Récriture d’Ordre Supérieur

Jean-Pierre Jouannaud
École Polytechnique
91400 Palaiseau, France

email: jouannaud@lix.polytechnique.fr
http: //w^3.lix.polytechnique.fr/Labo/Jean-Pierre.Jouannaud

Project LogiCal, Pôle Commun de Recherche en Informatique du Plateau de Saclay, CNRS, École Polytechnique, INRIA, Université Paris-Sud.

January 3, 2005

Outline

Notre but est de décrire

- les algèbres d'ordre supérieur polymorphes,
- la récriture avec filtrage simple,
- la récriture avec filtrage modulo $\beta\eta$,
- la réduction de la confluence à certaines formes de paires critiques,
- les preuves de terminaison basées sur le HORPO.

Given a set $\mathcal{S}$ of *sort symbols* of a fixed arity, denoted by $s : *^n \to *$, and a set $\mathcal{S}^\forall$ of *type variables*, the set $\mathcal{T}_{\mathcal{S}^\forall}$ of *polymorphic types* is generated from these sets by the constructor $\to$ for *functional types*:

$$\mathcal{T}_{\mathcal{S}^\forall} := \alpha \mid s(\mathcal{T}^n_{\mathcal{S}^\forall}) \mid (\mathcal{T}_{\mathcal{S}^\forall} \to \mathcal{T}_{\mathcal{S}^\forall})$$
$$\text{for } \alpha \in \mathcal{S}^\forall \text{ and } s : *^n \to * \ \in \mathcal{S}$$

$\mathcal{V}ar(\sigma)$ denotes the set of (type) variables of the type $\sigma \in \mathcal{T}_{\mathcal{S}^\forall}$. Types are *functional* when headed by the $\to$ symbol, and *data types* when they are headed by a sort symbol. $\to$ associates to the right.

Type substitutions

A *type substitution* is a mapping from $\mathcal{S}^\forall$ to $\mathcal{T}_{\mathcal{S}^\forall}$ extended to an endomorphism of $\mathcal{T}_{\mathcal{S}^\forall}$. We write $\sigma\xi$ for the application of the type substitution ξ to the type σ. We denote by $\mathcal{D}om(\sigma) = \{\alpha \in \mathcal{S}^\forall \mid \alpha\sigma \neq \alpha\}$ the domain of $\sigma \in \mathcal{T}_{\mathcal{S}^\forall}$, by $\sigma|_\mathcal{V}$ its restriction to the domain $\mathcal{D}om(\sigma) \cap \mathcal{V}$, by $\mathcal{R}an(\sigma) = \bigcup_{\alpha \in \mathcal{D}om(\sigma)} \mathcal{V}ar(\alpha\sigma)$ its *range*. By a renaming of the type σ apart from $V \subset \mathcal{X}$, we mean a type $\sigma\xi$ where ξ is a type renaming such that $\mathcal{D}om(\xi) = \mathcal{R}an(\sigma)$ and $\mathcal{R}an(\xi) \cap \mathcal{V} = \emptyset$.

We shall use α, β for type variables, $\sigma, \tau, \rho, \theta$ for arbitrary types, and ξ, ζ to denote type substitutions.

Signatures

Function symbols are meant to be algebraic operators equiped with a fixed number n of arguments (called the *arity*) of respective types $\sigma_1 \in \mathcal{T}_{\mathcal{S}^\forall}, \ldots, \sigma_n \in \mathcal{T}_{\mathcal{S}^\forall}$, and an *output type* $\sigma \in \mathcal{T}_{\mathcal{S}^\forall}$ such that $\mathcal{V}ar(\sigma) \subseteq \bigcup_i \mathcal{V}ar(\sigma_i)$:

$$\mathcal{F} = \biguplus_{\sigma_1,\ldots,\sigma_n,\sigma} \mathcal{F}_{\sigma_1 \times \ldots \times \sigma_n \to \sigma}$$

Membership of f to a set $\mathcal{F}_{\sigma_1 \times \ldots \times \sigma_n \to \sigma}$ is written $f : \sigma_1 \times \ldots \times \sigma_n \to \sigma$. A type declaration is *first-order* if it uses only sorts, and higher-order otherwise. It is *polymorphic* if it uses some polymorphic type, otherwise, it is *monomorphic*. Type instantiation does not change arities.

Terms

$$\mathcal{T} := \mathcal{X} \mid (\lambda\mathcal{X} : \mathcal{T}_{\mathcal{S}^\forall}.\mathcal{T}) \mid @(\mathcal{T},\mathcal{T}) \mid \mathcal{F}(\mathcal{T},\ldots,\mathcal{T}).$$

We may omit σ in $\lambda x : \sigma.u$ as well as $@$, writing $u(v)$ for $@(u,v)$. The term $u(\overline{v})$ is called a (partial) *left-flattening* of $s = u(v_1)\ldots(v_n)$, u being possibly an application itself. $\mathcal{V}ar(t)$ is the set of free variables of t. $\overline{s}$ shall be ambiguously used to denote a list, or a multiset, or a set of terms $s_1,\ldots,s_n$.
Terms are identified with finite labeled trees by considering $\lambda x : \sigma._$ as a unary function symbol taking a term u as argument to construct the term $\lambda x : \sigma.u$.

