mathcal{R}ed} [\![T]\!]^{\mathcal{R}ed}_{\rho[R/X]})$, every reducibility family validates the rule ($\exists$ I). This is less clear for ($\exists$ E). We only know that it is validated by reducibility families which are *closed under unions*, i.e. such that $\mathcal{R}ed(\bigcup \mathcal{A}) = \bigcup \mathcal{A}$ for all $\mathcal{A} \subseteq \mathcal{R}ed$. But closure under unions is not possible for all rewrite relations $\to_{\mathsf{R}}$. This is related to strong normalization problems with union types.

Union Types. Let $\mathcal{T}_{\sqcap\sqcup}$ be the extension of simple types with intersection types $T \sqcap U$ and union types $T \sqcup U$. $\mathcal{T}_{\sqcap\sqcup}$ is equipped with a binary relation $\sqsubseteq$ such that $(\mathcal{T}_{\sqcap\sqcup}, \sqsubseteq)$ is a preorder with all finite l.u.b.'s and g.l.b.'s and satisfying usual additional axioms for the arrow $\Rightarrow$ (see Sect. 5.1). The typing rules are those of $\boldsymbol{\lambda}$ extended with (Sub) and ($\sqcap$ I) (see Sect. 5.1). Intersections and unions are interpreted in $\mathcal{R}ed$ similarly as quantifications: $[\![T \sqcap U]\!]^{\mathcal{R}ed}_{\rho} \;=_{\text{def}}\; [\![T]\!]^{\mathcal{R}ed}_{\rho} \cap [\![U]\!]^{\mathcal{R}ed}_{\rho}$ and $[\![T \sqcup U]\!]^{\mathcal{R}ed}_{\rho} \;=_{\text{def}}\; \mathcal{R}ed([\![T]\!]^{\mathcal{R}ed}_{\rho} \cup [\![U]\!]^{\mathcal{R}ed}_{\rho})$. As with ($\exists$ E), if $\mathcal{R}ed$ is closed under unions, then $[\![\_]\!]^{\mathcal{R}ed}_{\rho}$ is adequate for the elimination of union:

$$(\sqcup\,\mathrm{E})\ \frac{\Gamma \vdash t : T_1 \sqcup T_2 \qquad \Gamma, x : T_1 \vdash c : C \qquad \Gamma, x : T_2 \vdash c : C}{\Gamma \vdash c[t/x] : C}$$

As ($\exists$ E), the rule ($\sqcup$ E) does not satisfy subject reduction [1, 18].

Problems with Strong Normalization. There are extensions of β-reduction with which the rule ($\sqcup$ E) allows to type non-strongly normalizing terms. We consider *simple rewrite systems* $\mathcal{R}$, that are finite sets of rewrite rules of the form $f(x_1, \dots, x_n) \mapsto_{\mathcal{R}} r$, where $f \in \Sigma_n$, $r \in \Lambda(\Sigma)$ and $x_1, \dots, x_n$ are distinct variables. Algebraic symbols are typed using a rule inspired from [3]:

$$(\textsc{Fun}_{\mathcal{R}})\ \frac{\Gamma \vdash \mathbf{t} : \mathbf{T} \qquad \forall f(\mathbf{x}) \mapsto_{\mathcal{R}} r, \quad \Gamma, \mathbf{x} : \mathbf{T} \vdash r : U}{\Gamma \vdash f(\mathbf{t}) : U}$$

For instance, with the non-deterministic simple rewrite system $x_1 + x_2 \mapsto_+ x_i$ ($i \in \{1, 2\}$), we have $x_1 : T_1, x_2 : T_2 \vdash x_1 + x_2 : T_1 \sqcup T_2$. By combining ($\textsc{Fun}_+$) and ($\sqcup$ E), we can type non-strongly normalizing terms, while the type system with ($\textsc{Fun}_+$) but without ($\sqcup$ E) is strongly normalizing [14].

Example 2.2 ([14]). Let $\delta = \lambda x.xx$, $t_1 = \lambda z.zy\delta$ and $t_2 = \lambda z.\delta$. There are types T_1, T_2, U, V such that $y : V \vdash t_i : T_i$ and $y : V, x : T_i \vdash xx : U$ (note that $t_i t_i \in \mathcal{SN}_{\beta+}$). Using ($\sqcup$ E) and ($\textsc{Fun}_+$) we get $y : V \vdash (t_1 + t_2)(t_1 + t_2) : U$. But $(t_1 + t_2)(t_1 + t_2) \to^*_{\beta+} t_1 t_2 \to^*_{\beta+} \delta\delta \notin \mathcal{SN}_{\beta+}$.

Call-by-Value Eliminations. In this paper, we propose and study a modified version of ($\exists$ E) and ($\sqcup$ E) which uses a call-by-value ($\mathsf{let}\ x = t\ \mathsf{in}\ c$). We consider the extended set of terms

$$\Lambda(\Sigma)_{\mathsf{let}} \quad ::= \quad x \quad | \quad \lambda x.t \quad | \quad t\,u \quad | \quad f(t_1, \dots, t_n) \quad | \quad (\mathsf{let}\ x = t\ \mathsf{in}\ u)\ ,$$

and we replace ($\exists$ E) and ($\sqcup$ E) by the following rules:

$$(\sqcup\,\mathrm{E}_{\mathsf{let}})\ \frac{\Gamma \vdash t : T_1 \sqcup T_2 \qquad \begin{array}{c}\Gamma, x : T_1 \vdash c : C \\ \Gamma, x : T_2 \vdash c : C\end{array}}{\Gamma \vdash (\mathsf{let}\ x = t\ \mathsf{in}\ c) : C} \qquad (\exists\,\mathrm{E}_{\mathsf{let}})\ \frac{\Gamma \vdash t : \exists X.T \quad \Gamma, x : T \vdash c : C}{\Gamma \vdash (\mathsf{let}\ x = t\ \mathsf{in}\ c) : C}\ X \notin \Gamma, C$$

The let is reduced according to a notion of value $\mathcal{V}$ issued from Girard's reducibility candidates (to be defined in Sect. 3). Given a rewrite relation $\to_R$ and a set of values $\mathcal{V}$, we extend $\to_R$ with the smallest rewrite relation $\to_{\mathcal{V}}$ on $\Lambda(\Sigma)_{\mathsf{let}}$ s.t.

$$(\mathsf{let}\ x = v\ \mathsf{in}\ c) \quad \to_{\mathcal{V}} \quad c[v/x] \qquad \text{if} \qquad v \in \mathcal{V}\ .$$

We show that the rules ($\sqcup\,\mathrm{E}_{\mathsf{let}}$) and ($\exists\,\mathrm{E}_{\mathsf{let}}$) lead to strongly normalizing systems regardless of closure under unions. We moreover show that this may allow to recover subject reduction with implicit existential quantification.

Remark 2.3. The subtyping rule $(T_1 \Rightarrow U) \sqcap (T_2 \Rightarrow U) \sqsubseteq (T_1 \sqcup T_2) \Rightarrow U$ causes strong normalization problems similar to those of ($\sqcup$ E) [14]. Our solution does not handle it because it would imply to have a call-by-value arrow, while we want to force call-by-value only locally in ($\sqcup\,\mathrm{E}_{\mathsf{let}}$).

3 Reducibility Candidates

Reducibility candidates, denoted by $\mathcal{CR}_{R\mathcal{E}}$, form a reducibility family depending only on a rewrite relation $\to_R$ and a set of elimination contexts $\mathcal{E}$ [15, 16]. They come with an inherent notion of values $\mathcal{V}_{R\mathcal{E}}$, which are the terms that interact with elimination contexts (see Def. 3.4 below).

An Axiomatization. Our axiomatization of reducibility has been first presented in [15]. We use the version of [16] where details and proofs can be found. Let $[\,] \in \mathcal{X}$ be a distinguished variable and $\to_{\mathsf{R}}$ be a rewrite relation on $\Lambda(\Sigma)$.

Definition 3.1 (Evaluation Contexts). *A set of evaluation contexts for* $\to_{\mathsf{R}}$ *is a set* $\mathcal{E}$ *of terms* $\mathsf{E}[\,]$ *containing exactly one occurrence of* $[\,]$*, and which is closed under reduction: if* $\mathsf{E}[\,] \in \mathcal{E}$ *and* $\mathsf{E}[\,] \to_{\mathsf{R}} \mathsf{t}$ *then* $\mathsf{t} = \mathsf{F}[\,] \in \mathcal{E}$*.*

If $\mathsf{t} \in \Lambda(\Sigma)$ *and* $\mathsf{E}[\,] \in \mathcal{E}$ *then we let* $\mathsf{E}[\mathsf{t}] \ =_{\text{def}}\ (\mathsf{E}[\,])[\mathsf{t}/[\,]]$*.*

Values and neutral terms are defined by an interaction property between terms and evaluation contexts.

