HAL open science

Union of Reducibility Candidates for Orthogonal Constructor Rewriting

Colin Riba

To cite this version:

Colin Riba. Union of Reducibility Candidates for Orthogonal Constructor Rewriting. 2008. inria-00204710

HAL Id: inria-00204710

https://inria.hal.science/inria-00204710v1

Preprint submitted on 15 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Union of Reducibility Candidates for Orthogonal Constructor Rewriting

Colin Riba

Projet Everest
INRIA Sophia Antipolis*
Colin.Riba@sophia.inria.fr

Abstract. We revisit Girard's reducibility candidates by proposing a general of the notion of neutral terms. They are the terms which do not interact with some contexts called elimination contexts. We apply this framework to constructor rewriting, and show that for orthogonal constructor rewriting, Girard's reducibility candidates are stable by union.

1 Introduction

The most flexible termination proof methods for various extensions of typed λ-calculi use type interpretations [2, 5, 6, 7, 16]. Among them we distinguish three families: Girard's reducibility candidates [10], Tait's saturated sets [20], and interpretations based on biorthogonality [11, 16]. An interesting way to compare different type interpretations is to study their stability by union. This is even a necessary property in some cases [2, 7, 21].

This paper concerns the extension of the simply-typed λ-calculus with constructor rewriting. We are not interested in termination criteria by themselves, but by the investigation of the closure properties of types interpretations that allow to *formulate* different termination criteria. We focus on Girard's reducibility candidates and their stability by union.

We give a generalization of the notion of neutral terms that allows to define Girard's reducibility candidates in a generic way. Neutral terms are the terms that do not interact with some contexts called elimination contexts. Terms which are not neutral are *observable* since they interact with some elimination contexts. We call them *values*. We instantiate this framework with constructor rewriting. In order to get interesting values, we use elimination contexts with *destructors* to eliminate the constructors.

Next, we study the question of stability by union. By instantiating the condition of [18], we show that reducibility candidates are stable by union for orthogonal constructor rewriting. The proof uses a result of [13] on the existence of external redexes for orthogonal CCERSs.

The paper is organized as follows. We expose our notations in Sec. 2.

* 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex, France.

Section 3 presents a general definition of reducibility families and type interpretations. We apply it by briefly discussing Tait's saturated sets for the pure λ-calculus and one possible extension to deal with rewriting.

Section 4 is devoted to Girard's reducibility candidates, of which we suggest a generalization in Sec. 4.1. We instantiate it in Sec. 4.2 to the framework of λ-calculus plus constructor rewriting.

We then discuss stability by union in Sec. 5. We first briefly present the key problems and known results. Section 5.1 recalls a necessary and sufficient condition for the stability by union of Girard's candidates. In Sec. 5.2, we show that this condition is met with *orthogonal* constructor rewriting.

We assume familiarity with typed λ-calculus [4], reducibility [9, 14] and rewriting [22]. Concerning CCERSs, we refer to [12]. The paper (except Sec. 5.2) is based on parts of the Phd thesis of the author [17] (in French).

2 Simply Typed λ-Calculus with Constructor Rewriting

Given a set A, $\vec{a}$ denotes a finite sequence of elements of A of length $|\vec{a}|$.

Terms and Types. We consider λ-terms with uncurried symbols $\mathtt{f}$ in a signature $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ and variables $x \in \mathcal{X}$:

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

where $\mathtt{f} \in \Sigma_n$. 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 the term t is written $t\sigma$ or $t[\sigma(x_1)/x_1, \ldots, \sigma(x_n)/x_n]$ if $\mathcal{D}\mathrm{om}(\sigma) = \{x_1, \ldots, x_n\}$.

Given *base types* $\mathtt{B} \in \mathcal{B}$, simple types are defined as usual:

$$T, U \in \mathcal{T}(\mathcal{B}) \quad ::= \quad \mathtt{B} \quad | \quad U \Rightarrow T\ .$$

Typing contexts are functions Γ of finite domain from variables to types, written $x_1 : T_1, \ldots, x_n : T_n$. Given a *type assignment* $\tau : \Pi_{n\in\mathbb{N}}.\Sigma_n \to \mathcal{T}(\mathcal{B})^{n+1}$, the *typing relation* $\Gamma \vdash_\tau t : T$ is inductively defined by the following rules:

$$(\textsc{Ax})\ \frac{}{\Gamma, x : T \vdash_\tau x : T}$$

$$(\textsc{Symb})\ \frac{\Gamma \vdash_\tau t_1 : T_1 \quad \ldots \quad \Gamma \vdash_\tau t_n : T_n}{\Gamma \vdash_\tau \mathtt{f}(t_1, \ldots, t_n) : T}\ \tau(\mathtt{f}) = (T_1, \ldots, T_n, T)$$

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

Constructor Rewriting. Assume given a set $\mathcal{C} \subseteq \Sigma$ of *constructor symbols* $\mathtt{c}$ of type $(\vec{T}, \mathtt{B})$ with $\mathtt{B} \in \mathcal{B}$. For normalization, $\mathtt{B}$ must occur only at positive positions in $\vec{T}$ [15]. As we are not interested in strong normalization *conditions*, we do not care of this restriction here.

A *constructor rewrite system* (or *rewrite system*) on $\mathcal{C}$ is a set $\mathcal{R}$ of rewrite rules $\mathtt{f}(\vec{l}) \mapsto_{\mathcal{R}} r$ such that $r \in \Lambda(\Sigma)$, $\mathsf{FV}(r) \subseteq \mathsf{FV}(\mathtt{f}(\vec{l}))$, $\mathtt{f} \in \Sigma \backslash \mathcal{C}$ (defined symbols are not constructors) and $\vec{l}$ are terms of the grammar

$$p \quad ::= \quad x \quad | \quad c(p_1, \ldots, p_n) \ ,$$

where $c \in \mathcal{C}$ (hence, $\vec{l}$ are patterns).