Definition

An *environment* Γ is a finite set of pairs written as $\{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$, where x_i is a variable, σ_i is a type, and $x_i \neq x_j$ for $i \neq j$.
$\mathcal{V}ar(\Gamma) = \{x_1, \ldots, x_n\}$ is the set of variables of Γ.
Given two environments Γ and Γ', their *composition* is the environment $\Gamma \cdot \Gamma' = \Gamma' \cup \{x : \sigma \in \Gamma \mid x \notin \mathcal{V}ar(\Gamma')\}$. Two environments Γ and Γ' are *compatible* if $\Gamma \cdot \Gamma' = \Gamma \cup \Gamma'$.

Our typing judgements are written as $\Gamma \vdash_{\mathcal{F}} s : \sigma$.

Règles de typage

Variables:

$$\frac{x : \sigma \in \Gamma}{\Gamma \vdash_{\mathcal{F}} x : \sigma}$$

Abstraction:

$$\frac{\Gamma \cdot \{x : \sigma\} \vdash_{\mathcal{F}} t : \tau}{\Gamma \vdash_{\mathcal{F}} (\lambda x : \sigma . t) : \sigma \rightarrow \tau}$$

Functions:

$$\frac{\begin{array}{c} f : \sigma_1 \times \ldots \times \sigma_n \rightarrow \sigma \in \mathcal{F} \\ \xi \text{ some type substitution of domain } \subseteq \bigcup_i \mathcal{V}ar(\sigma_i) \\ \Gamma \vdash_{\mathcal{F}} t_1 : \sigma_1\xi \quad \ldots \quad \Gamma \vdash_{\mathcal{F}} t_n : \sigma_n\xi \end{array}}{\Gamma \vdash_{\mathcal{F}} f(t_1, \ldots, t_n) : \sigma\xi}$$

Application:

$$\frac{\Gamma \vdash_{\mathcal{F}} s : \sigma \rightarrow \tau \quad \Gamma \vdash_{\mathcal{F}} t : \sigma}{\Gamma \vdash_{\mathcal{F}} @(s, t) : \tau}$$

Properties

Lemma

Given an environment Γ and a typable term s, there exists a unique type σ such that $\Gamma \vdash_{\mathcal{F}} s : \sigma$.

Lemma

$\Gamma \vdash_{\mathcal{F}} s : \sigma$ *implies* $\Gamma\xi \vdash_{\mathcal{F}} s\xi : \sigma\xi$ *for any* ξ.

Lemma

Given a signature $\mathcal{F}$, environment Γ, term s and type σ such that $\Gamma \vdash_{\mathcal{F}} s : \sigma$, then $\Gamma \cdot \Gamma' \vdash_{\mathcal{F}} s : \sigma$ for all Γ' compatible with Γ.

Properties

Lemma

Given a signature $\mathcal{F}$, environment Γ, term s and type σ such that $\Gamma \vdash_{\mathcal{F}} s : \sigma$, then for all $p \in \mathcal{D}om(s)$, there exists a canonical environment $\Gamma_{s|_p}$ and a type τ such that $\Gamma_{s|_p} \vdash_{\mathcal{F}} s|_p : \tau$ is a subproof of the proof of $\Gamma \vdash_{\mathcal{F}} s : \sigma$. Moreover, $\Gamma_{s|_{(p\cdot q)}} = (\Gamma_{s|_p})_{(s|_p)|_q}$.

Lemma

Given a signature $\mathcal{F}$, an environment Γ, two terms s and v, two types σ and τ, and a position $p \in \mathcal{P}os(s)$ such that $\Gamma \vdash_{\mathcal{F}} s : \sigma$, $\Gamma_{s|_p} \vdash_{\mathcal{F}} s|_p : \tau$ and $\Gamma_{s|_p} \vdash_{\mathcal{F}} v : \tau$, then $\Gamma \vdash_{\mathcal{F}} s[v]_p : \sigma$.

Substitutions

Definition

A *substitution*
$\gamma = \{(x_1 : \sigma_1) \mapsto (\Gamma_1, t_1), \ldots, (x_n : \sigma_n) \mapsto (\Gamma_n, t_n)\}$,
is a finite set of quadruples made of a variable symbol, a type, an environment and a term, such that
(i) $\forall i \in [1..n]$, $t_i \neq x_i$ and $\Gamma_i \vdash_{\mathcal{F}} t_i : \sigma_i$,
(ii) $\forall i \neq j \in [1..n]$, $x_i \neq x_j$, and
(iii) $\forall i \neq j \in [1..n]$, Γ_i and Γ_j are compatible environments.
We may omit the type σ_i and environment Γ_i in $(x_i : \sigma_i) \mapsto (\Gamma_i, t_i)$.

Substitutions

The set of (input) variables of the substitution γ is $\mathcal{V}ar(\gamma) = \{x_1, \ldots, x_n\}$, its *domain* is the environment $\mathcal{D}om(\gamma) = \{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$ while its *range* is the environment $\mathcal{R}an(\gamma) = \bigcup_{i \in [1..n]} \Gamma_i$.
Note that $\mathcal{R}an(\gamma)$ is indeed an environment by assumption (iii).