Definition 3.2 (Interacton, Values and Neutral Terms). *Let* $\mathcal{E}$ *be a set of evaluation contexts for* $\to_{\mathsf{R}}$*.*

- *A term* t interacts with $\mathsf{E}[\,] \in \mathcal{E}$ for $\to_{\mathsf{R}}$*, notation* $\mathsf{t} \bowtie_{\mathsf{R}} \mathsf{E}[\,]$*, if there is a term* w *such that* $\mathsf{E}[\mathsf{t}] \to_{\mathsf{R}} w$ *and which is not of the form* $\mathsf{E}'[\mathsf{t}']$ *with* $(\mathsf{E}[\,], \mathsf{t}) \to_{\mathsf{R}} (\mathsf{E}'[\,], \mathsf{t}')$*.*
- A value *for* $\to_{\mathsf{R}}$ *in* $\mathcal{E}$ *is a term* v *s.t.* $v \bowtie_{\mathsf{R}} \mathsf{E}[\,]$ *for some* $\mathsf{E}[\,] \in \mathcal{E}$*. We denote by* $\mathcal{V}_{\mathsf{R}\mathcal{E}}$ *the set of values for* $\to_{\mathsf{R}}$ *in* $\mathcal{E}$*. Given* $\mathsf{X} \subseteq \Lambda(\Sigma)$*, the set of* values of X *is* $\mathcal{V}_{\mathsf{R}\mathcal{E}}(\mathsf{X}) =_{\text{def}} \{v \in \mathcal{V}_{\mathsf{R}\mathcal{E}} \mid \mathsf{t} \to^*_{\mathsf{R}} v$ *for some* $\mathsf{t} \in \mathsf{X}\}$*. We write* $\mathcal{V}_{\mathsf{R}\mathcal{E}}(\mathsf{t})$ *for* $\mathcal{V}_{\mathsf{R}\mathcal{E}}(\{\mathsf{t}\})$*.*
- *A term* t *is* neutral for $\to_{\mathsf{R}}$ in $\mathcal{E}$ *if it is not a value. We denote by* $\mathcal{N}_{\mathsf{R}\mathcal{E}}$ *the set of neutral terms for* $\to_{\mathsf{R}}$ *in* $\mathcal{E}$*.*

Values are observable terms, since they interact with evaluation contexts. For instance, the values for β-reduction in evaluation contexts $\mathsf{E}[\,] \in \mathcal{E}_{\Rightarrow} \ ::= \ [\,] \mid \mathsf{E}[\,]\,\mathsf{t}$ are the λ-abstractions $\lambda x.\mathsf{t}$. Note that non-interaction is compositional:

Lemma 3.3. *If* $\mathsf{t} \not\bowtie_{\mathsf{R}} \mathsf{E}[\,]$ *and* $\mathsf{E}[\mathsf{t}] \not\bowtie_{\mathsf{R}} \mathsf{F}[\,]$ *then* $\mathsf{t} \not\bowtie_{\mathsf{R}} \mathsf{F}[\mathsf{E}[\,]]$*.*

Reducibility candidates are defined from a rewrite relation and a set of evaluation contexts satisfying some axioms. These axioms define elimination contexts. The modular extension of this framework with let, presented in Sect. 4, requires to introduces new axioms of closure under substitutions.

Definition 3.4 (Elimination Contexts). *Let* $\mathcal{E}$ *be a set of evaluation contexts for* $\to_{\mathsf{R}}$*. Then* $\mathcal{E}$ *is a set of* elimination contexts *for* $\to_{\mathsf{R}}$ *if*

- $[\,] \in \mathcal{E}$*,*
- ***variables are neutral:*** $\mathcal{X} \subseteq \mathcal{N}_{\mathsf{R}\mathcal{E}}$*,*
- $\mathcal{N}_{\mathsf{R}\mathcal{E}}$ ***is closed under composition with*** $\mathcal{E}$***:*** *if* $\mathsf{t} \in \mathcal{N}_{\mathsf{R}\mathcal{E}}$ *and* $\mathsf{E}[\,] \in \mathcal{E}$ *then* $\mathsf{E}[\mathsf{t}] \in \mathcal{N}_{\mathsf{R}\mathcal{E}}$*.*
- $\mathcal{N}_{\mathsf{R}\mathcal{E}} \setminus \mathcal{X}$ ***is closed under substitution:*** *if* $\mathsf{t} \in \mathcal{N}_{\mathsf{R}\mathcal{E}} \setminus \mathcal{X}$*, then* $\mathsf{t}\sigma \in \mathcal{N}_{\mathsf{R}\mathcal{E}}$*.*
- $\mathcal{V}_{\mathsf{R}\mathcal{E}}$ ***is closed under substitution:*** *if* $\mathsf{t} \in \mathcal{V}_{\mathsf{R}\mathcal{E}}$*, then* $\mathsf{t}\sigma \in \mathcal{V}_{\mathsf{R}\mathcal{E}}$*.*
- $\mathcal{E}$ ***is closed under substitution:*** *if* $\mathsf{E}[\,] \in \mathcal{E}$ *and* $\sigma : (\mathcal{X} \setminus \{[\,]\}) \to \Lambda(\Sigma)$ *then* $\mathsf{E}\sigma[\,] \in \mathcal{E}$*.*

For instance, the contexts $\mathcal{E}_{\Rightarrow}$ are elimination contexts for β-reduction. We assume given a set $\mathcal{E}$ of elimination contexts for $\rightarrow_{R}$.

Definition 3.5 (Reducibility Candidates). *The set of* reducibility candidates *for $\rightarrow_{R}$ in $\mathcal{E}$, written $\mathcal{CR}_{R\mathcal{E}}$, is the set of all $C \subseteq \mathcal{SN}_{R}$ such that*

($\mathcal{CR}0$) *if* $t \in C$ *and* $t \rightarrow_{R} u$ *then* $u \in C$,
($\mathcal{CR}1$) *if* $t \in \mathcal{N}_{R\mathcal{E}}$ *and* $\forall u.\ t \rightarrow_{R} u \implies u \in C$ *then* $t \in C$.

There is a closure operator $\mathcal{CR}_{R\mathcal{E}}(\_) : \mathcal{P}(\mathcal{SN}_{R}) \rightarrow \mathcal{P}(\mathcal{SN}_{R})$ such that $\mathcal{CR}_{R\mathcal{E}}(X)$ is the smallest reducibility candidate containing X if $X \subseteq \mathcal{SN}_{R}$. The closure of the empty set is the set of *hereditary neutral terms*, i.e. the set of strongly normalizing neutral terms which never reduce to a value. Since variables are neutral terms in normal form (recall that $\rightarrow_{R} \subseteq (\Lambda(\Sigma) \setminus \mathcal{X}) \times \Lambda(\Sigma)$), we get $\mathcal{X} \subseteq C$ for all candidate C. It follows that reducibility candidates form a reducibility family. Note that the greatest element of $\mathcal{CR}_{R\mathcal{E}}$ is $\mathcal{SN}_{R}$.

The axioms defining elimination contexts allow to show the following basic property of reducibility candidates. We use it to prove that elimination-based type interpretations (such as the function space $\_ \Rightarrow \_$) preserve $\mathcal{CR}_{R\mathcal{E}}$.

Lemma 3.6 ([16]). *Let* $t \in \mathcal{N}_{R\mathcal{E}}$, $E[\ ] \in \mathcal{E} \cap \mathcal{SN}_{R}$ *and* $C \in \mathcal{CR}_{R\mathcal{E}}$. *If* $E[u] \in C$ *for all* $u \in (t)_{R}$, *then* $E[t] \in C$.

Proof. By induction on $E[\ ] \in \mathcal{SN}_{R}$. Since $E[t]$ is neutral, it is sufficient to show that $(E[t])_{R} \subseteq C$. Let $w \in (E[t])_{R}$. Since t is neutral $w = (E'[t'])$ with $(t, E[\ ]) \rightarrow_{R} (t', E'[\ ])$. If $t \rightarrow_{R} t'$ then $w \in C$ by assumption. Otherwise $E[\ ] \rightarrow_{R} E'[\ ]$, and for all $u \in (t)_{R}$ we have $E[u] \rightarrow_{R} E'[u]$, hence $E'[u] \in C$ by ($\mathcal{CR}0$). Thus $E'[t] \in C$ by induction hypothesis. □

The Values of Reducibility Candidates. An important property of reducibility candidates is that they are uniquely determined by their values:

Lemma 3.7 ([16]). *Given* $X \subseteq \mathcal{SN}_{R}$ *and* $t \in \mathcal{SN}_{R}$, *we have* $t \in \mathcal{CR}_{R\mathcal{E}}(X)$ *if and only if* $\mathcal{V}_{R\mathcal{E}}(t) \subseteq \mathcal{V}_{R\mathcal{E}}(X)$.

This paper builds on the following consequences of Lem. 3.7. Write $\mathcal{A} \subseteq^{*} \mathcal{B}$ when $\mathcal{A}$ is a non-empty subset of $\mathcal{B}$.