A rewrite system $\mathcal{R}$ is *typed* if for each rewrite rule $\mathtt{f}(\vec{l}) \mapsto_{\mathcal{R}} r$ with $\tau(\mathtt{f}) = (\vec{T}, T)$, there exists a (necessarily unique) context Γ with $\mathcal{D}om(\Gamma) = \mathsf{FV}(\mathtt{f}(\vec{l}))$ such that

$$\Gamma \vdash_\tau \mathtt{f}(\vec{l}) : T \qquad \text{and} \qquad \Gamma \vdash_\tau r : T \ .$$

Example 2.1. We consider the type `Nat` of Peano's numbers, with constructors $\mathtt{0} : \mathtt{Nat}$ and $\mathtt{S} : (\mathtt{Nat}, \mathtt{Nat})$. The following system, defining addition, is a constructor rewrite system:

$$\mathtt{plus}(x, \mathtt{0}) \quad \mapsto \quad x \qquad\qquad \mathtt{plus}(x, \mathtt{S}(y)) \quad \mapsto \quad \mathtt{plus}(\mathtt{S}(x), y) \ .$$

Reductions. A rewrite relation is a binary relation $\to_R \subseteq (\Lambda(\Sigma) \setminus \mathcal{X}) \times \Lambda(\Sigma)$ which is stable by contexts and substitutions. Let $(t)_R =_{\text{def}} \{v \mid t \to_R v\}$ and write $(t_1, \ldots, t_n) \to_R (u_1, \ldots, u_n)$ for the product extension of $\to_R$. A term t is R-*reducible* (or *reducible*) if $(t)_R \neq \emptyset$. We denote by $\mathcal{SN}_R$ the set of *strongly normalizing* terms for $\to_R$. Note that it is the smallest set of terms such that

$$\forall t. \quad (\forall u. \quad t \to_R u \quad \Longrightarrow \quad u \in \mathcal{SN}_R) \quad \Longrightarrow \quad t \in \mathcal{SN}_R \ .$$

Given a constructor rewrite system $\mathcal{R}$, we let $\to_{\beta\mathcal{R}}$ be the smallest rewrite relation on $\Lambda(\Sigma)$ containing $\mapsto_{\mathcal{R}}$ and β-reduction: $(\lambda x.t)u \mapsto_\beta t[u/x]$.

3 Reducibility Families

In this section, we present a general notion of reducibility family and of type interpretation. We then briefly take a look at their instantiation to deal with the pure λ-calculus and with the combination of λ-calculus with rewriting.

Definition 3.1. *Let $\to_R$ be a rewrite relation on $\Lambda(\Sigma)$.*

(i) The function space *is the function $_ \Rightarrow _ : \mathcal{P}(\Lambda(\Sigma))^2 \to \mathcal{P}(\Lambda(\Sigma))$ defined as*

$$A \Rightarrow B \quad =_{def} \quad \{t \mid \forall u. \quad u \in A \quad \Longrightarrow \quad t\,u \in B\} \ .$$

(ii) A reducibility family *for $\to_R$ is a set of sets $\mathcal{R}\mathrm{ed} \subseteq \{A \mid \mathcal{X} \subseteq A \subseteq \mathcal{SN}_R\}$ which is stable by intersections and by the function space.*

(iii) A type interpretation *in $\mathcal{R}\mathrm{ed}$ is a map $[\![_]\!] : \mathcal{T}(\mathcal{B}) \to \mathcal{R}\mathrm{ed}$ such that for all $T, U \in \mathcal{T}(\mathcal{B})$ we have $[\![U \Rightarrow T]\!] = [\![U]\!] \Rightarrow [\![T]\!]$.*

(iv) A type interpretation $[\![_]\!]$ is adequate *if for all Γ, t, T and σ we have*

$$\big(\Gamma \vdash_\tau t : T \quad \wedge \quad \sigma \models_{[\![_]\!]} \Gamma\big) \quad \Longrightarrow \quad t\sigma \in [\![T]\!] \ ,$$

where $\sigma \models_{[\![_]\!]} \Gamma$ iff $\sigma(x) \in [\![\Gamma(x)]\!]$ for all $x \in \mathcal{D}om(\Gamma)$.

Pure λ*-Calculus.* Let $\mathcal{R}\mathtt{ed}$ be a reducibility family and $[\![_]\!] : \mathcal{T}(\mathcal{B}) \to \mathcal{R}\mathtt{ed}$ be a type interpretation. Let us see, in the case of the pure λ-calculus, some sufficient conditions to ensure that $[\![_]\!]$ is adequate. As usual, we reason by induction on $\Gamma \vdash \mathsf{t} : \mathsf{T}$ and by cases on the last applied typing rule.

We only have to check the rules (Ax), ($\Rightarrow$ E) and ($\Rightarrow$ I). The rule (Ax) is trivial while ($\Rightarrow$ E) is dealt with by definition of the function space $_ \Rightarrow _$. Concerning the rule ($\Rightarrow$ I), it is sufficient that for all $\mathsf{A} \in \mathcal{R}\mathtt{ed}$,

$$\forall \mathsf{t}, \mathsf{u} \in \Lambda. \quad (\mathsf{t}[\mathsf{u}/\mathsf{x}] \in \mathsf{A} \quad \wedge \quad \mathsf{u} \in \mathcal{SN}_\beta) \quad \Longrightarrow \quad (\lambda \mathsf{x}.\mathsf{t})\mathsf{u} \in \mathsf{A} \ . \tag{1}$$

As for $\mathcal{X} \subseteq \mathsf{A} \subseteq \mathcal{SN}_\beta$, condition (1) has to be preserved by $_ \Rightarrow _ : \mathcal{R}\mathtt{ed}^2 \to \mathcal{R}\mathtt{ed}$. We can conveniently formulate this by using *elimination contexts* [1]. For the pure λ-calculus, they are defined by the grammar

$$\mathsf{E}[\ ] \in \mathcal{E}_\Rightarrow \quad ::= \quad [\ ] \quad | \quad \mathsf{E}[\ ]\,\mathsf{t} \ .$$