Lemma

Given
$\gamma = \{(x_1 : \sigma_1) \mapsto (\Gamma_1, t_1), \ldots, (x_n : \sigma_n) \mapsto (\Gamma_n, t_n)\}$,
then $\mathcal{R}an(\gamma) \vdash_{\mathcal{F}} t_i : \sigma_i$.

Compatibility

Definition

A substitution γ is *compatible* with an environment Γ if
(i) $\mathcal{D}om(\gamma)$ is compatible with Γ,
(ii) $\mathcal{R}an(\gamma)$ is compatible with $\Gamma \setminus \mathcal{D}om(\gamma)$.
We will also say that γ is compatible with the judgement $\Gamma \vdash_{\mathcal{F}} s : \sigma$.

Substitution instance

Definition

A substitution γ compatible with a judgement $\Gamma \vdash_{\mathcal{F}} s : \sigma$ operates as an endomorphism on s and yields the term $s\gamma$ defined as:

If	$s = x \in \mathcal{X}$ and $x \notin \mathcal{V}ar(\gamma)$
then	$s\gamma = x$
If	$s = x \in \mathcal{X}$ and $(x : \sigma) \mapsto (\Gamma, t) \in \Gamma$
then	$s\gamma = t$
If	$s = @(u, v)$
then	$s\gamma = @(u\gamma, v\gamma)$
If	$s = f(u_1, \ldots, u_n)$
then	$s\gamma = f(u_1\gamma, \ldots, u_n\gamma)$
If	$s = \lambda x : \tau.u$

Type invariance

Lemma

Given a signature $\mathcal{F}$ and a substitution γ compatible with the judgement $\Gamma \vdash_{\mathcal{F}} s : \sigma$, then $\Gamma \cdot \mathcal{R}an(\gamma) \vdash_{\mathcal{F}} s\gamma : \sigma$.

Exemple

Let $\mathcal{S} = \{o_1,\ o_2,\ o_3,\ o_4\}$, $\mathcal{S}^\forall = \{\alpha : *, \beta : *\}$, and $\mathcal{F} = \{f : \alpha \times \beta \to \alpha, g : \alpha \times \beta \to \beta\}$.
Let $\Gamma = \{x_1 : o_1,\ x_2 : o_2,\ x_3 : o_3,\ x_4 : o_4\}$, and $s = g(f(x_1, x_2), f(x_3, x_4))$. Then $\Gamma \vdash_{\mathcal{F}} s : o_3$. Let $\gamma = \{x_1 : o_1 \mapsto (\{x_1 : o_2,\ x_6 : o_1\}, g(x_1, x_6)), x_3 : o_3 \mapsto (\{x_2 : o_2,\ x_5 : o_3\}, g(x_2, x_5)), x_6 : o_2 \mapsto (\{x_1 : o_2,\ x_5 : o_3\}, f(x_1, x_5))\}$.
$\mathcal{D}om(\gamma) = \{x_1 : o_1,\ x_3 : o_3,\ x_6 : o_2\}$, and $\mathcal{R}an(\gamma) = \{x_1 : o_2, x_2 : o_2, x_5 : o_3,\ x_6 : o_1\}$.
$s\gamma = g(f(g(x_1, x_6), x_2), f(g(x_2, x_5), x_4))$.
$\Gamma \cdot \mathcal{R}an(\gamma) = \{x_1 : o_2,\ x_2 : o_2,\ x_3 : o_3,\ x_4 : o_4,\ x_5 : o_3,\ x_6 : o_1\}$.
$\Gamma \cdot \mathcal{R}an(\gamma) \vdash_{\mathcal{F}} s\gamma : o_3$.

Definition

Given a regular signature $\mathcal{F}$, a *rewrite rule* is a quadruple written $\Gamma \vdash l \rightarrow r : \sigma$, where l and r are higher-order terms such that
(i) $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$,
(ii) $\Gamma \vdash_{\mathcal{F}} l : \sigma$ and $\Gamma \vdash_{\mathcal{F}} r : \sigma$.
The rewrite rule is said to be *polymorphic* if σ is a polymorphic type. A *plain term rewriting system*, or simply *term rewriting system* is a set of rewrite rules.

Definition

Given a plain higher-order rewriting system R and an environment Γ, a term s such that $\Gamma \vdash_{\mathcal{F}} s : \sigma$ rewrites to a term t at position p with the rule $\Gamma_i \vdash l_i \rightarrow r_i : \sigma_i$, the type substitution ξ and the term substitution γ, written $\Gamma \vdash s \xrightarrow[\Gamma_i \vdash l_i \rightarrow r_i]{p} t$, or $s \rightarrow_R t$ assuming the environment Γ, if:

(i) $\mathcal{D}om(\gamma) \subseteq \Gamma_i\xi$,

(ii) $\Gamma_i\xi \cdot \mathcal{R}an(\gamma) \subseteq \Gamma_{s|_p}$,

(iii) $s|_p = l_i\xi\gamma$,

(iv) $t = s[r_i\xi\gamma]_p$.

Conservation du typage par récriture

Lemma

Assume that $\Gamma \vdash_{\mathcal{F}} s : \sigma$ *and* $\Gamma \vdash s \to_R t$*. Then* $\Gamma \vdash_{\mathcal{F}} t : \sigma$*.*

Proof.