Corollary 3.8.

$$\begin{array}{rcll}
\mathcal{CR}_{R\mathcal{E}}(A) \subseteq \mathcal{CR}_{R\mathcal{E}}(B) & \Leftrightarrow & \mathcal{V}_{R\mathcal{E}}(A) \subseteq \mathcal{V}_{R\mathcal{E}}(B) & (A, B \subseteq \mathcal{SN}_{R}) \\
\mathcal{V}_{R\mathcal{E}}(\mathcal{CR}_{R\mathcal{E}}(\bigcup \mathcal{A})) & = & \bigcup \mathcal{V}_{R\mathcal{E}}(\mathcal{A}) & (\mathcal{A} \subseteq^{*} \mathcal{CR}_{R\mathcal{E}}) \\
\mathcal{CR}_{R\mathcal{E}}(B \cup (C_1 \cap C_2)) & = & \mathcal{CR}_{R\mathcal{E}}(B \cup C_1) \cap \mathcal{CR}_{R\mathcal{E}}(B \cup C_2) & (B, C_1, C_2 \in \mathcal{CR}_{R\mathcal{E}}) \\
B \cap \mathcal{CR}_{R\mathcal{E}}(C_1 \cup C_2) & = & \mathcal{CR}_{R\mathcal{E}}((B \cap C_1) \cup (B \cap C_2)) & (B, C_1, C_2 \in \mathcal{CR}_{R\mathcal{E}})
\end{array}$$

The first equality says that $\mathcal{CR}_{R\mathcal{E}}$ is in some sense closed under union w.r.t. values. It will justify the typing rules ($\sqcup\, E_{\mathsf{let}}$) and ($\exists\, E_{\mathsf{let}}$) (Sect. 4.2). The last two ones state the distributivity of the candidate lattice, which is used in Sect. 5.1. All these properties are independent from the closure under unions of $\mathcal{CR}_{R\mathcal{E}}$.

Orthogonality. We now briefly discuss biorthogonality. See [16] for details on biorthogonality in our framework. Let $\rightarrow_{\mathsf{R}}$ be a rewrite relation on $\Lambda(\Sigma)$ and $\mathcal{E}$ be a set of elimination contexts for $\rightarrow_{\mathsf{R}}$. Define $\perp \subseteq \Lambda(\Sigma) \times \mathcal{E}$ as $\mathsf{t} \perp \mathsf{E}[\ ]$ if and only if $\mathsf{E}[\mathsf{t}] \in \mathcal{SN}_{\mathsf{R}}$. Define the *orthogonal* of $\mathcal{A} \subseteq \Lambda(\Sigma)$ (resp. $\mathcal{B} \subseteq \mathcal{E}$) as

$$\begin{array}{rlll} & \mathcal{A}^{\perp} & =_{\mathrm{def}} & \{\mathsf{E}[\ ] \mid \forall \mathsf{t}.\quad \mathsf{t} \in \mathcal{A} \quad \Longrightarrow \quad \mathsf{t} \perp \mathsf{E}[\ ]\} \\ (\text{resp.} & \mathcal{B}^{\perp} & =_{\mathrm{def}} & \{\mathsf{t} \mid \forall \mathsf{E}[\ ].\quad \mathsf{E}[\ ] \in \mathcal{B} \quad \Longrightarrow \quad \mathsf{t} \perp \mathsf{E}[\ ]\}) \end{array}$$

The induced map $(\_)^{\perp\perp}$ on $\mathcal{P}(\mathcal{SN}_{\mathsf{R}})$ is a closure operator [16]. Write $\mathcal{CR}^{\perp\!\!\perp}_{\mathsf{R}\mathcal{E}}$ for $\{\mathcal{A}^{\perp\perp} \mid \mathcal{A} \subseteq^{*} \mathcal{SN}_{\mathsf{R}}\} = \{\mathcal{B}^{\perp} \mid \mathcal{B} \subseteq^{*} \mathcal{E} \cap \mathcal{SN}_{\mathsf{R}}\}$. It is a reducibility family thanks to:

Lemma 3.9 ([16]). $\mathcal{CR}^{\perp\!\!\perp}_{\mathsf{R}\mathcal{E}} \subseteq \mathcal{CR}_{\mathsf{R}\mathcal{E}}$.

4 A Call-by-Value Extension of Reducibility

We now insert the let presented in Sect. 2 into the reducibility candidates issued from a rewrite relation $\rightarrow_{\mathsf{R}}$ on $\Lambda(\Sigma)$ and a set $\mathcal{E}$ of elimination contexts for $\rightarrow_{\mathsf{R}}$.

We first extend the set of terms from $\Lambda(\Sigma)$ to $\Lambda(\Sigma)_{\mathsf{let}}$ (defined in Sect. 2). We then extend $\rightarrow_{\mathsf{R}}$ to the smallest rewrite relation on $\Lambda(\Sigma)_{\mathsf{let}}$ containing the original relation $\rightarrow_{\mathsf{R}}$ (which was defined on $\Lambda(\Sigma)$). Now, let $\rightarrow_{\mathcal{V}}$ be the smallest rewrite relation on $\Lambda(\Sigma)_{\mathsf{let}}$ such that $(\mathsf{let}\ \mathsf{x} = \mathsf{t}\ \mathsf{in}\ \mathsf{u}) \rightarrow_{\mathcal{V}} \mathsf{u}[\mathsf{t}/\mathsf{x}]$ if $\mathsf{t} \in \mathcal{V}_{\mathsf{R}\mathcal{E}}$.

The delicate operation is to extend the set of elimination contexts. We need to extend $\mathcal{E}$ both with contexts of the form $(\mathsf{let}\ \mathsf{x} = \mathsf{E}[\ ]\ \mathsf{in}\ \mathsf{c})$ and with contexts having the same shape as those in $\mathcal{E}$, but built on $\Lambda(\Sigma)_{\mathsf{let}}$ rather than on $\Lambda(\Sigma)$. This second operation is easy to express in the usual cases where $\mathcal{E}$ is defined by a grammar (see Sect. 5.1 and Sect. 5.2). However, performing this operation on an arbitrary set $\mathcal{E}$ while preserving the axioms of Def. 3.4 leads us to some technicalities. We chose to close $\mathcal{E}$ by $\Lambda(\Sigma)_{\mathsf{let}}$-substitution, but this causes difficulties w.r.t. stability by reduction. A solution is to close by substitution only the contexts which are *linear* in $\mathcal{X}$, but they need not to be stable by reduction. We therefore work with the set $\mathcal{E}^{\mathrm{Rlin}}$, defined as the set of R-normal linear $\mathsf{E}[\ ] \in \mathcal{E}$.

Definition 4.1. *Let $\mathcal{E}_{\mathsf{let}}$ be the smallest set such that*

$$\begin{array}{rcl} \big(\mathsf{E}[\ ] \in \mathcal{E}^{\mathrm{Rlin}} \quad \wedge \quad \sigma : (\mathcal{X} \setminus \{[\ ]\}) \rightarrow \Lambda(\Sigma)_{\mathsf{let}}\big) & \Longrightarrow & \mathsf{E}[\ ]\sigma \in \mathcal{E}_{\mathsf{let}} \\ (\mathsf{E}[\ ], \mathsf{F}[\ ] \in \mathcal{E}_{\mathsf{let}} \quad \wedge \quad \mathsf{t} \in \Lambda(\Sigma)_{\mathsf{let}}) & \Longrightarrow & \mathsf{E}[(\mathsf{let}\ \mathsf{x} = \mathsf{F}[\ ]\ \mathsf{in}\ \mathsf{t})] \in \mathcal{E}_{\mathsf{let}} \end{array}$$

We now show that $\mathcal{E}_{\mathsf{let}}$ is a set of elimination contexts for $\rightarrow_{\mathsf{R}\mathcal{V}}$. This allows to define reducibility candidates and biorthogonals. We then show that this turns the closure of an union into an elimination-based interpretation.

4.1 Values, Neutral Terms and Elimination Contexts

In this section, we give the main steps of the proof that $\mathcal{E}_{\mathsf{let}}$ is a set of elimination contexts for $\rightarrow_{\mathsf{R}\mathcal{V}}$. We first give a characterization of the values and neutral terms for $\rightarrow_{\mathsf{R}\mathcal{V}}$ in $\mathcal{E}_{\mathsf{let}}$ in terms of those for $\rightarrow_{\mathsf{R}}$ in $\mathcal{E}$. Note that the let ensures that values for $\rightarrow_{\mathsf{R}}$ in $\mathcal{E}$ are values for $\rightarrow_{\mathsf{R}\mathcal{V}}$ in $\mathcal{E}_{\mathsf{let}}$. The characterization of neutral terms relies on the fact that values and non-variable neutral terms are not unifiable in $\Lambda(\Sigma)_{\mathsf{let}}$, which follows from the substitution axioms of Def. 3.4.