Then, we get the following clauses: for all $\mathsf{E}[\ ] \in \mathcal{E}$, all $\mathsf{x} \in \mathcal{X}$ and all $\mathsf{t}, \mathsf{u} \in \Lambda$,

$$\mathsf{E}[\ ] \in \mathcal{SN}_\beta \quad \Longrightarrow \quad \mathsf{E}[\mathsf{x}] \in \mathsf{A} \ , \tag{2}$$

$$(\mathsf{E}[\mathsf{t}[\mathsf{u}/\mathsf{x}]] \in \mathsf{A} \quad \wedge \quad \mathsf{u} \in \mathcal{SN}_\beta) \quad \Longrightarrow \quad \mathsf{E}[(\lambda \mathsf{x}.\mathsf{t})\mathsf{u}] \in \mathsf{A} \ . \tag{3}$$

The sets $\mathsf{A} \subseteq \mathcal{SN}_\beta$ satisfying (2) and (3) are *Tait's saturated sets* [20]. The set of saturated sets, denoted by $\mathcal{SAT}_\beta$, forms a reducibility family. Showing that $\mathcal{SAT}_\beta$ is not empty amounts to show that $\mathcal{SN}_\beta \in \mathcal{SAT}_\beta$. In this case, properties (2) and (3) are consequences of two important facts.

First, a reduction step from a term of the form $\mathsf{E}[(\lambda \mathsf{x}.\mathsf{t})\mathsf{u}]$ (resp. $\mathsf{E}[\mathsf{x}]$) occurs either in the elimination context $\mathsf{E}[\ ]$ or in the term $(\lambda \mathsf{x}.\mathsf{t})\mathsf{u}$, but involves no interaction between them:

$$\forall \mathsf{v}. \quad \mathsf{E}[\mathsf{x}] \to_\beta \mathsf{v} \quad \Longrightarrow \quad (\mathsf{v} = \mathsf{E}'[\mathsf{x}] \ \text{ with } \ \mathsf{E}[\ ] \to_\beta \mathsf{E}'[\ ]) \tag{4}$$

$$\forall \mathsf{v}. \ \mathsf{E}[(\lambda \mathsf{x}.\mathsf{t})\mathsf{u}] \to_\beta \mathsf{v} \quad \Longrightarrow \quad (\mathsf{v} = \mathsf{E}'[\mathsf{s}] \ \text{ with } \ (\mathsf{E}[\ ], (\lambda \mathsf{x}.\mathsf{t})\mathsf{u}) \to_\beta (\mathsf{E}'[\ ], \mathsf{s})) \tag{5}$$

Second, property (3) follows from (5) and the fact that $(\lambda \mathsf{x}.\mathsf{t})\mathsf{u} \in \mathcal{SN}_\beta$ as soon as $\mathsf{t}[\mathsf{u}/\mathsf{x}] \in \mathcal{SN}_\beta$ and $\mathsf{u} \in \mathcal{SN}_\beta$. This property holds in turn thanks to the *Weak Standardization Lemma*, which was used in [3] for extensions of the Calculus of Constructions. This is obvious for the pure λ-calculus.

Lemma 3.2 (Weak Standardization). *A reduct of a* β*-redex* $(\lambda \mathsf{x}.\mathsf{t})\mathsf{u}$ *is either* $\mathsf{t}[\mathsf{u}/\mathsf{x}]$ *or a* β*-redex* $(\lambda \mathsf{x}.\mathsf{t}')\mathsf{u}'$ *with* $(\mathsf{t}, \mathsf{u}) \to_\beta (\mathsf{t}', \mathsf{u}')$.

λ*-Calculus with Rewriting.* To deal with rewriting, we must consider the rule (Symb). For constructor, we have to use specific interpretations of base types (eg. using inductive types, as in [6]). We concentrate on symbols $\mathtt{f} \in \Sigma \setminus \mathcal{C}$. Given $\mathtt{f}$ of type $(\vec{\mathsf{T}}, \mathsf{T})$ and $\vec{\mathsf{t}} \in [\![\vec{\mathsf{T}}]\!]$, we have to make sure that $\mathtt{f}(\vec{\mathsf{t}}) \in [\![\mathsf{T}]\!]$. Sufficient conditions for this are given by *termination criteria*, a subject that we do not treat in this paper (see for instance [6, 1, 2, 5, 7]).

Here, we are interested in the exploration of reducibility families that allow to *formulate* termination criteria. As for the λ-calculus, we can use a non-interaction property similar to (5):

$$\forall v.\quad \mathsf{E}[\mathtt{f}(\vec{t})] \to_{\beta\mathcal{R}} v \quad\Longrightarrow\quad \big(v = \mathsf{E}'[s] \ \text{ with } \ (\mathsf{E}[\ ], \mathtt{f}(\vec{t})) \to_{\beta\mathcal{R}} (\mathsf{E}'[\ ], s)\big)\ . \tag{6}$$

But rewrite systems do not satisfy in general the weak standardization lemma. Therefore, in order to get $\mathtt{f}(\vec{t}) \in \mathcal{SN}_{\beta\mathcal{R}}$, we need $v \in \mathcal{SN}_{\beta\mathcal{R}}$ *for all* v such that $\mathtt{f}(\vec{t}) \to_{\beta\mathcal{R}} v$. This is subsumed by the clause

$$\big(\forall v.\quad \mathsf{E}[\mathtt{f}(\vec{t})] \to_{\beta\mathcal{R}} v \quad\Longrightarrow\quad v \in \mathsf{A}\big) \quad\Longrightarrow\quad \mathsf{E}[\mathtt{f}(\vec{t})] \in \mathsf{A}\ . \tag{7}$$

In this case, we also need saturated sets to be stable by reduction: if $t \in \mathsf{A}$ and $t \to_{\beta\mathcal{R}} u$ then $u \in \mathsf{A}$.

4 Neutral Terms and Reducibility Candidates

We now turn to Girard's reducibility candidates [10]. They form a reducibility family in which properties (3) and (7) can be formulated in a uniform and elegant way. This is due to *neutral terms*, that enjoy non-interaction properties such as (4), (5) and (6).

We first give a general formulation, and then apply it to constructor rewriting.

4.1 A General Formulation

We give a generalization of the original notion of neutral terms that allows to define reducibility candidates in a generic way. The key idea is that neutral terms are the terms that do not interact with *elimination contexts*. In the whole section, we assume given a rewrite relation $\to_{\mathsf{R}}$.

Elimination contexts will be defined as a special case of *evaluation contexts*.