By Lemma 3, $\Gamma_i\xi \vdash_{\mathcal{F}} l_i\xi : \sigma_i\xi$. By conditions (i,ii), γ is compatible with the environment $\Gamma_i\xi$.
Hence, $\Gamma_i\xi \cdot \mathcal{R}an(\gamma) \vdash_{\mathcal{F}} l_i\xi\gamma : \sigma_i\xi$ by lemma 11.
By condition (ii) and lemma 4, $\Gamma_{s|_p} \vdash_{\mathcal{F}} l_i\xi\gamma : \sigma_i\xi$, and therefore, by condition (iii), $\Gamma_{s|_p} \vdash_{\mathcal{F}} s|_p : \sigma_i\xi$ (this tells us how to compute ξ).
Similarly,$\Gamma_{s|_p} \vdash_{\mathcal{F}} r_i\xi\gamma : \sigma_i\xi$. By lemma 6, $\Gamma \vdash_{\mathcal{F}} s[r_i\xi\gamma]_p : \sigma_i\xi$. Condition (iv) concludes. □

The following three equations originate from the λ-calculus, and are called α-, β- and η-conversions:

$$
\begin{array}{rcl}
\{v : \tau\} \vdash & \lambda x : \sigma . v =_\alpha \lambda y : \sigma . v\{x \mapsto y\} \\
& \text{if } y \notin \mathcal{BV}ar(v) \cup (\mathcal{V}ar(v) \setminus \{x\}) \\
\{u : \sigma,\ v : \tau\} \vdash & @(\lambda x : \sigma . v, u) =_\beta v\{x \mapsto u\} \\
\{u : \sigma \to \tau\} \vdash & \lambda x : \sigma . @(u, x) =_\eta u \\
& \text{if } x \notin \mathcal{V}ar(u)
\end{array}
$$

The above equations are equation schemas : all occurrences of u and v stand for arbitrary terms to which substitutions $\{x \rightarrow y\}$ and $\{x \rightarrow u\}$ apply. α-convertible terms are considered identical. $\overset{*}{\underset{\beta}{\longleftrightarrow}}$ is the congruence generated by the β-equality, and $\longrightarrow_\beta$ the β-reduction rule:

$$\{u : \alpha,\ v : \beta\} \vdash_{\mathcal{F}} @(\lambda x : \alpha.v, u) \ \longrightarrow_\beta\ v\{x \mapsto u\}$$

Gödel's system T

Let $\mathcal{S} = \{\mathbb{N}\}$, $\mathcal{S}^{\forall} = \{\alpha\}$,
$\mathcal{F} = \{0 :\to \mathbb{N},\ s : \mathbb{N} \to \mathbb{N},\ + : \mathbb{N} \times \mathbb{N} \to \mathbb{N},$
$rec : \mathbb{N} \times \alpha \times (\mathbb{N} \to \alpha \to \alpha) \to \alpha\}$.
Gödel's recursor for natural numbers is defined by the following rewrite rules:

$$\{U : \alpha,\ X : \mathbb{N} \to \alpha \to \alpha\} \vdash rec(0, U, X) \to U$$

$$\{x : \mathbb{N},\ U : \alpha,\ X : \mathbb{N} \to \alpha \to \alpha\} \vdash rec(s(x), U, X) \to @(X, x, rec(x, U, X))$$

$$\{\} \vdash s = rec(S(0), 0, rec(0, \lambda x : \mathbb{N}\, y : \mathbb{N}. +(x, y), \lambda x : \mathbb{N}\, y : \mathbb{N} \to \mathbb{N} \to \mathbb{N}\, z : \mathbb{N}. y(+(x, z)))))$$

$$s \to^{3}_{\{U:\alpha,\ X:\mathbb{N}\to\alpha\to\alpha\} \vdash rec(0,U,X)\to U} rec(S(0), 0, \lambda x : \mathbb{N}\, y : \mathbb{N}. +(x, y))$$

$$\to^{\epsilon}_{\{x:\mathbb{N},\ U:\alpha,\ X:\mathbb{N}\to\alpha\to\alpha\} \vdash rec(S(x),U,X)\to @(X,x,rec(x,U,X))} @(\lambda x, y : \mathbb{N}. +(x, y), 0, rec(0, 0, \lambda x, y : \mathbb{N}. +(x, y)))$$

$$\to^{\epsilon}_{\beta} @(\lambda y : \mathbb{N}. +(0, y), rec(0, 0, \lambda x, y : \mathbb{N}. +(x, y)))$$

$$\to^{\epsilon}_{\beta} +(0, rec(0, 0, \lambda x, y : \mathbb{N}. +(x, y)))$$

$$\to^{2}_{\{U:\alpha,\ X:\mathbb{N}\to\alpha\to\alpha\} \vdash rec(0,U,X)\to U} +(0, 0)$$

$$\to^{\epsilon}_{\{x:\mathbb{N}\} \vdash +(x,0)\to x} 0$$

Vocabulaire

A term s such that $s \xrightarrow[R]{p} t$ is called *reducible*. $s|_p$ is a *redex* and t the *reduct*. Irreducible terms are said in R-*normal form*. A substitution γ is in R-normal form if $x\gamma$ is in R-normal form for all x. We denote by $\xrightarrow{*}$ the reflexive, transitive closure of the rewrite relation $\longrightarrow$, and by $\xleftrightarrow{*}$ its reflexive, symmetric, transitive closure.