Proposition 4.2 (Values and Neutral Terms).

$$\begin{aligned}
\mathcal{V}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}} &= \{v\sigma \mid v \in \Lambda(\Sigma) \cap \mathcal{V}_{\mathsf{R}\mathcal{E}} \quad \land \quad \sigma : \mathcal{X} \to \Lambda(\Sigma)_{\mathsf{let}}\}\ ,\\
\mathcal{N}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}} &= \mathcal{X} \quad \cup \quad \{(\mathsf{let}\ x = t\ \mathsf{in}\ c) \mid t, c \in \Lambda(\Sigma)_{\mathsf{let}}\}\\
&\cup \quad \{n\sigma \mid n \in (\Lambda(\Sigma) \cap \mathcal{N}_{\mathsf{R}\mathcal{E}}) \setminus \mathcal{X} \quad \land \quad \sigma : \mathcal{X} \to \Lambda(\Sigma)_{\mathsf{let}}\}\ .
\end{aligned}$$

In order to show that $\mathcal{E}_{\mathsf{let}}$ is a set of elimination contexts for $\to_{\mathsf{R}\mathcal{V}}$, we have to check the axioms of Defs. 3.1 and 3.4. Concerning Def. 3.1, linearity in $[\ ]$ follows from an easy induction on $\mathcal{E}_{\mathsf{let}}$, using linearity in $[\ ]$ of $\mathcal{E}$ in the base case. Stability by reduction is a consequence of the condition $\mathsf{E}[\ ] \in \mathcal{E}^{\mathsf{Rlin}}$ in the first clause of the definition of $\mathcal{E}_{\mathsf{let}}$.

Concerning Def. 3.4, stability by substitution for values and non-variable neutral terms follows directly from Prop. 4.2. Stability by substitution for contexts follows from an easy induction on $\mathcal{E}_{\mathsf{let}}$. Moreover, variables are neutral by Prop. 4.2, and the preservation of neutral terms by composition with elimination contexts is a consequence of Prop. 4.2 shown by induction on $\mathcal{E}_{\mathsf{let}}$. Hence we have

Theorem 4.3. *$\mathcal{E}_{\mathsf{let}}$ is a set of elimination contexts for $\to_{\mathsf{R}\mathcal{V}}$.*

We thus obtain reducibility candidates $\mathfrak{CR}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}}$ and biorthogonals $\mathfrak{CR}^{\perp\!\!\perp}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}}$ directly from a rewrite relation $\to_{\mathsf{R}}$ on $\Lambda(\Sigma)$ and a set of elimination contexts $\mathcal{E}$.

4.2 An Elimination-Based Interpretation of Unions

In this section, we show that for reducibility candidates and biorthogonals, the let turns $\mathcal{R}ed(\bigcup\mathcal{A})$ into the elimination-based interpretation $\bigvee_{\mathcal{R}ed}\mathcal{A}$ defined as

$$\Big\{\mathsf{t} \mid \forall \mathsf{C} \in \mathcal{R}ed,\ \forall \mathsf{c} \text{ s.t. } u \in \bigcup\mathcal{A} \Rightarrow \mathsf{c}[u/x] \in \mathsf{C},\ (\mathsf{let}\ x = \mathsf{t}\ \mathsf{in}\ \mathsf{c}) \in \mathsf{C}\Big\}\ .$$

Hence, we get elimination-based interpretations of union and implicit existential types. This ensures the adequacy of reducibility candidates and biorthogonals w.r.t. $(\exists\,\mathrm{E}_{\mathsf{let}})$ and $(\sqcup\,\mathrm{E}_{\mathsf{let}})$, but does not change the definition of the type interpretation using the closure of unions (see Sect. 5).

Write $\mathfrak{CR}$ for $\mathfrak{CR}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}}$, $\mathfrak{CR}^{\perp\!\!\perp}$ for $\mathfrak{CR}^{\perp\!\!\perp}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}}$, $\mathcal{N}$ for $\mathcal{N}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}}$, $\mathcal{V}$ for $\mathcal{V}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}}$ and $\mathcal{SN}$ for $\mathcal{SN}_{\mathsf{R}\mathcal{V}}$. We begin by showing that $\bigvee_{\mathfrak{CR}}\mathcal{A}$ (resp. $\bigvee_{\mathfrak{CR}^{\perp\!\!\perp}}\mathcal{A}$) is a reducibility candidate (resp. a biorthogonal).

Lemma 4.4. *If $\mathcal{A} \subseteq^* \mathfrak{CR}$ then $\bigvee_{\mathfrak{CR}}\mathcal{A} \in \mathfrak{CR}$.*

Proof. Write $\bigvee\mathcal{A}$ for $\bigvee_{\mathfrak{CR}}\mathcal{A}$. We first show that $\bigvee\mathcal{A} \subseteq \mathcal{SN}$. Let $\mathsf{t} \in \bigvee\mathcal{A}$ and take $\mathsf{C} =_{\mathrm{def}} \mathfrak{CR}(\bigcup\mathcal{A})$. Since $\bigcup\mathcal{A} \subseteq \mathfrak{CR}(\bigcup\mathcal{A})$, we get $(\mathsf{let}\ x = \mathsf{t}\ \mathsf{in}\ x) \in \mathsf{C} \subseteq \mathcal{SN}$, hence $\mathsf{t} \in \mathcal{SN}$. Stability by reduction $(\mathfrak{CR}0)$ is immediate. For the clause $(\mathfrak{CR}1)$, take $\mathsf{t} \in \mathcal{N}$ such that $(\mathsf{t})_{\mathsf{R}\mathcal{V}} \subseteq \bigvee\mathcal{A}$. Let $\mathsf{C} \in \mathfrak{CR}$ and c such that $u \in \bigcup\mathcal{A}$ implies $\mathsf{c}[u/x] \in \mathsf{C}$. Since $(\mathsf{let}\ x = [\ ]\ \mathsf{in}\ \mathsf{c}) \in \mathcal{E}_{\mathsf{let}} \cap \mathcal{SN}$ and $(\mathsf{let}\ x = \mathsf{t}'\ \mathsf{in}\ \mathsf{c}) \in \mathsf{C}$ for all $\mathsf{t}' \in (\mathsf{t})_{\mathcal{R}\mathcal{V}}$, we have $(\mathsf{let}\ x = \mathsf{t}\ \mathsf{in}\ \mathsf{c}) \in \mathsf{C}$ by Lem. 3.3. □

Lemma 4.5. *If $\mathcal{A} \subseteq^* \mathfrak{CR}^{\perp\!\!\perp}$ then $\bigvee_{\mathfrak{CR}^{\perp\!\!\perp}}\mathcal{A} \in \mathfrak{CR}^{\perp\!\!\perp}$.*

Proof. Since $\bigvee_{\mathcal{CR}^{\perp\!\!\perp}} \mathcal{A}$ is the orthogonal of the non-empty subset of $\mathcal{SN}$

$$\left\{ \mathsf{E}[(\mathsf{let}\ \mathsf{x} = [\,]\ \mathsf{in}\ \mathsf{c})] \mid \mathsf{C} \in \mathcal{CR}^{\perp\!\!\perp} \ \wedge\ \mathsf{E}[\,] \in \mathsf{C}^{\perp} \ \wedge\ \left(\mathsf{u} \in \bigcup \mathcal{A} \Rightarrow \mathsf{c}[\mathsf{u}/\mathsf{x}] \in \mathsf{C}\right) \right\} .$$

□

We now show that $\bigvee_{\mathcal{R}ed}$ is the closure of the union when $\mathcal{R}ed \in \{\mathcal{CR}, \mathcal{CR}^{\perp\!\!\perp}\}$.

Theorem 4.6. *If $\mathcal{R}ed \in \{\mathcal{CR}, \mathcal{CR}^{\perp\!\!\perp}\}$ and $\mathcal{A} \subseteq^{*} \mathcal{R}ed$ then $\bigvee_{\mathcal{R}ed} \mathcal{A} = \mathcal{R}ed(\bigcup \mathcal{A})$.*

Proof. $(\supseteq)$**.** Since $\mathcal{R}ed$ is defined by a closure operator, by Lem. 4.4 and 4.5 it is sufficient to show that $\bigcup \mathcal{A} \subseteq \bigvee_{\mathcal{R}ed} \mathcal{A}$. Let $\mathsf{t} \in \bigcup \mathcal{A}$, $\mathsf{C} \in \mathcal{R}ed$ and c s.t. $\mathsf{u} \in \bigcup \mathcal{A}$ implies $\mathsf{c}[\mathsf{u}/\mathsf{x}] \in \mathsf{C}$. Note that $\mathsf{t}, \mathsf{c} \in \mathcal{SN}$ since $\bigcup \mathcal{A}$ is not empty. By Lem. 3.9, we have $\mathsf{C} \in \mathcal{CR}$. Since $(\mathsf{let}\ \mathsf{x} = \mathsf{t}\ \mathsf{in}\ \mathsf{c})$ is neutral, it is sufficient to show that $((\mathsf{let}\ \mathsf{x} = \mathsf{t}\ \mathsf{in}\ \mathsf{c}))_{\mathsf{R}\mathcal{V}} \subseteq \mathsf{C}$. We reason by induction on pairs (t, c) ordered by the product extension of $\rightarrow_{\mathsf{R}\mathcal{V}}$. Let w such that $(\mathsf{let}\ \mathsf{x} = \mathsf{t}\ \mathsf{in}\ \mathsf{c}) \rightarrow_{\mathsf{R}\mathcal{V}} w$. There are two cases.

