The computability path ordering

(joint work with J.-P. Jouannaud and A. Rubio)

Frédéric Blanqui

INVENTEURS DU MONDE NUMÉRIQUE

Deduc⊢eam

LSV seminar, 13 January 2015, Cachan, France

Goal

<u>automate</u> the proof of termination of <u>higher-order</u> rewrite systems

Outline

1. Introduction to higher-order rewriting

2. Extending RPO to λ-terms

Higher-order rewriting = rewriting on λ-terms

$$\boxed{x \mid \mathsf{f} \mid \lambda x.t \mid tt}$$

$$(\lambda x.t)u \rightarrow_\beta t_x^u$$

$$\lambda x.tx \rightarrow_\eta t \text{ if } x \notin \mathrm{FV}(t)$$

$$\mathsf{f}\vec{l} \rightarrow_{\mathcal{R}} r$$

Example: map function on lists

- nil : $\mathbb{L}\alpha$
- cons : $\alpha \Rightarrow \mathbb{L}\alpha \Rightarrow \mathbb{L}\alpha$
- map : $(\alpha \Rightarrow \beta) \Rightarrow \mathbb{L}\alpha \Rightarrow \mathbb{L}\beta$

$$\begin{array}{rcl} \mathsf{map}\ F\ \mathsf{nil} & \rightarrow_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ F\ (\mathsf{cons}\ x\ l) & \rightarrow_{\mathcal{R}} & \mathsf{cons}\ (F\ x)\ (\mathsf{map}\ F\ l) \end{array}$$

map $(\lambda x.2 * x)$ (cons 5 l)
$\rightarrow_{\mathcal{R}}$ cons $((\lambda x.2 * x)\ 5)$ (map $(\lambda x.2 * x)$ l)
$\rightarrow_{\beta}$ cons $(2 * 5)$ (map $(\lambda x.2 * x)$ l)
...

Example: recursor on natural numbers

- $0 : \mathbb{N}$
- $\mathsf{s} : \mathbb{N} \Rightarrow \mathbb{N}$
- $\mathsf{natrec} : \alpha \Rightarrow (\mathbb{N} \Rightarrow \alpha \Rightarrow \alpha) \Rightarrow \mathbb{N} \Rightarrow \alpha$

$$
\begin{array}{rcl}
\mathsf{natrec}\ U\ V\ 0 & \rightarrow_{\mathcal{R}} & U \\
\mathsf{natrec}\ U\ V\ (\mathsf{s}\ n) & \rightarrow_{\mathcal{R}} & V\ n\ (\mathsf{natrec}\ U\ V\ n)
\end{array}
$$

Example: recursor on ordinals

- $0 : \mathbb{O}$
- $\mathsf{s} : \mathbb{O} \Rightarrow \mathbb{O}$
- $\mathsf{lim} : (\mathbb{N} \Rightarrow \mathbb{O}) \Rightarrow \mathbb{O}$
- ordrec :
  $\alpha \Rightarrow (\mathbb{O} \Rightarrow \alpha \Rightarrow \alpha) \Rightarrow ((\mathbb{N} \Rightarrow \mathbb{O}) \Rightarrow (\mathbb{N} \Rightarrow \alpha) \Rightarrow \alpha) \Rightarrow \mathbb{O} \Rightarrow \alpha$

$$
\begin{array}{rcl}
\mathsf{ordrec}\ U\ V\ W\ 0 & \rightarrow_{\mathcal{R}} & U \\
\mathsf{ordrec}\ U\ V\ W\ (\mathsf{s}\ x) & \rightarrow_{\mathcal{R}} & V\ x\ (\mathsf{ordrec}\ U\ V\ W\ x) \\
\mathsf{ordrec}\ U\ V\ W\ (\mathsf{lim}\ F) & \rightarrow_{\mathcal{R}} & W\ F\ (\lambda n.\mathsf{ordrec}\ U\ V\ W\ (F\ n))
\end{array}
$$

Example: dependent choice operator

"Verifying Process Algebra Proofs in Type Theory", Sellink (1993):