A term is *strongly normalizable* if there are no infinite rewriting sequences issuing from it. The relation $\longrightarrow$ is *strongly normalizing* if all terms are strongly normalizable.

It is *confluent* if $s \longrightarrow^* u$ and $s \longrightarrow^* v$ implies that $u \longrightarrow^* t$ and $v \longrightarrow^* t$ for some t.

Higher-order plain orderings

‘

Definition: a *higher-order reduction ordering* $\succ$ is a

well-founded ordering of the set of judgements:

(i) *monotonicity*: $(\Gamma \vdash_{\mathcal{F}} s : \sigma) \succ (\Gamma \vdash_{\mathcal{F}} t : \sigma)$ implies $(\Gamma \cdot \Gamma' \vdash_{\mathcal{F}} u[s] : \tau) \succ (\Gamma \cdot \Gamma' \vdash_{\mathcal{F}} u[t] : \tau)$ $\forall \Gamma' \vdash_{\mathcal{F}} u[x : \sigma] : \tau$ with Γ, Γ' compatible

(ii) *stability*: $(\Gamma \vdash_{\mathcal{F}} s : \sigma) \succ (\Gamma \vdash_{\mathcal{F}} t : \sigma)$ implies $(\Gamma \cdot \mathcal{R}an(\gamma) \vdash_{\mathcal{F}} s\gamma : \sigma) \succ (\Gamma \cdot \mathcal{R}an(\gamma) \vdash_{\mathcal{F}} t\gamma : \sigma)$ $\forall \gamma$ compatible with Γ

(iii) *compatibility*: $(\Gamma \vdash_{\mathcal{F}} s : \sigma) \succ (\Gamma \vdash_{\mathcal{F}} t : \sigma)$ implies $(\Gamma' \vdash_{\mathcal{F}} s : \sigma) \succ (\Gamma' \vdash_{\mathcal{F}} t : \sigma)$ $\forall \Gamma'$ s.t. Γ, Γ' compatible, $\Gamma' \vdash_{\mathcal{F}} s, t : \sigma$

(iv) *functionality*: $(\Gamma \vdash_{\mathcal{F}} s : \sigma \longrightarrow_{\beta} t : \sigma)$ implies $(\Gamma \vdash_{\mathcal{F}} s : \sigma) \succ (\Gamma \vdash_{\mathcal{F}} t : \sigma)$.

(v) *polymorphicity*: $(\Gamma \vdash_{\mathcal{F}} s : \sigma) \succ (\Gamma \vdash_{\mathcal{F}} t : \sigma)$ implies $(\Gamma\xi \vdash_{\mathcal{F}} s\xi : \sigma\xi) \succ (\Gamma\xi \vdash_{\mathcal{F}} t\xi : \sigma\xi)$ $\forall \xi$.

Strong normalization

Theorem

Let $\succeq$ be a higher-order reduction ordering and $R = \{\Gamma_i \vdash_{\mathcal{F}} l_i \rightarrow r_i\}_{i \in I}$ be a higher-order rewrite system such that $\Gamma_i \vdash_{\mathcal{F}} l_i \succ r_i$ for every $i \in I$. Then the relation $\longrightarrow_R \cup \longrightarrow_\beta$ is strongly normalizing.

Proof.

Let $\Gamma \vdash_{\mathcal{F}} s : \sigma$ and $\Gamma \vdash_{\mathcal{F}} s \xrightarrow[\Gamma_i\xi \,\vdash\, l_i\xi \to r_i\xi : \sigma_i\xi]{p} t$. By definition, $\Gamma_{s|_p} \vdash_{\mathcal{F}} s|_p : \sigma_i\xi$, $\mathcal{D}om(\gamma) \subseteq \Gamma_i\xi$, $\Gamma_i\xi \cdot \mathcal{R}an(\gamma) \subseteq \Gamma_{s|_p}$, $s|_p = l_i\xi\gamma$, and $t = s[r_i\xi\gamma]_p$.
By assumption, $\Gamma_i \vdash_{\mathcal{F}} l_i \succ r_i : \sigma_i$.
By polymorphism, $\Gamma_i\xi \vdash_{\mathcal{F}} l_i\xi \succ r_i\xi : \sigma_i\xi$.
By stability, $\Gamma_i\xi \cdot \mathcal{R}an(\gamma) \vdash_{\mathcal{F}} l_i\xi\gamma \succ r_i\xi\gamma : \sigma_i\xi$.
By compatibility, $\Gamma_{s|_p} \vdash_{\mathcal{F}} l_i\xi\gamma \succ r_i\xi\gamma : \sigma_i\xi$.
By monotonicity of $\succ$ for terms of equal type,
$\Gamma_{s|_p} \cdot \Gamma \vdash_{\mathcal{F}} s[l_i\xi\gamma] = s \succ s[r_i\xi\gamma] = t : \sigma$.
By compatibility again, $\Gamma \vdash_{\mathcal{F}} s \succ t$.
Finally, the case of a β-reduction is similar. □

La définition de la récriture normale fait intervenir des formes normales vis-à-vis de la β-réduction et de la η-expansion.