Definition 4.1 (Evaluation Contexts). *Let* $[\ ] \in \mathcal{X}$ *be a distinguished variable.* A set of evaluation contexts for $\to_{\mathsf{R}}$ *is a set* $\mathcal{E}$ *of terms* $\mathsf{E}[\ ]$ *linear in* $[\ ]$*, which is*

(i) stable by reduction: if $\mathsf{E}[\ ] \in \mathcal{E}$ *and* $\mathsf{E}[\ ] \to_{\mathsf{R}} t$ *then* $t = \mathsf{F}[\ ] \in \mathcal{E}$*;*
(ii) stable by composition: if $\mathsf{E}[\ ] \in \mathcal{E}$ *and* $\mathsf{F}[\ ] \in \mathcal{E}$ *then* $\mathsf{E}[\mathsf{F}[\ ]] \in \mathcal{E}$*, where* $\mathsf{E}[t] =_{def} \mathsf{E}[\ ](t/[\ ])$.

We now assume given a set $\mathcal{E}$ of evaluation contexts for $\to_{\mathsf{R}}$.

Definition 4.2 (Neutral Terms). A neutral term for $\to_{\mathsf{R}}$ in $\mathcal{E}$ *is a term* t *such that for all* $\mathsf{E}[\ ] \in \mathcal{E}$*,*

$$\forall v.\quad \mathsf{E}[t] \to_{\mathsf{R}} v \quad\Longrightarrow\quad (v = \mathsf{E}'[s] \quad \textit{with} \quad (\mathsf{E}[\ ], t) \to_{\mathsf{R}} (\mathsf{E}'[\ ], s))\ .$$

We denote by $\mathcal{N}_{\mathsf{R}\mathcal{E}}$ *the set of neutral terms for* $\to_{\mathsf{R}}$ *in* $\mathcal{E}$*.*

The terms that are not neutral interact with evaluation contexts. They are therefore observable, and we think of them as being values.

Definition 4.3 (Values). A value *for* $\to_{\mathsf{R}}$ *in* $\mathcal{E}$ *is a term which is not neutral. We denote by* $\mathcal{V}_{\mathsf{R}\mathcal{E}}$ *the set of values for* $\to_{\mathsf{R}}$ *in* $\mathcal{E}$.

Example 4.4. In the pure λ-calculus, taking $\mathcal{E}_{\Rightarrow}$ as evaluation contexts, the values are exactly the terms of the form $\lambda x.t$.

To build reducibility candidates, we are interested in neutral terms and evaluation contexts that enjoy some properties. This leads to the notion of *elimination contexts*.

Definition 4.5 (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 all variables are neutral* ($\mathcal{X} \subseteq \mathcal{N}_{\mathsf{R}\mathcal{E}}$), *and if for all* $t \in \mathcal{N}_{\mathsf{R}\mathcal{E}}$ *and all* $E[\ ] \in \mathcal{E}$ *we have* $E[t] \in \mathcal{N}_{\mathsf{R}\mathcal{E}}$.

Example 4.6. For the pure λ-calculus, $\mathcal{E}_{\Rightarrow}$ is a set of elimination contexts.

We now define reducibility candidates in the usual way: our generalization regards neutral terms and their definition using elimination contexts. Assume that $\mathcal{E}$ is a set of elimination contexts for $\to_{\mathsf{R}}$.

Definition 4.7 (Reducibility Candidates). *The set* $\mathcal{CR}_{\mathsf{R}\mathcal{E}}$ *of* reducibility candidates *for* $\to_{\mathsf{R}}$ *in* $\mathcal{E}$ *is the set of all* $C \subseteq \mathcal{SN}_{\mathsf{R}}$ *such that*

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

Note that $\mathcal{CR}_{\mathsf{R}\mathcal{E}}$ is a complete lattice for $\subseteq$ whose top element is $\mathcal{SN}_{\mathsf{R}}$. In order to verify that $\mathcal{X}$ is contained in any candidate, it is interesting to look at the least reducibility candidate. This is $\mathcal{HN}_{\mathsf{R}\mathcal{E}}$, the set of *hereditary neutral terms*, defined as the smallest set of terms such that

$$\forall t \in \mathcal{N}_{\mathsf{R}\mathcal{E}}.\quad (\forall u.\quad t \to_{\mathsf{R}} u \quad\Longrightarrow\quad u \in \mathcal{HN}_{\mathsf{R}\mathcal{E}}) \quad\Longrightarrow\quad t \in \mathcal{HN}_{\mathsf{R}\mathcal{E}}\ .$$

Since variables are neutral terms in normal form, we have $\mathcal{X} \subseteq \mathcal{HN}_{\mathsf{R}\mathcal{E}} \subseteq C$ for every $C \in \mathcal{CR}_{\mathsf{R}\mathcal{E}}$.

When $\mathcal{E}$ contains contexts of the form $[\ ]\,t$ with $t \in \mathcal{SN}_{\mathsf{R}}$, reducibility candidates form a reducibility family. To show this, it remains to see that $_ \Rightarrow _$ is a function from $\mathcal{CR}^2_{\mathsf{R}\mathcal{E}}$ to $\mathcal{CR}_{\mathsf{R}\mathcal{E}}$. This follows from a simple but fundamental consequence of the non-interaction between neutral terms and evaluation contexts:

Lemma 4.8. *Let* $t \in \mathcal{N}_{\mathsf{R}\mathcal{E}}$ *and* $E[\ ] \in \mathcal{E} \cap \mathcal{SN}_{\mathsf{R}}$. *Then, for all* $C \in \mathcal{CR}_{\mathsf{R}\mathcal{E}}$,

$$(\forall u.\quad t \to_{\mathsf{R}} u \quad\Longrightarrow\quad E[u] \in C) \quad\Longrightarrow\quad E[t] \in C\ .$$

Proof. Since $E[t] \in \mathcal{N}_{\mathsf{R}\mathcal{E}}$, we have to show that $(E[t])_{\mathsf{R}} \subseteq C$. We have $t \in \mathcal{SN}_{\mathsf{R}}$ because $(t)_{\mathsf{R}} \subseteq C \subseteq \mathcal{SN}_{\mathsf{R}}$, and moreover we have $E[\ ] \in \mathcal{SN}_{\mathsf{R}}$ by assumption. We can therefore reason by induction on pairs $(E[\ ], t)$ ordered by $\to_{\mathsf{R}}$.