$w = \mathsf{c}[\mathsf{t}/\mathsf{x}]$ **&** $\mathsf{t} \in \mathcal{V}$**.** Since $\mathsf{t} \in \bigcup \mathcal{A}$, we get $\mathsf{c}[\mathsf{t}/\mathsf{x}] \in \mathsf{C}$ by assumption on C.

$w = (\mathsf{let}\ \mathsf{x} = \mathsf{t}'\ \mathsf{in}\ \mathsf{c}')$ **&** $(\mathsf{t}, \mathsf{c}) \rightarrow_{\mathsf{R}\mathcal{V}} (\mathsf{t}', \mathsf{c}')$**.** By $(\mathcal{CR}0)$, we have $\mathsf{t}' \in \bigcup \mathcal{A}$ and that $\mathsf{u} \in \bigcup \mathcal{A} \Rightarrow \mathsf{c}'[\mathsf{u}/\mathsf{x}] \in \mathsf{C}$. We get $w \in \mathsf{C}$ by induction hypothesis.

$(\subseteq)$**.** We have $\bigvee_{\mathcal{R}ed} \mathcal{A} \in \mathcal{R}ed$ by Lem. 4.4 and Lem. 4.5. Since $\mathcal{CR}^{\perp\!\!\perp} \subseteq \mathcal{CR}$ we have $\bigvee_{\mathcal{R}ed} \mathcal{A}, \mathcal{R}ed(\bigcup \mathcal{A}) \in \mathcal{CR}$. Therefore, by Cor. 3.8 it is sufficient to show that $\mathcal{V}(\bigvee_{\mathcal{R}ed} \mathcal{A}) \subseteq \mathcal{R}ed(\bigcup \mathcal{A})$. Let $\mathsf{v} \in \mathcal{V}(\bigvee_{\mathcal{R}ed} \mathcal{A})$ and take $\mathsf{C} =_{\mathrm{def}} \mathcal{R}ed(\bigcup \mathcal{A})$. Since $\bigcup \mathcal{A} \subseteq \mathsf{C}$, we obtain that $(\mathsf{let}\ \mathsf{x} = \mathsf{v}\ \mathsf{in}\ \mathsf{x}) \in \mathsf{C}$. But $(\mathsf{let}\ \mathsf{x} = \mathsf{v}\ \mathsf{in}\ \mathsf{x}) \rightarrow_{\mathcal{V}} \mathsf{v}$ since $\mathsf{v} \in \mathcal{V}$. Hence $\mathsf{v} \in \mathsf{C} = \mathcal{R}ed(\bigcup \mathcal{A})$ by $(\mathcal{CR}0)$. □

5 Applications

We now discuss three applications. Our approach is to start from a given calculus made of a type system and a rewrite relation $\rightarrow_{\mathsf{R}}$. We provide a set of elimination contexts $\mathcal{E}$. This gives reducibility candidates (and biorthogonals) as in Sect. 3. We then apply the method of Sect. 4. Terms are extended with let, the rewrite relation $\rightarrow_{\mathsf{R}}$ is extended with $\rightarrow_{\mathcal{V}}$, and we obtain elimination contexts $\mathcal{E}_{\mathsf{let}}$ by Def. 4.1. Using Thm. 4.3, we obtain reducibility candidates $\mathcal{CR}$ (and biorthogonals $\mathcal{CR}^{\perp\!\!\perp}$) for $\rightarrow_{\mathsf{R}\mathcal{V}}$ in $\mathcal{E}_{\mathsf{let}}$. We then extend the type system with $(\sqcup\, \mathrm{E}_{\mathsf{let}})$ or $(\exists\, \mathrm{E}_{\mathsf{let}})$. Adequacy w.r.t. these rules is ensured by Thm. 4.6.

The first application concerns union types, the second one existential types and the third one deals with existential quantification in realizability models based on reducibility candidates and biorthogonals.

5.1 Union Types

We apply the framework presented in Sect. 4 to a calculus with union types, intersection types and (possibly non-deterministic) simple rewrite rules. We focus on reducibility candidates and do not consider biorthogonals since their lattice is not distributive. Some proofs are postponed until Sect. 5.2.

We consider simple types with unions and intersections $\mathcal{T}_{\sqcap\sqcup}$. As in [5], they are equipped with a preorder $\sqsubseteq$ so that $(\mathcal{T}, \sqsubseteq)$ is a distributive lattice satisfying the additional axioms

$$(T \Rightarrow U_1) \sqcap (T \Rightarrow U_2) \sqsubseteq T \Rightarrow (U_1 \sqcap U_2) \qquad \text{and} \qquad \frac{U_1 \sqsubseteq U_2 \qquad T_2 \sqsubseteq T_1}{U_2 \Rightarrow T_2 \sqsubseteq U_1 \Rightarrow T_1}$$

Given a simple rewrite system $\mathcal{R}$, we add to $\boldsymbol{\lambda}$ the rule $(\textsc{Fun}_{\mathcal{R}})$ and

$$(\sqcap\mathrm{I})\ \frac{\Gamma \vdash t : T \qquad \Gamma \vdash t : U}{\Gamma \vdash t : T \sqcap U} \qquad\qquad (\textsc{Sub})\ \frac{\Gamma \vdash t : T \qquad T \sqsubseteq U}{\Gamma \vdash t : U}$$

We consider reducibility candidates for $\rightarrow_{\beta\mathcal{R}}$ in elimination contexts $\mathcal{E}_{\Rightarrow}$. By Thm. 4.3, we thus obtain reducibility candidates $\mathfrak{CR}$ for $\rightarrow_{\beta\mathcal{R}\mathcal{V}}$ in the elimination contexts $\mathcal{E}_{\Rightarrow\mathsf{let}}$. Note that $(\mathsf{let}\ x = t\ \mathsf{in}\ c) \rightarrow_{\mathcal{V}} c[t/x]$ if and only if $t = \lambda y.u$. Therefore, continuing Ex. 2.2, we have $y : V \vdash (\mathsf{let}\ x = t_1 + t_2\ \mathsf{in}\ xx) : U$, but $(\mathsf{let}\ x = t_1 + t_2\ \mathsf{in}\ xx)$ does not reduce to $t_1 t_2$: since $t_1 + t_2$ is not a value, we must choose between t_1 and t_2 before performing the substitution.

Note that $\mathcal{E}_{\Rightarrow\mathsf{let}}$ is the following set of contexts (see Lem. 5.2):

$$E[\,] \quad ::= \quad [\,] \quad | \quad E[\,]\,t \quad | \quad (\mathsf{let}\ x = E[\,]\ \mathsf{in}\ t)\ .$$

We now add the rule $(\sqcup\mathrm{E}_{\mathsf{let}})$ to the system.

Types are interpreted as in Sect. 2. Write $[\![\_]\!]_\rho$ for $[\![\_]\!]^{\mathfrak{CR}}_\rho$. The correctness of the subtyping relation is standard [14], excepted that distributivity is ensured by Cor. 3.8. We get $[\![\_]\!]_\rho : \mathcal{T}_{\sqcap\sqcup} \to \mathfrak{CR}$ since $\_ \Rightarrow \_ : \mathfrak{CR}^2 \to \mathfrak{CR}$ (Lem. 5.3).

Theorem 5.1. *If* $\Gamma \vdash t : T$ *and* $(\rho, \sigma) \models \Gamma$ *then* $t\sigma \in [\![T]\!]_\rho$.

Proof. By induction on $\Gamma \vdash t : T$. The case of $(\sqcup\mathrm{E}_{\mathsf{let}})$ follows from Thm. 4.6, which entails that $[\![T \sqcup U]\!]_\rho = [\![T]\!]_\rho \vee_{\mathfrak{CR}} [\![U]\!]_\rho$ (see the proof of Thm. 5.5). Adequacy w.r.t. the other rules is standard [14]. □

5.2 Implicit Existential Types

We apply the framework of Sect. 4 to a calculus with existential, product and sum types. We show that it enjoys strong normalization and subject reduction.