η-reduction and expansion

- η-reduction: si $x \notin \mathcal{V}ar(u)$ alors $\{u : \alpha \to \beta\} \vdash_{\mathcal{F}} \lambda x : \alpha.@(u, x) \to u$
- The use of η-expansion is restricted by spelling out in which context it applies:

$$\{u : \sigma_1 \to \ldots \to \sigma_n \to \sigma\} \vdash_{\mathcal{F}} s[u]_p \longrightarrow^p_\eta s[\lambda x_1 : \sigma_1, \ldots, x_n : \sigma_n.@(u, x_1, \ldots, x_n)]_p$$

$$\textbf{if} \begin{cases} \sigma \text{ is a canonical output type} \\ x_1, \ldots, x_n \notin \mathcal{V}ar(u) \\ u \text{ is not an abstraction} \\ s|_q \text{ is not an application in case } p = q \cdot 1 \end{cases}$$

Note that the first argument of an application is not recursively expanded on top.

η-reduction and expansion

- η-reduction: si $x \notin \mathcal{V}ar(u)$ alors $\{u : \alpha \to \beta\} \vdash_{\mathcal{F}} \lambda x : \alpha.@(u, x) \to u$
- The use of η-expansion is restricted by spelling out in which context it applies:

$$\{u : \sigma_1 \to \ldots \to \sigma_n \to \sigma\} \vdash_{\mathcal{F}} s[u]_p \longrightarrow^p_\eta s[\lambda x_1 : \sigma_1, \ldots, x_n : \sigma_n.@(u, x_1, \ldots, x_n)]_p$$

if $\begin{cases} \sigma \text{ is a canonical output type} \\ x_1, \ldots, x_n \notin \mathcal{V}ar(u) \\ u \text{ is not an abstraction} \\ s|_q \text{ is not an application in case } p = q \cdot 1 \end{cases}$

Note that the first argument of an application is not recursively expanded on top.

The simply typed λ-calculus is confluent modulo α-conversion, and terminating with respect to β-reductions and either the above notion of η-expansions, or the more usual notion of η-reduction, therefore defining normal forms up to α-equivalence.

We write $s\downarrow_\beta$ for the unique β-*normal form* of the term *s*, $s\uparrow^\eta$ for the unique η-*long form* of *s* wrt. η-expansion, $s\downarrow_\eta$ for the unique η-normal form of *s* wrt. η-reduction, and $u \updownarrow_\beta^\eta$ for its unique normal form with respect to β-reductions and η-expansions, also called η-*long normal form*. Terms in η-long normal form are called *normalized*.

Normal terms

Lemma

Normalized terms are of the following two forms:
(i) $\lambda\overline{x}:\overline{\rho}.@(X, v_1, \ldots, v_p)$*, for some* $\overline{x}:\overline{\rho}$*,* $X : \tau_1 \to \ldots \to \tau_p \to \tau \in \mathcal{X}$ *where* $p > 0$ *and* τ *is a data type or a type variable, and normalized terms* $v_1, \ldots, v_p$*, omitting* $@()$ *when* $p = 0$*;*
(ii) $\lambda\overline{x}:\overline{\rho}.@(F(u_1, \ldots, u_n), v_1, \ldots, v_p)$*, for some* $\overline{x}:\overline{\rho}$*,* $F \in \mathcal{F}_{\sigma_1\times\ldots\times\sigma_n\to(\tau_1\to\ldots\to\tau_p\to\tau)}$ *where* τ *is a data type or a type variable, and normalized terms* $u_1, \ldots, u_n, v_1, \ldots, v_p$*, omitting* $@()$ *when* $p = 0$ *and the other two parentheses when* $n = 0$*.*

Normal terms

In normalized terms, the first argument of an application cannot be in η-long form.

Definition

A term t is *tail expanded* (resp. *tail normal*) if:
(i) $t \in \mathcal{X}$, or
(ii) $t = f(u_1, \dots, u_n)$, $u_1, \dots, u_n$ are in η-long form (normalized), or
(iii) $t = @(u_1, \dots, u_n)$, u_1 is tail expanded (tail normal and not an abstraction) and $u_2, \dots, u_n$ in η-long form, or
(iv) $t = \lambda x : \sigma.u$, u is tail expanded (tail normal) and not of the form $@(v, x)$ with $x \notin \mathcal{V}ar(v)$.

Lemma

Every normalized term t of type σ contains a $\beta\eta$-equivalent tail normal subterm of type σ, which is a proper subterm iff σ is functional.

$t\uparrow^{\neq\Lambda}$ (resp. $t\updownarrow^{\neq\Lambda}$) the unique tail expanded (resp. tail normal) term s η-equivalent ($\beta\eta$-) to t :

$$(\lambda x.u)\uparrow^{\neq\Lambda} = \lambda x.(u\uparrow^{\neq\Lambda}) \text{ with } x \notin \mathcal{V}ar(v) \text{ if } u = @(v,x)$$

$$@(u_1, u_2, \ldots, u_n)\uparrow^{\neq\Lambda} = @(u_1\uparrow^{\neq\Lambda}, u_2\uparrow, \ldots, u_n\uparrow)$$

$$(\lambda x.u)\uparrow = \lambda x.(u\uparrow) \qquad @(\overline{u})\uparrow = (@(\overline{u})\uparrow^{\neq\Lambda})\uparrow^{\Lambda}$$

$$f(\overline{u})\uparrow^{\neq\Lambda} = f(\overline{u}\uparrow) \qquad f(\overline{u})\uparrow = (f(\overline{u})\uparrow^{\neq\Lambda})\uparrow^{\Lambda}$$

Propriété: s tail normal, ξ type substitution: $s\xi\updownarrow^{\neq\Lambda} = s\xi\uparrow^{\neq\Lambda}$.