Let v such that $E[t] \to_{\mathsf{R}} v$. Since t is neutral, we have $v = E'[t']$ with $(E[\ ], t) \to_{\mathsf{R}} (E'[\ ], t')$, and there are two cases.

Case of $E[\,] \to_R E'[\,]$. Note that $E'[\,] \in \mathcal{E}$. For all $u \in (t)_R$, since $E[u] \to_R E'[u]$ and $E[u] \in C$, we have $E'[u] \in C$ by ($\mathcal{CR}0$). Hence, we can apply the induction hypothesis on $(E'[\,], t)$ and we conclude that $E'[t] \in C$.

Case of $t \to_R t'$. In this case, we have $E[t'] \in C$ by assumption. □

Proposition 4.9. *If $\mathcal{E}$ contains $[\,]\,t$ for all $t \in \Lambda(\Sigma)$, then $\mathcal{CR}_{R\mathcal{E}}$ is a reducibility family.*

Proof. It remains to show that $_ \Rightarrow _ : \mathcal{CR}^2_{R\mathcal{E}} \to \mathcal{CR}_{R\mathcal{E}}$. Let $A, B \in \mathcal{CR}_{R\mathcal{E}}$.

First, we have $A \Rightarrow B \subseteq \mathcal{SN}_R$: for all $t \in A \Rightarrow B$, since $\mathcal{X} \subseteq A$ we have $tx \in B \subseteq \mathcal{SN}_R$, hence $t \in \mathcal{SN}_R$.

Let us now check the clauses ($\mathcal{CR}0$) and ($\mathcal{CR}1$).

($\mathcal{CR}0$) Let $t \in A \Rightarrow B$ and $u \in (t)_R$. For all $a \in A$, we have $ta \in B$, hence $ua \in B$ by ($\mathcal{CR}0$) applied to B. It follows that $u \in A \Rightarrow B$.

($\mathcal{CR}1$) Let $t \in \mathcal{N}_{R\mathcal{E}}$ such that $(t)_R \subseteq A \Rightarrow B$ and let $a \in A$. For all $u \in (t)_R$, we have $ua \in B$. Since $[\,]\,a \in \mathcal{SN}_R$, it follows from Lem. 4.8 that $ta \in B$. We conclude that $t \in A \Rightarrow B$. □

Example 4.10. For the pure λ-calculus, $\mathcal{CR}_{\beta\mathcal{E}_\Rightarrow}$ is the usual set of reducibility candidates. In particular, each $C \in \mathcal{CR}_{\beta\mathcal{E}_\Rightarrow}$ satisfies property (3).

4.2 Application to Constructor Rewriting

Let $\mathcal{R}$ be a constructor rewrite system on $\mathcal{C}$. If we use elimination contexts of the form $\mathcal{E}_\Rightarrow$, then the values are the terms of the form $\lambda x.t$.

However, we would like to build values from constructors. This is particularly useful with inductive types [6]. According to Def. 4.3, we have to make them observable. To this end, we introduce appropriate *destructors* in elimination contexts. To each $c \in \mathcal{C}$ of type $(\vec{T}, B)$ with $|\vec{T}| > 0$ and each $i \in \{1, \ldots, |\vec{T}|\}$, we associate a new unary *destructor symbol* $d_{c,i}$ defined by the rewrite rule

$$d_{c,i}(c(x_1, \ldots, x_n)) \quad \mapsto_{\mathcal{D}} \quad x_i \ .$$

Let $\mho$ be a new nullary symbol. For the elimination of a nullary constructor c, we use a new unary destructor d_c defined by the rewrite rule

$$d_c(c) \quad \mapsto_{\mathcal{D}} \quad \mho \ .$$

Lemma 4.11. *Let $\mathcal{E}_{\Rightarrow\mathcal{C}}$ be the set of terms defined by the grammar*

$$E[\,] \in \mathcal{E}_{\Rightarrow\mathcal{C}} \quad ::= \quad [\,] \quad | \quad E[\,]\,t \quad | \quad d(E[\,]) \ ,$$

where d is a destructor of a constructor of $\mathcal{C}$. Then,

(i) $\mathcal{E}_{\Rightarrow\mathcal{C}}$ is a set of evaluation contexts for $\to_{\beta\mathcal{RD}}$.

(ii) The values in $\mathcal{E}_{\Rightarrow\mathcal{C}}$ for $\to_{\beta\mathcal{RD}}$ are exactly the terms of the form
- *$\lambda x.t$; or*
- *$c(\vec{t})$ with $c \in \mathcal{C}$.*

(iii) $\mathcal{E}_{\Rightarrow\mathcal{C}}$ *is a set of elimination contexts for* $\to_{\beta\mathcal{RD}}$.
(iv) Each $\mathsf{C} \in \mathcal{CR}_{\beta\mathcal{RD}\mathcal{E}_{\Rightarrow\mathcal{C}}}$ *satisfies properties (3) and (7).*

Proof. (i) It is clear that $\mathcal{E}_{\beta\mathcal{RD}\mathcal{E}_{\Rightarrow\mathcal{C}}}$ is a set of evaluation contexts for $\to_{\beta\mathcal{RD}}$.

(ii) It is clear that the terms of (ii) are values. We check that if $\mathtt{t}$ is a value, then it is of one of these forms. Assume now that $\mathsf{E}[\ ] \in \mathcal{E}_{\Rightarrow\mathcal{C}}$ is a minimal context that interact with $\mathtt{t}$. Note that the top symbol of $\mathsf{E}[\ ]$ is either an application or a destructor $\mathtt{d}$. We consider these two cases:

The top symbol of $\mathsf{E}[\ ]$ is an application. In this case, $\mathsf{E}[\ ]$ is of the form $\mathsf{F}[\ ]\,\mathtt{u}$. By minimality, $\mathsf{F}[\ ] = [\ ]$ and $\mathtt{t}$ is an abstraction.