- $+ : \mathbb{P} \Rightarrow \mathbb{P} \Rightarrow \mathbb{P}$
- $\Sigma : (\mathbb{D} \Rightarrow \mathbb{P}) \Rightarrow \mathbb{P}$
- $; : \mathbb{P} \Rightarrow \mathbb{P} \Rightarrow \mathbb{P}$
- ...

$$
\begin{array}{rcl}
\Sigma(\lambda d.P) & \rightarrow_{\mathcal{R}} & P \\
\Sigma X + Xd & \rightarrow_{\mathcal{R}} & \Sigma X \\
\Sigma(\lambda d.Xd + Yd) & \rightarrow_{\mathcal{R}} & \Sigma X + \Sigma Y \\
\Sigma X; P & \rightarrow_{\mathcal{R}} & \Sigma(\lambda d.Xd; P)
\end{array}
$$

...

Example: formal derivation

- $\mathsf{sin}, \mathsf{cos} : \mathbb{R} \Rightarrow \mathbb{R}$
- $+, \times : \mathbb{R} \Rightarrow \mathbb{R} \Rightarrow \mathbb{R}$
- $\mathsf{D} : (\mathbb{R} \Rightarrow \mathbb{R}) \Rightarrow (\mathbb{R} \Rightarrow \mathbb{R})$
- ...

$$\begin{array}{rcl}
\mathsf{D}\ (\lambda x.V) & \rightarrow_{\mathcal{R}} & \lambda x.O \\
\mathsf{D}\ (\lambda x.x) & \rightarrow_{\mathcal{R}} & \lambda x.1 \\
\mathsf{D}\ (\lambda x.F\ x + G\ x) & \rightarrow_{\mathcal{R}} & \lambda x.\mathsf{D}\ F\ x + \mathsf{D}\ G\ x \\
\mathsf{D}\ (\lambda x.\mathsf{sin}\ (F\ x)) & \rightarrow_{\mathcal{R}} & \lambda x.\mathsf{cos}\ (F\ x) \times \mathsf{D}\ F\ x
\end{array}$$

...

Example: recursor on continuations

- D : $\mathbb{C}$
- C : $((\mathbb{C} \Rightarrow \mathbb{L}) \Rightarrow \mathbb{L}) \Rightarrow \mathbb{C}$
- contrec :
  $\alpha \Rightarrow (((\mathbb{C} \Rightarrow \mathbb{L}) \Rightarrow \mathbb{L}) \Rightarrow ((\alpha \Rightarrow \mathbb{L}) \Rightarrow \mathbb{L}) \Rightarrow \alpha) \Rightarrow \mathbb{C} \Rightarrow \alpha$
- ex : $\mathbb{C} \Rightarrow \mathbb{L}$

$$
\begin{array}{rcl}
\mathsf{contrec}\ U\ V\ \mathsf{D} & \rightarrow_{\mathcal{R}} & U \\
\mathsf{contrec}\ U\ V\ (\mathsf{C}\ F) & \rightarrow_{\mathcal{R}} & W\ F\ (\lambda x.F(\lambda y.x\ (\mathsf{contrec}\ U\ V\ y))) \\
\mathsf{ex}\ (\mathsf{C}\ F) & \rightarrow_{\mathcal{R}} & F\ \mathsf{ex}
\end{array}
$$

The higher-order rewriting zoo

CRS	Combinatory Reduction Systems	1980	Klop
ERS	Expression Reduction Systems	1990	Khasidashvili
HOASL	Higher-Order Alg. Spec. Languages	1991	Jouannaud and Okada
HRS	Higher-order Rewrite Systems	1991	Nipkow
HORS	Higher-Order Rewrite Systems	1994	Van Oostrom

rewrite relations with matching modulo $\beta\eta$:

HRS	$\rightarrow_{\mathcal{R}}\rightarrow_{\beta}^{!}$
CRS	$\rightarrow_{\mathcal{R}}\rightarrow_{\beta}^{*}$
HOASL	$\rightarrow_{\mathcal{R}} \cup \rightarrow_{\beta}$

Why matching modulo $\beta\eta$?