The System. We let $\mathcal{T}_{\forall\exists\times+}$ be the extension of $\mathcal{T}_{\forall\exists}$ with the binary product $T \times U$ and the binary sum $T + U$. Terms are built on the signature

$$\Sigma \quad =_{\mathrm{def}} \quad \{\langle \_, \_ \rangle,\ \pi_1\,\_,\ \pi_2\,\_,\ \mathsf{inj}_1\,\_,\ \mathsf{inj}_2\,\_,\ \mathsf{case}(\_,\_,\_)\}\ .$$

We extend β-reduction with $\pi_i\langle t_1, t_2\rangle \rightarrow_\beta t_i$ and $\mathsf{case}(\mathsf{inj}_i t, u_1, u_2) \rightarrow_\beta u_i t$. The type system is that of $\boldsymbol{\lambda}_\forall$ enriched with $(\exists\mathrm{I})$ and the following rules:

$$(\times\mathrm{I})\ \frac{\Gamma \vdash t_1 : T_1 \qquad \Gamma \vdash t_2 : T_2}{\Gamma \vdash \langle t_1, t_2\rangle : T_1 \times T_2} \qquad\qquad (\times\mathrm{E})\ \frac{\Gamma \vdash t : T_1 \times T_2}{\Gamma \vdash \pi_i t : T_i}\ (i \in \{1,2\})$$

$$(+\mathrm{I})\ \frac{\Gamma \vdash t : T_i}{\Gamma \vdash \mathsf{inj}_i t : T_1 + T_2}\ (i \in \{1,2\}) \qquad\qquad (+\mathrm{E})\ \frac{\Gamma \vdash t : T_1 + T_2 \qquad \begin{array}{c}\Gamma \vdash u_1 : T_1 \Rightarrow U \\ \Gamma \vdash u_2 : T_2 \Rightarrow U\end{array}}{\Gamma \vdash \mathsf{case}(t, u_1, u_2) : U}$$

Reducibility. We consider the elimination contexts

$$\mathsf{E}[\ ] \in \mathcal{E} \quad ::= \quad [\ ] \quad | \quad \mathsf{E}[\ ]\,\mathsf{t} \quad | \quad \pi_i\,\mathsf{E}[\ ] \quad | \quad \mathsf{case}(\mathsf{E}[\ ], \mathsf{t}, \mathsf{u}) \ .$$

The values are the terms of the form $\lambda x.\mathsf{t}$, $\langle \mathsf{t}, \mathsf{u}\rangle$ or $\mathsf{inj}_i\ \mathsf{t}$. We consider reducibility candidates for $\rightarrow_\beta$ in $\mathcal{E}$. By Thm. 4.3, we thus obtain reducibility candidates $\mathfrak{CR}$ and biorthogonals $\mathfrak{CR}^{\perp\perp}$ for $\rightarrow_{\beta\mathcal{V}}$ in the elimination contexts $\mathcal{E}_{\mathsf{let}}$.

Lemma 5.2. $\mathcal{E}_{\mathsf{let}}$ *is the set of contexts*

$$\mathsf{E}[\ ] \ ::= \ [\ ] \ | \ \mathsf{E}[\ ]\,\mathsf{t} \ | \ \pi_i\,\mathsf{E}[\ ] \ | \ \mathsf{case}(\mathsf{E}[\ ], \mathsf{t}, \mathsf{u}) \ | \ (\mathsf{let}\ x = \mathsf{E}[\ ]\ \mathsf{in}\ \mathsf{t}) \ .$$

We add the rule $(\exists\,\mathrm{E}_{\mathsf{let}})$ to the system. In the remaining of this section, we assume that $\mathcal{R}ed \in \{\mathfrak{CR}, \mathfrak{CR}^{\perp\perp}\}$. We extend the interpretation of Sect. 2 as follows:

$$\begin{array}{lll}
[\![\mathsf{T} \times \mathsf{U}]\!]^{\mathcal{R}ed}_\rho & =_{\mathrm{def}} & \{\mathsf{t} \mid \pi_1 \mathsf{t} \in [\![\mathsf{T}]\!]^{\mathcal{R}ed}_\rho \quad \wedge \quad \pi_2 \mathsf{t} \in [\![\mathsf{U}]\!]^{\mathcal{R}ed}_\rho\} \\
[\![\mathsf{T} + \mathsf{U}]\!]^{\mathcal{R}ed}_\rho & =_{\mathrm{def}} & \{\mathsf{t} \mid \forall \mathsf{C} \in \mathcal{R}ed, \quad \forall \mathsf{s} \in [\![\mathsf{T}]\!]^{\mathcal{R}ed}_\rho \Rightarrow \mathsf{C}, \quad \forall \mathsf{u} \in [\![\mathsf{U}]\!]^{\mathcal{R}ed}_\rho \Rightarrow \mathsf{C}, \\
 & & \hfill \mathsf{case}(\mathsf{t}, \mathsf{s}, \mathsf{u}) \in \mathsf{C}\ \}
\end{array}$$

The correctness of the definition relies on the fact that the shape of elimination contexts has not been destroyed by their extension with let (Lem. 5.2).

Lemma 5.3. *If* $\rho : \mathcal{X}_{\mathcal{T}} \rightarrow \mathcal{T}_{\forall\exists\times+}$ *and* $\mathsf{T} \in \mathcal{T}_{\forall\exists\times+}$ *then* $[\![\mathsf{T}]\!]^{\mathcal{R}ed}_\rho \in \mathcal{R}ed$.

Proof. In both cases for $\mathcal{R}ed$, we reason by induction on T. The result is trivial if T is a variable. The cases of $\forall \mathsf{X}.\mathsf{T}$ and $\exists \mathsf{X}.\mathsf{T}$ follow from the fact that $\mathcal{R}ed$ is defined by a closure operator. We detail the case of $\mathsf{A} \Rightarrow \mathsf{B}$. Write $[\![\mathsf{T}]\!]$ for $[\![\mathsf{T}]\!]^{\mathcal{R}ed}_\rho$.

$\mathcal{R}ed = \mathfrak{CR}$**.** We have $[\![\mathsf{A}]\!] \Rightarrow [\![\mathsf{B}]\!] \subseteq \mathcal{SN}_{\beta\mathcal{V}}$ since $[\![\mathsf{A}]\!]$ is not empty and $[\![\mathsf{B}]\!] \subseteq \mathcal{SN}_{\beta\mathcal{V}}$. Stability by reduction follows from $(\mathfrak{CR}0)$ on $[\![\mathsf{B}]\!]$. We get $(\mathfrak{CR}1)$ by Lem. 3.6, using that $[\ ]\mathsf{t} \in \mathcal{E}_{\mathsf{let}}$ for all $\mathsf{t} \in \Lambda(\Sigma)_{\mathsf{let}}$ thanks to Lem. 5.2.

$\mathcal{R}ed = \mathfrak{CR}^{\perp\perp}$**.** Since $[\![\mathsf{A}]\!] \Rightarrow [\![\mathsf{B}]\!]$ is orthogonal to $\{\mathsf{E}[[\ ]\mathsf{t}] \mid \mathsf{t} \in [\![\mathsf{A}]\!] \wedge \mathsf{E}[\ ] \in [\![\mathsf{B}]\!]^\perp\}$ which is a non-empty subset of $\mathcal{SN}_{\beta\mathcal{V}}$. □

Since type constructors are interpreted by eliminations, Lem. 5.3 implies adequacy w.r.t. the elimination rules $(\Rightarrow \mathrm{E})$, $(\times \mathrm{E})$ and $(+\mathrm{E})$. Adequacy w.r.t. the rules $(\Rightarrow \mathrm{I})$, $(\times \mathrm{I})$ and $(+\mathrm{I})$ follows from the following saturation lemma (recall that $\mathcal{SN}_{\beta\mathcal{V}} \in \mathfrak{CR}$ and $[\ ] \in \mathcal{E}_{\mathsf{let}}$).

Lemma 5.4. *For all* $\mathsf{u}, \mathsf{s} \in \mathcal{SN}_{\beta\mathcal{V}}$*, all* $\mathsf{E}[\ ] \in \mathcal{E}_{\mathsf{let}}$ *and all* $\mathsf{C} \in \mathfrak{CR}$*,*

$$\mathsf{E}[(\lambda x.\mathsf{t})\mathsf{u}] \in \mathsf{C} \quad \textit{if} \quad \mathsf{E}[\mathsf{t}[\mathsf{u}/x]] \in \mathsf{C}$$

$$\begin{array}{lll@{\qquad}lll}
\mathsf{E}[\pi_1 \langle \mathsf{t}, \mathsf{u}\rangle] \in \mathsf{C} & \textit{if} & \mathsf{E}[\mathsf{t}] \in \mathsf{C} & \mathsf{E}[\mathsf{case}(\mathsf{inj}_1 \mathsf{t}, \mathsf{u}, \mathsf{s})] \in \mathsf{C} & \textit{if} & \mathsf{E}[\mathsf{u}\mathsf{t}] \in \mathsf{C} \\
\mathsf{E}[\pi_2 \langle \mathsf{u}, \mathsf{t}\rangle] \in \mathsf{C} & \textit{if} & \mathsf{E}[\mathsf{t}] \in \mathsf{C} & \mathsf{E}[\mathsf{case}(\mathsf{inj}_2 \mathsf{t}, \mathsf{s}, \mathsf{u})] \in \mathsf{C} & \textit{if} & \mathsf{E}[\mathsf{u}\mathsf{t}] \in \mathsf{C}
\end{array}$$

Proof. Note that $\mathsf{t}, \mathsf{E}[\ ] \in \mathcal{SN}_{\beta\mathcal{V}}$ since $\mathsf{C} \subseteq \mathcal{SN}_{\beta\mathcal{V}}$. We detail the case of $\mathsf{E}[(\lambda x.\mathsf{t})\mathsf{u}]$. Since $(\lambda x.\mathsf{t})\mathsf{u}$ is neutral, by Lem. 3.6 it is sufficient to show that $\mathsf{E}[w] \in \mathsf{C}$ if $(\lambda x.\mathsf{t})\mathsf{u} \rightarrow_\beta w$. We reason by induction on (t, u) ordered by the product extension of $\rightarrow_\beta$. Let w such that $(\lambda x.\mathsf{t})\mathsf{u} \rightarrow_\beta w$. If $w = \mathsf{t}[\mathsf{u}/x]$ then we are done by assumption. Otherwise, $w = (\lambda x.\mathsf{t}')\mathsf{u}'$ with $(\mathsf{t}, \mathsf{u}) \rightarrow_\beta (\mathsf{t}', \mathsf{u}')$ and we conclude by induction hypothesis, using that C is closed under reduction. □

Using Thm. 4.6 for the existential quantification, adequacy follows as usual.