The top symbol of $\mathsf{E}[\ ]$ is a destructor. In this case, $\mathsf{E}[\ ]$ is of the form $\mathtt{d}(\mathsf{F}[\ ])$. By minimality, $\mathsf{F}[\ ] = [\ ]$ and $\mathtt{t}$ is a constructor.

(iii) The fact that $\mathcal{E}_{\Rightarrow\mathcal{C}}$ is a set elimination contexts is a direct consequence of the shape of values (ii).

(iv) From (ii) we know that terms of the form $(\lambda\mathtt{x}.\mathtt{t})\mathtt{u}$ and $\mathtt{f}(\vec{\mathtt{t}})$ with $\mathtt{f} \in \Sigma \setminus \mathcal{C}$ are neutral. Properties (3) and (7) then follows from Lem. 4.8. □

Example 4.12. Consider the system presented at example 2.1. Its values are the terms of the form

$$\lambda\mathtt{x}.\mathtt{t} \qquad \mathtt{S}(\mathtt{t}) \qquad \mathtt{0}\ .$$

Indeed, we have

$$(\lambda\mathtt{x}.\mathtt{t})\,\mathtt{u} \ \to_\beta\ \mathtt{t}[\mathtt{u}/\mathtt{x}] \qquad \mathtt{d}_{\mathtt{S},1}(\mathtt{S}(\mathtt{t})) \ \to_{\mathcal{D}}\ \mathtt{t} \qquad \mathtt{d}_{\mathtt{0}}(\mathtt{0}) \ \to_{\mathcal{D}}\ \mho\ .$$

5 Stability by Union

A reducibility family $\mathcal{R}\mathtt{ed}$ is *stable by union* if

$$\forall\mathcal{R}.\quad \mathcal{R} \subseteq \mathcal{R}\mathtt{ed} \qquad \Longrightarrow \qquad \bigcup\mathcal{R} \in \mathcal{R}\mathtt{ed}\ .$$

The main question on stability by union is the following: given a rewrite relation $\to_{\mathsf{R}}$, does there exists a reducibility family $\mathcal{R}\mathtt{ed}$ for $\to_{\mathsf{R}}$ which is stable by union *and* leads to an adequate type interpretation ?

For the pure λ-calculus, it is well-known that the answer is positive: Tait's saturated sets (presented in Sec. 3) are stable by union and lead to an adequate type interpretation. This has been exploited for instance in [2, 21].

The question becomes more difficult with rewriting. We have seen in Sec. 3 that rewrite systems do not satisfy in general the weak standardization lemma (Lem. 3.2), and that we need a reducibility family satisfying a clause like (7). But this is precisely what makes stability by union difficult. Assume given $\mathcal{R} \subseteq \mathcal{R}\mathtt{ed}$ such that for all $\mathsf{v} \in (\mathsf{E}[\mathtt{f}(\vec{\mathtt{t}})])_{\beta\mathcal{R}}$, we have $\mathsf{v} \in \bigcup\mathcal{R}$. Then, unless we find some $\mathsf{A} \in \mathcal{R}$ such that $(\mathsf{E}[\mathtt{f}(\vec{\mathtt{t}})])_{\beta\mathcal{R}} \subseteq \mathsf{A}$, there is no reason to have $\mathsf{E}[\mathtt{f}(\vec{\mathtt{t}})] \in \bigcup\mathcal{R}$.

Besides, using intersection and union types, we have shown in [19] that there are confluent typed rewrite systems for which no reducibility family that is stable by union leads to an adequate type interpretation.

However, we can in some cases obtain a reducibility family which is stable by union. In [18], we have given a necessary and sufficient condition for reducibility candidates to be stable by union; and in [19], we have given a necessary and sufficient condition for the closure by union of Biorthogonals [11, 16, 8] to be reducibility candidates. The second condition is strictly more general than the first one.

We now recall the condition established in [18] for the stability by union of reducibility candidates, and then show that it is met with *orthogonal* constructor rewriting.

5.1 Reducibility Candidates

The study of stability by union of reducibility candidates of [18] carries on with our generalization of neutral terms and elimination contexts. The key observation is a characterization of the membership of a term to a candidate using a *weak observational preorder*.

Definition 5.1. *Let* $t \lesssim_{\mathcal{N}} u$ *if and only if* $t, u \in \mathcal{SN}_R$ *and*

$$\forall v \in \mathcal{V}_{R\mathcal{E}}. \quad t \to_R^* v \quad \Longrightarrow \quad u \to_R^* v \ .$$

Every candidate $C \in \mathcal{CR}_{R\mathcal{E}}$ is a non-empty subset of $\mathcal{SN}_R$ downward-closed wrt. $\lesssim_{\mathcal{N}}$. Reducibility candidates are stable by union exactly when the converse is also true.

Theorem 5.2 ([18]). *The following are equivalent:*

(i) $\mathcal{CR}_{R\mathcal{E}}$ *is stable by union;*
(ii) $\mathcal{CR}_{R\mathcal{E}}$ *is the set of non-empty* $C \subseteq \mathcal{SN}_R$ *that are downward-closed wrt.* $\lesssim_{\mathcal{N}}$*;*
(iii) every strongly normalizable neutral term t *which is reducible has a reduct* u *such that* $t \lesssim_{\mathcal{N}} u$*. Such* u *is a* strong principal reduct[1] *of* t*.*

Example 5.3 ([18, 21]). For the pure λ-calculus, thanks to the weak standardization lemma (Lem. 3.2), $\mathcal{CR}_{\beta\mathcal{E}_{\Rightarrow}}$ is stable by union.

5.2 Application to Orthogonal Constructor Rewriting

Recall that a rewrite system $\mathcal{R}$ is *orthogonal* if it is left linear and has no critical pairs [22]. For instance, the system of Ex. 2.1 is orthogonal.

In this section, we show that if $\mathcal{R}$ is an orthogonal constructor rewrite system, then $\mathcal{CR}_{\beta\mathcal{RDE}_{\Rightarrow c}}$ is stable by union. According to Thm. 5.2, this amounts to show that every strongly normalizing reducible neutral term has a strong principal reduct. We prove it by using a general theorem on external redexes [13] (see also [12]). This result applies to the framework of orthogonal CCERSs, of which our higher-order rewrite systems $\beta \cup \mathcal{R} \cup \mathcal{D}$ are an instance.