Definition

A *normal rewrite rule* is a rewrite rule $\Gamma \vdash l \rightarrow r : \sigma$ such that l and r are tail normal terms. A *normal term rewriting system* is a set of normal rewrite rules.

Definition

A tail normal term s such that $\Gamma \vdash_{\mathcal{F}} s : \sigma$ rewrites to a term t at position p with the tail normal rule $\Gamma_i \vdash l_i \rightarrow r_i : \sigma_i$, the type substitution ξ and the term substitution γ,

$$\Gamma \vdash s \xrightarrow[\Gamma_i\xi \vdash l_i\xi \rightarrow r_i\xi : \sigma_i\xi]{p} t$$

if the following conditions are satisfied :

(i) $\mathcal{D}om(\gamma) \subseteq \Gamma_i\xi$

(ii) $\Gamma_i\xi \cdot \mathcal{R}an(\gamma) \subseteq \Gamma_{s|_p}$

(iii) $\begin{cases} s|_p \text{ is tail normal} \\ s|_p \longleftrightarrow^*_{\beta\eta} l_i\xi \uparrow^{\neq\eta} \gamma \end{cases}$

(iv) $t = (s[r_i\xi \uparrow^{\neq\Lambda} \gamma \downarrow_\beta]_p)\downarrow_\beta$

Assuming that $s|_p$ is tail normal is not a restriction since we can always chose p fulfilling the property. Computing t requires climbing up $s[r_i \xi \uparrow^{\neq \Lambda} \gamma \downarrow_\beta]_p$ from the position p to the root as long as the symbol on the path is the application. No climbing is needed when the output type of a function symbol is a data type, a frequently met assumption.
Higher-order pattern matching is open for order strictly bigger than 4, but is decidable in linear time when the lefthand sides of rules are patterns in the sense of Miller.

Lemma

Let s be a tail normal term such that $\Gamma \vdash_{\mathcal{F}} s : \sigma$, $\Gamma \vdash s \longrightarrow_{R^{\eta}_{\beta}} t$. *Then* $\Gamma \vdash_{\mathcal{F}} t : \sigma$, *t is tail normal.*

Proof.

By Lemma 3, $\Gamma_i\xi \vdash_{\mathcal{F}} l_i\xi : \sigma_i\xi$. By conditions (i) and (ii) in Definition 20, γ is compatible with $\Gamma_i\xi$. Hence, by Lemma 11, $\Gamma_i\xi \cdot \mathcal{R}an(\gamma) \vdash_{\mathcal{F}} l_i\xi\gamma : \sigma_i\xi$. By condition (ii) and lemma 4, $\Gamma_{s|_p} \vdash_{\mathcal{F}} l_i\xi\gamma : \sigma_i\xi$, and therefore, by condition (iii), $\Gamma_{s|_p} \vdash_{\mathcal{F}} s|_p : \sigma_i\xi$. Similarly, $\Gamma_{s|_p} \vdash_{\mathcal{F}} r_i\xi\gamma : \sigma_i\xi$. By lemma 6, $\Gamma \vdash_{\mathcal{F}} s[r_i\xi\gamma]_p : \sigma_i\xi$. Using now condition (iv) and Lemma 14 yields $\Gamma \vdash_{\mathcal{F}} t : \sigma$. □

Symbolic derivation

$$
\begin{array}{rcl}
D(\lambda x.y) & \rightarrow & \lambda x.0 \quad \text{if } y \neq x \\
D(\lambda x.x) & \rightarrow & \lambda x.1 \\
D(\lambda x.sin(F\ x)) & \rightarrow & \lambda x.cos(F\ x) \times (D(F)\ x) \\
D(\lambda x.cos(F\ x)) & \rightarrow & \lambda x. - sin(F\ x) \times (D(F)\ x) \\
D(\lambda x.(F\ x) + (G\ x)) & \rightarrow & \lambda x.(D(F)\ x) + (D(G)\ x) \\
D(\lambda x.(F\ x) \times (G\ x)) & \rightarrow & \lambda x. \\
\multicolumn{3}{r}{(D(\lambda y.(F\ y))\ x) \times (G\ x) + (F\ x) \times (D(\lambda y.(G\ y))\ x)}
\end{array}
$$

Note that $D(\lambda x.sin(x)) =_\beta D(\lambda x.sin(\lambda y.y\ x))$, hence $D(\lambda x.sin(x)) \longrightarrow \lambda x.cos(\lambda x.x\ x) \times (D(\lambda x.x)\ x)\downarrow_\beta = \lambda x.cos(x) \times (\lambda x.1\ x) \longrightarrow \lambda x.cos(x) \times 1$, requiring higher-order matching for firing the third rule.