Theorem 5.5. *If* $\Gamma \vdash \mathsf{t} : \mathsf{T}$ *and* $(\rho, \sigma) \models_{\mathcal{Red}} \Gamma$ *then* $\mathsf{t}\sigma \in [\![\mathsf{T}]\!]^{\mathcal{Red}}_{\rho}$.

Proof. By induction on $\Gamma \vdash \mathsf{t} : \mathsf{T}$. We detail the case of $(\exists\,\mathrm{E}_{\mathsf{let}})$. Let $(\rho, \sigma) \models \Gamma$. By induction hypothesis we have $\mathsf{t}\sigma \in [\![\exists \mathsf{X}.\mathsf{T}]\!]^{\mathcal{Red}}_{\rho}$. Since $\mathsf{X} \notin \Gamma, \mathsf{U}$, if $\mathsf{C} \in \mathcal{Red}$ and $\mathsf{u} \in [\![\mathsf{T}]\!]^{\mathcal{Red}}_{\rho[\mathsf{C}/\mathsf{X}]}$ then $(\rho[\mathsf{C}/\mathsf{X}], \sigma[\mathsf{u}/\mathsf{x}]) \models_{\mathcal{Red}} \Gamma, \mathsf{x} : \mathsf{T}$. By induction hypothesis again, we deduce that $\mathsf{c}\sigma[\mathsf{u}/\mathsf{x}] \in [\![\mathsf{U}]\!]^{\mathcal{Red}}_{\rho}$ for all $\mathsf{u} \in \bigcup_{\mathsf{C} \in \mathcal{Red}} [\![\mathsf{T}]\!]^{\mathcal{Red}}_{\rho[\mathsf{C}/\mathsf{X}]}$. We obtain $(\mathsf{let}\ \mathsf{x} = \mathsf{t}\sigma\ \mathsf{in}\ \mathsf{c}\sigma) \in [\![\mathsf{C}]\!]^{\mathcal{Red}}_{\rho}$ since $[\![\exists \mathsf{X}.\mathsf{T}]\!]^{\mathcal{Red}}_{\rho} = \bigvee_{\mathsf{C} \in \mathcal{Red}} [\![\mathsf{T}]\!]^{\mathcal{Red}}_{\rho[\mathsf{C}/\mathsf{X}]}$ by Thm. 4.6. □

Subject Reduction. We sketch the proof that let allows to recover subject reduction for implicit existential types. The key-point is that the reduction of a $(\mathsf{let}\ \mathsf{x} = \mathsf{t}\ \mathsf{in}\ \mathsf{c})$ obtained from $(\exists\,\mathrm{E}_{\mathsf{let}})$ is only made when t is a value, in which case we know that its type has been obtained by a $(\exists\,\mathrm{I})$ rule (Prop. 5.6). This prevents us from the counter example of [18] (Ex. 2.1): we can type $(\mathsf{let}\ w = \mathrm{I}xy\ \mathsf{in}\ zww)$ but this term does not reduce to u_1 since $\mathrm{I}xy$ is not a value.

Proposition 5.6 (Determinacy of Typing). *Let* $\mathsf{v} \in \mathcal{V}_{\beta\mathcal{V}\mathcal{E}_{\mathsf{let}}}$. *If* $\Gamma \vdash \mathsf{v} : \exists \mathsf{X}.\mathsf{T}$ *then there is* U *such that* $\Gamma \vdash \mathsf{v} : \mathsf{T}[\mathsf{U}/\mathsf{X}]$.

The remaining of the proof directly follows that of [9] for $\boldsymbol{\lambda}_{\forall}$. It relies on usual substitution and inversion properties.

Substitution. (i) If $\Gamma, \mathsf{x} : \mathsf{U} \vdash \mathsf{t} : \mathsf{T}$ and $\Gamma \vdash \mathsf{u} : \mathsf{U}$ then $\Gamma \vdash \mathsf{t}[\mathsf{u}/\mathsf{x}] : \mathsf{T}$.
(ii) If $\Gamma \vdash \mathsf{t} : \mathsf{T}$ then $\Gamma[\mathsf{U}/\mathsf{X}] \vdash \mathsf{t} : \mathsf{T}[\mathsf{U}/\mathsf{X}]$.
Inversion. (i) If $\Gamma \vdash \lambda \mathsf{x}.\mathsf{t} : \mathsf{U} \Rightarrow \mathsf{T}$ then $\Gamma, \mathsf{x} : \mathsf{U} \vdash \mathsf{t} : \mathsf{T}$.
(ii) If $\Gamma \vdash \langle \mathsf{t}_1, \mathsf{t}_2 \rangle : \mathsf{A}_1 \times \mathsf{A}_2$ then $\Gamma \vdash \mathsf{t}_i : \mathsf{A}_i$ for all $i \in \{1, 2\}$.
(iii) If $\Gamma \vdash \mathsf{inj}_i \mathsf{t} : \mathsf{A}_1 + \mathsf{A}_2$ then $\Gamma \vdash \mathsf{t} : \mathsf{A}_i$.

Theorem 5.7 (Subject Reduction). *If* $\Gamma \vdash \mathsf{t} : \mathsf{T}$ *and* $\mathsf{t} \to_{\beta\mathcal{V}} \mathsf{u}$ *then* $\Gamma \vdash \mathsf{u} : \mathsf{T}$.

5.3 Realizability Semantics of Existential Quantification

We now discuss the realizability interpretation of existential quantification in presence of the let. Realizability with implicit existential types is used e.g. in [12].

Given a reducibility family $\mathcal{Red}$, write $\mathsf{t} \Vdash_{\mathcal{Red}} \mathsf{A}$ (read t *realizes* A) if t is a closed term such that $\mathsf{t} \in [\![\mathsf{A}]\!]^{\mathcal{Red}}_{\rho}$ for all $\rho : \mathcal{X}_{\mathcal{T}} \to \mathcal{Red}$. In realizability models based on $\boldsymbol{\lambda}_{\forall}$, the existential quantification is usually encoded as $\tilde{\exists} \mathsf{X}.\mathsf{T} =_{\mathrm{def}} \forall \mathsf{Y}.(\forall \mathsf{X}.\mathsf{T} \Rightarrow \mathsf{Y}) \Rightarrow \mathsf{Y}$ where $\mathsf{Y} \notin \mathsf{T}$. Using Thm 4.6, we define terms t, u such that for reducibility candidates and biorthogonals, we have $\mathsf{t} \Vdash (\exists \mathsf{X}.\mathsf{T}) \Rightarrow (\tilde{\exists} \mathsf{X}.\mathsf{T})$ and $\mathsf{u} \Vdash (\tilde{\exists} \mathsf{X}.\mathsf{T}) \Rightarrow (\exists \mathsf{X}.\mathsf{T})$ (Thm. 5.9). This means that in the corresponding realizability models, the let makes the two existential quantifications equivalent w.r.t. provability.

We assume given a rewrite relation $\to_{\mathsf{R}}$ on $\Lambda(\Sigma)$, which contains $\to_{\beta}$ and which is compatible with $\to_{\beta}$ in the following sense: if $(\lambda \mathsf{x}.\mathsf{t})\mathsf{u} \to_{\mathsf{R}} w$ and w is not of the form $(\lambda \mathsf{x}.\mathsf{t}')\mathsf{u}'$ with $(\mathsf{t}, \mathsf{u}) \to_{\mathsf{R}} (\mathsf{t}', \mathsf{u}')$, then $w = \mathsf{t}[\mathsf{u}/\mathsf{x}]$. We also assume given a set $\mathcal{E}$ of elimination contexts for $\rightarrow_{\mathsf{R}}$ such that $[\,]\,\mathsf{t} \in \mathcal{E}$ for all $\mathsf{t} \in \Lambda(\Sigma)$ and such that $(\lambda \mathsf{x}.\mathsf{t})\mathsf{u}$ is neutral for all $\mathsf{t}, \mathsf{u} \in \Lambda(\Sigma)$. Note that this include the reduction systems presented in Sect. 5.1 and Sect. 5.2.