[1] Called "principal reduct" in [18].

We use the notions of *descendants* and of *redex-arguments* of [13, 12]. The key notion is that of *external redex* [13, 12]: a redex u in a term t is *external* if for any derivation $P : t \to^*_{\beta\mathcal{RD}} v$, no descendant of u along P appears inside redex-arguments. Hence every descendant of an external redex is external. Note that every external redex is outermost, but the converse is false.

Using external redexes, we have the following weak standardization lemma. We apply it in Lem. 5.5 to show that every neutral term that has an external redex has a strong principal reduct.

Lemma 5.4 (Weak Standardization). *Assume that $\mathcal{R}$ is orthogonal. Let $t \to_{\beta\mathcal{RD}} u$ by contracting an external redex of t. If $t \to_{\beta\mathcal{RD}} v$ by contracting a different redex, then there is w such that $u \to^*_{\beta\mathcal{RD}} w$ and $v \to_{\beta\mathcal{RD}} w$ by contracting a descendant of $t \to_{\beta\mathcal{RD}} u$ (which is therefore external in v).*

Proof. We use the following standard notations: a position in a term t is a finite word on $\mathbb{N} \setminus \{0\}$, $\phi \cdot \psi$ denotes the concatenations of the words ϕ and ψ, and $t[u]_\phi$ is the term t in which the subterm at position ϕ is textually replaced by u.

Let ϕ and ψ be the respective positions of the redexes contracted in $t \to_{\beta\mathcal{RD}} u$ and $t \to_{\beta\mathcal{RD}} v$. We show that there exists w such that the redex contracted in $v \to_{\beta\mathcal{RD}} w$ is a descendant of ϕ. It is an external redex of v because ϕ is external in t.

Note that since ϕ is external in t, for every ϕ_1, ϕ_2 such that $\phi = \phi_1 \cdot \phi_2$, ϕ_2 is external in $t|_{\phi_1}$: if a descendant ϕ_2' of ϕ_2 appears in a redex argument of a reduct u of $t|_{\phi_1}$, then $\phi_1 \cdot \phi_2'$, which is a descendant of ϕ, appears in a redex argument of $t[u]_{\phi_1}$, which is a reduct of t.

We now reason by induction on t.

$t \in \mathcal{X}$**.** Not possible.

$t = \lambda x.t_1$**.** In this case, $u = \lambda x.u_1$ and $v = \lambda x.v_1$. Moreover we have $\phi = 1 \cdot \phi_1$, $\psi = 1 \cdot \psi_1$, $t_1 \to_{\beta\mathcal{RD}} u_1$ by contracting ϕ_1 and $t_1 \to_{\beta\mathcal{RD}} v_1$ by contracting ψ_1. Since ϕ_1 is external in t_1, by induction hypothesis there is w_1 such that $u_1 \to^*_{\beta\mathcal{RD}} w_1$ and $v_1 \to_{\beta\mathcal{RD}} w_1$ by contracting a descendant of ϕ_1. It follows that $u \to^*_{\beta\mathcal{RD}} \lambda x.w_1$ and $v \to_{\beta\mathcal{RD}} \lambda x.w_1$ by contracting a descendant of ϕ.

$t = t_1\, t_2$ **with t_1 not an abstraction.** There are $i, j \in \{1, 2\}$ such that $\phi = i \cdot \phi_1$ and $\psi = j \cdot \psi_1$. Moreover, $u = u_1\, u_2$ and $v = v_1\, v_2$ with $t_i \to_{\beta\mathcal{RD}} u_i$, $t_j \to_{\beta\mathcal{RD}} v_j$, $u_j = t_j$ and $v_i = t_i$. There are two cases.

If $i = j$, then we reason as in the case of $t = \lambda x.t_1$.

Otherwise $i \neq j$. Let $w = w_1\, w_2$ with $w_i = u_i$ and $w_j = v_j$. We have $u \to_{\beta\mathcal{RD}} w$ and $v \to_{\beta\mathcal{RD}} w$ by contracting a descendant of ϕ.

$t = (\lambda x.t_1)t_2$**.** Since ϕ is external in t, ϕ is the root β-redex of t. Therefore $u = t_1[t_2/x]$ and $v = (\lambda x.t_1')t_2'$ with $(t_1, t_2) \to_{\beta\mathcal{RD}} (t_1', t_2')$. We are done by taking $w =_{\text{def}} t_1'[t_2'/x]$ since $(\lambda x.t_1')t_2'$ is a descendant of $(\lambda x.t_1)t_2$ and $t_1[t_2/x] \to^*_{\beta\mathcal{RD}} t_1'[t_2'/x]$.

$t = f(\vec{t})$ **and t is not a $\mathcal{RD}$-redex.** We reason as in the case of $t_1\, t_2$ with t_1 not an abstraction.

$t = f(\vec{t})$ **and** t **is a** $\mathcal{RD}$**-redex.** Since ϕ is external in t, it is the roof $\mathcal{RD}$-redex of t. Hence there is a rule $l \mapsto_{\mathcal{RD}} r$ and a substitution σ such that $t = l\sigma$ and $u = r\sigma$. Because l is linear, there is a substitution σ' such that $\sigma \to_{\beta\mathcal{RD}} \sigma'$ and $v = l\sigma'$. We are done by taking $w =_{\text{def}} r\sigma'$ since $l\sigma'$ is a descendant of $l\sigma$ and $r\sigma \to^*_{\beta\mathcal{RD}} r\sigma'$. □

Lemma 5.5. *Assume that* $\mathcal{R}$ *is orthogonal and let* $t \in \mathcal{N}_{\beta\mathcal{RDE}_{\Rightarrow c}}$. *If* $t \to^*_{\beta\mathcal{RD}} v$ *with* $v \in \mathcal{V}_{\beta\mathcal{RDE}_{\Rightarrow c}}$, *then for all* u *such that* $t \to_{\beta\mathcal{RD}} u$ *by contracting an external redex of* t, *we have* $u \to^*_{\beta\mathcal{RD}} v$.