Higher-order normal ordering

Definition: a *higher-order normal reduction ordering*

$\succ$ is a well-founded ordering of the set of judgements such that:
(i) *tail monotonicity*: $(\Gamma \vdash_{\mathcal{F}} s : \sigma) \succ (\Gamma \vdash_{\mathcal{F}} t : \sigma)$ implies $(\Gamma \cdot \Gamma' \vdash_{\mathcal{F}} u[s] : \tau) \succ (\Gamma \cdot \Gamma' \vdash_{\mathcal{F}} u[t] : \tau)$ $\forall s, t$ tail expanded terms and $\forall \Gamma' \vdash_{\mathcal{F}} u[x : \sigma] : \tau$ such that Γ, Γ' are compatible, and $u[s]$ and $u[t]$ are tail expanded;
(ii) *stability* for all terms;
(iii) *compatibility* for all tail expanded terms;
(iv) *tail functionality*: $(\Gamma \vdash_{\mathcal{F}} s : \sigma \longrightarrow_{\beta} t : \sigma)$ implies $(\Gamma \vdash_{\mathcal{F}} s : \sigma) \succ (\Gamma \vdash_{\mathcal{F}} t : \sigma)$ for all tail expanded terms s and t.

Higher-order normal ordering

Definition

A subrelation $\succ_\beta^\eta$ of a higher-order normal reduction ordering $\succ$ is said to be
(i) *β-stable* if $(\Gamma \vdash_\mathcal{F} s) \succ_\beta^\eta (\Gamma \vdash_\mathcal{F} t)$ implies $(\Gamma \cdot \mathcal{R}an(\gamma) \vdash_\mathcal{F} s\gamma{\downarrow_\beta}) \succ (\Gamma \cdot \mathcal{R}an(\gamma) \vdash_\mathcal{F} t\gamma{\downarrow_\beta})$ for all tail normal terms s, t and tail normal substitution γ compatible with Γ;
(ii) *η-polymorphic* if $(\Gamma \vdash_\mathcal{F} s) \succ_\beta^\eta (\Gamma \vdash_\mathcal{F} t)$ implies $(\Gamma\xi \vdash_\mathcal{F} s\xi{\uparrow^{\neq\Lambda}}) \succ_\beta^\eta (\Gamma\xi \vdash_\mathcal{F} t\xi{\uparrow^{\neq\Lambda}})$ for all tail normal terms s, t and all type substitution ξ.

Theorem

Assume that $\succ$ is a higher-order normalized reduction ordering and that $\succ_{\beta}^{\eta}$ is a β-stable and η-polymorphic subrelation of $\succ$. Let $R = \{\Gamma_i \vdash l_i \rightarrow r_i : \sigma_i\}_{i \in I}$ be a higher-order rewrite system such that $(\Gamma_i \vdash_{\mathcal{F}} l_i) \succ_{\beta}^{\eta} (\Gamma_i \vdash_{\mathcal{F}} r_i)$ for every $i \in I$. Then the relation $\longrightarrow_{R_{\beta}^{\eta}}$ is strongly normalizing.

Proof

Let $\Gamma \vdash_{\mathcal{F}} s \xrightarrow[\Gamma_i\xi \vdash l_i\xi \to r_i\xi : \sigma_i\xi]{p} t$. By confluence of $\updownarrow^{\neq\Lambda}$,
$s|_p = l_i\xi \uparrow^{\neq\Lambda} \gamma \downarrow_\beta$ and $s = s[l_i\xi \uparrow^{\neq\Lambda} \gamma \downarrow_\beta]_p$.
By η-polymorphism: $\Gamma_i \vdash_{\mathcal{F}} l_i \succ_\beta^\eta r_i$ implies
$\Gamma_i\xi \vdash_{\mathcal{F}} l_i\xi \uparrow^{\neq\Lambda} \succ_\beta^\eta r_i\xi \uparrow^{\neq\Lambda}$. By β-stability:
$\Gamma_i\xi \cdot \mathcal{R}an(\gamma) \vdash_{\mathcal{F}} (l_i\xi \uparrow^{\neq\Lambda})\gamma \downarrow_\beta \succ \vdash_{\mathcal{F}} (r_i\xi \uparrow^{\neq\Lambda})\gamma \downarrow_\beta$.
By compatibility:
$\Gamma_{s|_p} \vdash_{\mathcal{F}} (l_i\xi \uparrow^{\neq\Lambda} \gamma) \downarrow_\beta \succ (r_i\xi \uparrow^{\neq\Lambda})\gamma \downarrow_\beta$.
By monotonicity:
$\Gamma_{s|_p} \cdot \Gamma \vdash_{\mathcal{F}} s[l_i\xi \uparrow^{\neq\Lambda} \gamma \downarrow_\beta]_p = s \succ s[r_i\xi \uparrow^{\neq\Lambda} \gamma \downarrow_\beta]_p$.
By compatibility:
$\Gamma \vdash_{\mathcal{F}} s[l_i\xi \uparrow^{\neq\Lambda} \gamma \downarrow_\beta]_p = s \succ s[r_i\xi \uparrow^{\neq\Lambda} \gamma \downarrow_\beta]_p$.
By tail functionnality: $\Gamma \vdash_{\mathcal{F}} s[r_i\xi \uparrow^{\neq\Lambda} \gamma \downarrow_\beta]_p \succ t$.
By transitivity: $\Gamma \vdash_{\mathcal{F}} s \succ t$.