We obtain λ-terms with let and extend $\rightarrow_{\mathsf{R}}$ with $\rightarrow_{\mathcal{V}}$. Elimination contexts are extended to $\mathcal{E}_{\mathsf{let}}$. We thus obtain reducibility families $\mathcal{CR}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}}$ and $\mathcal{CR}^{\perp\!\!\perp}_{\mathsf{R}\mathcal{V}\mathcal{E}_{\mathsf{let}}}$, that we write $\mathcal{CR}$ and $\mathcal{CR}^{\perp\!\!\perp}$ respectively. Note that compatibility with $\rightarrow_{\beta}$ is preserved: if $(\lambda \mathsf{x}.\mathsf{t})\mathsf{u} \rightarrow_{\mathsf{R}\mathcal{V}} w$ then either $w = \mathsf{t}[\mathsf{u}/\mathsf{x}]$ or $w = (\lambda \mathsf{x}.\mathsf{t}')\mathsf{u}'$ with $(\mathsf{t}, \mathsf{u}) \rightarrow_{\mathsf{R}\mathcal{V}} (\mathsf{t}', \mathsf{u}')$. This ensures that if $\mathsf{t}[\mathsf{u}/\mathsf{x}] \in [\![\mathsf{T}]\!]^{\mathcal{R}ed}_{\rho}$ for all $\mathsf{u} \in [\![\mathsf{U}]\!]^{\mathcal{R}ed}_{\rho}$ then $\mathsf{t} \in [\![\mathsf{U} \Rightarrow \mathsf{T}]\!]^{\mathcal{R}ed}_{\rho}$. Moreover, we still have $[\,]\,\mathsf{t} \in \mathcal{E}_{\mathsf{let}}$ for all $\mathsf{t} \in \Lambda(\Sigma)_{\mathsf{let}}$

The formulas we consider are the types $\mathcal{T}_{\forall\exists}$. They are interpreted as in Sect. 5.2, using Thm. 4.6. In particular $[\![\exists \mathsf{X}.\mathsf{T}]\!]^{\mathcal{R}ed}_{\rho} = \bigvee_{\mathcal{R}ed} \left\{ [\![\mathsf{T}]\!]^{\mathcal{R}ed}_{\rho[\mathsf{C}/\mathsf{X}]} \mid \mathsf{C} \in \mathcal{R}ed \right\}$.

Definition 5.8. *Given a type* T*, let* $\Box(\mathsf{T}) =_{\mathrm{def}} \forall \mathsf{X}.(\mathsf{T} \Rightarrow \mathsf{X}) \Rightarrow \mathsf{X}$*, where* $\mathsf{X} \notin \mathsf{T}$.

Note that the boxed type $\Box(\mathsf{T})$ is not the double-negation of T. Boxed types are useful because of the following intermediate properties: given $\mathcal{R}ed \in \{\mathcal{CR}, \mathcal{CR}^{\perp\!\!\perp}\}$,

$$\begin{array}{rcl}
\lambda \mathsf{x}.\mathsf{x} & \Vdash_{\mathcal{R}ed} & (\tilde{\exists} \mathsf{X}.\mathsf{T}) \Rightarrow \Box(\exists \mathsf{X}.\mathsf{T}) \\
\lambda \mathsf{x}.\lambda \mathsf{y}.\mathsf{y}\mathsf{x} & \Vdash_{\mathcal{R}ed} & \mathsf{T} \Rightarrow \Box(\mathsf{T}) \\
\lambda \mathsf{x}.\mathsf{x}(\lambda \mathsf{y}.\mathsf{y}) & \Vdash_{\mathcal{R}ed} & \Box(\mathsf{T}) \Rightarrow \mathsf{T} \\
\lambda \mathsf{t}.\lambda \mathsf{x}.\mathsf{t}\ (\lambda \mathsf{y}.(\mathsf{let}\ \mathsf{z} = \mathsf{y}\ \mathsf{in}\ \mathsf{x}\mathsf{z})) & \Vdash_{\mathcal{R}ed} & \Box(\exists \mathsf{X}.\mathsf{T}) \Rightarrow (\tilde{\exists} \mathsf{X}.\mathsf{T})
\end{array}$$

Theorem 5.9. *Let* $\mathcal{R}ed \in \{\mathcal{CR}, \mathcal{CR}^{\perp\!\!\perp}\}$.

(i) $\lambda \mathsf{y}.\lambda \mathsf{x}.(\mathsf{let}\ \mathsf{x} = \mathsf{y}\ \mathsf{in}\ \mathsf{x}\mathsf{z}) \Vdash_{\mathcal{R}ed} (\exists \mathsf{X}.\mathsf{T}) \Rightarrow (\tilde{\exists} \mathsf{X}.\mathsf{T})$
(ii) $\lambda \mathsf{x}.\mathsf{x}(\lambda \mathsf{y}.\mathsf{y}) \Vdash_{\mathcal{R}ed} (\tilde{\exists} \mathsf{X}.\mathsf{T}) \Rightarrow (\exists \mathsf{X}.\mathsf{T})$

6 Conclusion

We proposed a let-elimination of union types and implicit existential quantifications. This provides a way to obtain strongly normalizing systems, and for the existential quantification, to get subject reduction as well. We also have shown that the obtained existential quantification coincides with its usual encoding w.r.t. provability in realizability models built from reducibility candidates.

Further Work. There are different way in which this work can be extended. First, to study subject reduction of union types with let. Second, to extend the reduction of let with usual permutative conversions. Third, to study the obtained existential quantification in Krivine's realizability [10]. Another direction is to explore links with classical logic: the obtained let seems to correspond to a form of $\tilde{\mu}$ in the sequent-based λ-calculus of [4]. The elimination rules with let would be seen as the translation on terms of implicit right-introduction rules on contexts.

Acknowledgments. We thank Alexandre Miquel for suggesting the study of realizability models and the use of boxed types.

References

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

[2] F. Blanqui and C. Riba. Combining Typing and Size Constraints for Checking the Termination of Higher-Order Conditional Rewrite Systems. In *Proceedings of LPAR'06*, volume 4246 of *LNAI*, 2006.

[3] T. Coquand and A. Spiwack. A Proof of Strong Normalisation using Domain Theory. In *Proceedings of LiCS'06*, pages 307–316. IEEE Computer Society, 2006.

[4] P.-L. Curien and H. Herbelin. The Duality of Computation. In *Proceedings of ICFP'00*, pages 233–243. ACM, 2000.

[5] M. Dezani-Ciancaglini, U. de' Liguoro, and P. Piperno. A Filter Model for Concurrent Lambda-Calculus. *Siam Journal on Computing*, 27(5):1376–1419, 1998.

[6] J. Dunfield and F. Pfenning. Tridirectional Typechecking. In *Proceedings of POPL'04*, pages 281–292. ACM, 2004.

[7] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[8] H. Ishihara and T. Kurata. Completeness of Intersection and Union Type Assignment Systems for Call-by-Value λ-Models. *Theoretical Computer Science*, 272:197–221, 2002.

[9] J.-L. Krivine. *Lambda-Calculus, Types and Models*. Ellis Horwood, 1993.

[10] J.-L. Krivine. Realizability in Classical Logic. To appear in *Panoramas et synthèses*, Société Mathématique de France, disponible sur HAL, 2004.

[11] S. Lindley and I. Stark. Reducibility and TT-Lifting for Computation Types. In *TLCA*, volume 3461 of *LNCS*, pages 262–277. Springer, 2005.

[12] P. Oliva and T. Streicher. On Krivine's Realizability Interpretation of Classical Second-Order Arithmetic. *Fundameta Informaticae*, 84(2):207–220, 2008.

[13] M. Parigot. Proofs of Strong Normalization for Second Order Classical Natural Deduction. *Journal of Symbolic Logic*, 62(4):1461–1479, 1997.

[14] C. Riba. Strong Normalization as Safe Interaction. In *Proceedings of LiCS'07*, pages 13–22. IEEE Computer Society, 2007.

[15] C. Riba. Stability by Union of Reducibility Candidates for Orthogonal Constructor Rewriting. In *Proceedings of CiE'08*, volume 5028 of *LNCS*, pages 498–510. Springer, 2008.

[16] C. Riba. Toward a General Rewriting-Based Framework for Reducibility. Submitted (available from the author's homepage), 2008.

[17] W. W. Tait. A Realizability Interpretation of the Theory of Species. In R. Parikh, editor, *Logic Colloquium*, volume 453 of *LNCS*, pages 240–251, 1975.

[18] M. Tatsuta. Simple Saturated Sets for Disjunction and Second-Order Existential Quantification. In *Proceedings of TLCA'07*, volume 4583 of *LNCS*, pages 366–380. Springer, 2007.

[19] J. Vouillon and P.-A. Melliès. Semantic Types: A Fresh Look at the Ideal Model for Types. In *Proceedings of POPL'04*. ACM, 2004.