Proof. Let u such that $t \to_{\beta\mathcal{RD}} u$ by contracting an external redex of t and let $v \in \mathcal{V}_{\beta\mathcal{RDE}_{\Rightarrow c}}$ such that $t \to^*_{\beta\mathcal{RD}} v$. Since t is neutral, we have $t \to^+_{\beta\mathcal{RD}} v$. We reason by induction on the length of the derivation.

Base Case. Since t is neutral, it follows from Lem. 4.11.(ii) that $t \to_{\beta\mathcal{RD}} v$ by contracting a top redex of t. Since every external redex is outermost, it follows that every external redex of t is a root redex of t. By orthogonality, this is *the* redex contracted in $t \to_{\beta\mathcal{RD}} v$, and we have $u = v$.

Induction Case. Assume that $t \to_{\beta\mathcal{RD}} s \to^*_{\beta\mathcal{RD}} v$. If s is a value, then reasoning as in the base case we have $s = u$, hence $u \to^*_{\beta\mathcal{RD}} v$. Otherwise s is neutral. If s is obtained from t by contracting the same redex as in $t \to_{\beta\mathcal{RD}} u$, then $s = u$ and $u \to^*_{\beta\mathcal{RD}} v$. Otherwise, by weak standardization (Lem. 5.4), there is w such that $u \to^*_{\beta\mathcal{RD}} w$ and $s \to_{\beta\mathcal{RD}} w$ by contracting an external redex of s. We then have $w \to^*_{\beta\mathcal{RD}} v$ by induction hypothesis, hence $u \to^*_{\beta\mathcal{RD}} v$. □

If t is neutral and $t \to_{\beta\mathcal{RD}} u$ by contracting an external redex of t, then we have $t \lesssim_{\mathcal{N}} u$. Hence u is a strong principal reduct of t. Then, by Thm. 5.2 $\mathcal{CR}_{\beta\mathcal{RDE}_{\Rightarrow c}}$ is stable by union if all neutral terms have external redexes. It remains to show this last property, which follows from the next theorem, proved in [13]. We can apply it because the CCERS $\beta \cup \mathcal{R} \cup \mathcal{D}$ is orthogonal as soon as the rewrite system $\mathcal{R}$ is orthogonal.

Theorem 5.6 ([13, 12]). *If* $\mathcal{R}$ *is orthogonal then every reducible term has an external redex.*

Theorem 5.7. *If* $\mathcal{R}$ *is orthogonal then* $\mathcal{CR}_{\beta\mathcal{RDE}_{\Rightarrow c}}$ *is stable by union.*

References

[1] A. Abel. Termination Checking with Types. *RAIRO – Theoretical Informatics and Applications*, 38(4):277–319, 2004. Special Issue (FICS'03). 4

[2] A. Abel. Semi-Continuous Sized Types and Termination. In *Proceedings of CSL'06*, volume 4207 of *LNCS*, pages 72–88. Springer, 2006. 1, 4, 8

[3] T. Altenkirch. *Constructions, Inductive Types and Strong Normalization*. PhD thesis, University of Edinburgh, 1993. 4

[4] H.P. Barendregt. Lambda Calculi with Types. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2. Oxford University Press, 1992. 2

[5] G. Barthe, B. Grégoire, and F. Pastawski. Type-Based Termination of Recursives Definitions in the Calculus of Inductive Constructions. In *Proceedings of LPAR'06*, pages 257–271, 2006. 1, 4

[6] F. Blanqui, J.-P. Jouannaud, and M. Okada. Inductive-Data-Types Systems. *Theoretical Computer Science*, 271:41–68, 2002. 1, 4, 7

[7] 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. 1, 4

[8] V. Danos and J.-L. Krivine. Disjunctive Tautologies as Synchronisation Schemes. In *Proceedings of CSL'00*, volume 1862 of *LNCS*, pages 292–301, 2000. 9

[9] J.H. Gallier. On Girard's "Candidats de Reducibilité". In P. Odifredi, editor, *Logic and Computer Science*. Academic Press, 1989. 2

[10] J.-Y. Girard. *Interprétation Fonctionnelle et Élimination des Coupures de l'Arithmétique d'Ordre Supérieur*. PhD thesis, Université Paris 7, 1972. 1, 5

[11] J.-Y. Girard. Linear Logic. *Theoretical Computer Science*, 50:1–102, 1987. 1, 9

[12] J. Glauert, D. Kesner, and Z. Khasidashvili. Expression Reduction Systems and Extensions: An Overview. In *Processes, Terms and Cycles: Steps to the Road of Infinity, Essays Dedicated to Jan Willem Klop, on the Occasion of His 60th Birthday*, volume 3838 of *LNCS*, pages 496–553. Springer, 2005. 2, 9, 10, 11

[13] Z. Khasidashvili, M. Ogawa, and V. van Oostrom. Perpetuality ans Uniform Normalization in Orthogonal Rewrite Systems. *Information and Computation*, 164(1):118–152, 2001. 1, 9, 10, 11

[14] J.-L. Krivine. *Lambda-Calcul, Types et Modèles*. Masson, 1990. 2

[15] N. P. Mendler. Recursive Types and Type Constraints in Second Order Lambda-Calculus. In *Proceedings of LiCS'87*, pages 30–36. IEEE Computer Society, 1987. 2

[16] M. Parigot. Proofs of Strong Normalization for Second Order Classical Natural Deduction. *Journal of Symbolic Logic*, 62(4):1461–1479, 1997. 1, 9

[17] C. Riba. *Definitions par Réécriture dans le λ-Calcul: Confluence, Réductibilité et Typage*. PhD thesis, INPL, 2007. 2

[18] C. Riba. On the Stability by Union of Reducibility Candidates. In *Proceedings of FoSSaCS'07*, volume 4423 of *LNCS*, 2007. 1, 9

[19] C. Riba. Strong Normalization as Safe Interaction. In *Proceedings of LiCS'07*, pages 13–22. IEEE Computer Society, 2007. 8, 9

[20] 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. 1, 4

[21] 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. 1, 8, 9

[22] Terese. *Term Rewriting Systems*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003. M. Bezem, J.W. Klop and R.C. de Vrijer, eds. 2, 9