with the rule D $(\lambda x.\mathsf{sin}\ (F\ x)) \rightarrow_{\mathcal{R}} \lambda x.\mathsf{cos}\ (F\ x) \times$ D $F\ x$

$\not\leftarrow_{\mathcal{R}}$ D sin $\leftarrow_{\eta}$ D $(\lambda x.\mathsf{sin}\ x) \leftarrow_{\beta}$ D $(\lambda x.\mathsf{sin}\ ((\lambda x.x)\ x)) \rightarrow_{\mathcal{R}}$

Automated termination techniques for HOR

- syntactic recursion schema (Jouannaud and Okada 1991), computability closure (B., Jouannaud and Okada 1999, B. 2001)
- polynomial interpretation (Van de Pol 1996, Fuhs and Kop 2012)
- inclusion in a well-founded relation (Jouannaud and Rubio 1999)
- size annotations (Giménez 1996, Hughes, Pareto and Sabry 1996, Abel 2002, Barthe et al 2004, B. 2004)
- size change principle (Jones and Bohr 2004, Wahlstedt 2007)
- semantic labeling (Hamana 2007, B. and Roux 2009)
- dependency pairs (Kusakari and Sakai 2005, B. 2006, Kop 2010)

Relations between these techniques

- the notion of computability closure can be extended to handle size annotations (B. 2004), improve HORPO (Jouannaud and Rubio 1999) and dependency pairs (Kusakari et al. 2009)
- size annotations are a particular case of semantic labeling (B. and Roux 2009)
- HORPO is the fixpoint of the computability closure (B. 2006)

Outline

Recursive path ordering (Dershowitz 1979)

given a well-founded quasi-ordering $\geq_{\mathcal{F}}$ on function symbols

$\boxed{t = \mathsf{f}\vec{t} > u}$ if either:

($\mathcal{F}\triangleright$) $t_i \geq u$ for some i

($\mathcal{F}>$) $u = \mathsf{g}\vec{u}$, $\mathsf{f} >_{\mathcal{F}} \mathsf{g}$ and P: $(\forall i)[t > u_i]$

($\mathcal{F}=$) $u = \mathsf{g}\vec{u}$, $\mathsf{f} \simeq_{\mathcal{F}} \mathsf{g}$, $\vec{t} >_{\mathrm{mul}} \vec{u}$ and P

extension to $>_{\mathrm{lex}}$ by Kamin and Lévy (1980)

Termination proofs:

- Dershowitz (1979): Kruskal tree theorem
- Lescanne (1982): inductive proof + axiom of choice
- Buchholtz (1995): inductive proof
- Jouannaud and Rubio (1999): based on Tait and Girard computability predicates ($\Leftrightarrow$ Buchholtz)

Extension to λ-calculus?

First attempts. . .

- 1992: Loria-Sáenz and Steinbach
- 1995: Lysne and Piris
- 1996: Jouannaud and Rubio

Importance of types

pattern-matching on negative types leads to non-termination (Mendler 1987):

- $\mathsf{c} : (\mathbb{T} \Rightarrow \mathbb{B}) \Rightarrow \mathbb{T}$
- $\mathsf{f} : \mathbb{T} \Rightarrow (\mathbb{T} \Rightarrow \mathbb{B})$

$$\mathsf{f}\ (\mathsf{c}\ x) \quad \rightarrow_{\mathcal{R}} \quad x$$

let $\omega : \mathbb{T} \Rightarrow \mathbb{B} := \lambda x.\mathsf{f}xx$

$\mathsf{f}\ (\mathsf{c}\ \omega)\ (\mathsf{c}\ \omega) \rightarrow_{\mathcal{R}} \omega\ (\mathsf{c}\ \omega) \rightarrow_{\beta} \mathsf{f}\ (\mathsf{c}\ \omega)\ (\mathsf{c}\ \omega) \ldots$

HORPO-99 (Jouannaud and Rubio 1999)

given a well-founded quasi-ordering $\geq_{\mathcal{F}}$ on function symbols

$\boxed{t > u}$ if $\tau(t) = \tau(u)$ and either:

($\mathcal{F}\rhd$) $t = \mathsf{f}\vec{t}$ and $t_i \geq u$ for some i

($\mathcal{F}>$) $t = \mathsf{f}\vec{t}$, $u = \mathsf{g}\vec{u}$, $\mathsf{f} >_{\mathcal{F}} \mathsf{g}$ and P

($\mathcal{F}=$) $t = \mathsf{f}\vec{t}$, $u = \mathsf{g}\vec{u}$, $\mathsf{f} \simeq_{\mathcal{F}} \mathsf{g}$, $\vec{t} >_{\mathrm{stat(f)}} \vec{u}$ and P

($\mathcal{F}\rhd$) ($\mathcal{F}>$) ($\mathcal{F}=$)

($\mathcal{F}@$) $t = \mathsf{f}\vec{t}$, $u = u_1 \ldots u_n$, $n \geq 2$ and P

($@=$) $t = t_1 t_2$, $u = u_1 u_2$ and $t_1 t_2 >_{\mathrm{mul}} u_1 u_2$

($\lambda=$) $t = \lambda x.a$, $u = \lambda x.b$ and $a > b$

where P is: $(\forall i)[t > u_i \vee (\exists j) t_j \geq u_i]$

Example with HORPO-99

$$\begin{array}{rcl} \Sigma(\lambda d.Xd + Yd) & \rightarrow_{\mathcal{R}} & \Sigma(\lambda d.Xd) + \Sigma(\lambda d.Yd) \\ \Sigma X; P & \rightarrow_{\mathcal{R}} & \Sigma(\lambda d.Xd; P) \end{array}$$

- $\Sigma(\lambda d.Xd + Yd) > \Sigma(\lambda d.Xd) + \Sigma(\lambda d.Yd)$
  because $\tau(\Sigma(\lambda d.Xd + Yd)) = \tau(\Sigma(\lambda d.Xd) + \Sigma(\lambda d.Yd))$ and, by taking $\Sigma >_{\mathcal{F}} +$, after $(\mathcal{F}>)$:
- $\Sigma(\lambda d.Xd + Yd) > \Sigma(\lambda d.Xd)$ and $\Sigma(\lambda d.Xd + Yd) > \Sigma(\lambda d.Yd)$
  because $\tau(\Sigma(\lambda d.Xd + Yd)) = \tau(\Sigma(\lambda d.Xd)$ and, after $(\mathcal{F}=)$:
- $\lambda d.Xd + Yd > \lambda d.Xd$ because, after $(\lambda=)$:
- $Xd + Yd > Xd$ after $(\mathcal{F}\triangleright)$

- $\Sigma X; P \not> \Sigma(\lambda d.Xd; P)$

HORPO-07 (Jouannaud and Rubio 2007)

given:
- a well-founded quasi-ordering $\geq_{\mathcal{F}}$ on function symbols
- a well-founded quasi-ordering $\geq_{\mathcal{T}}$ on types such that ... (a sort can be bigger than an arrow type)

$\boxed{t : T > u : U}$ if $T \geq_{\mathcal{T}} U$ and either:

$(\mathcal{F}\triangleright)$ $(\mathcal{F}>)$ $(\mathcal{F}=)$ $(\mathcal{F}@)$ $(@=)'$ $(\lambda=)$

$(@=)'$ $t = t_1 t_2$, $u = u_1 \ldots u_n$, $n \geq 2$ and $t_1 t_2 >_{\mathrm{mul}} u_1 \ldots u_n$

$(\lambda=)$ $t = \lambda x.a$, $u = \lambda y.b$, $\tau(x) \simeq_{\mathcal{T}} \tau(y)$, $x \notin \mathrm{FV}(u)$ and $a > b_y^x$

$(@\triangleright)$ $t = t_1 t_2$ and $t_i \geq u$ for some i

$(\lambda\triangleright)$ $t = \lambda x.a$, $x \notin \mathrm{FV}(u)$ and $a \geq u$

$(\mathcal{F}\lambda)$ $t = \mathsf{f}\vec{t}$, $u = \lambda x.b$, $x \notin \mathrm{FV}(b)$ and $t > b$

$(@\beta)$ $t = (\lambda x.a)b$ and $a_x^b \geq u$

$(\lambda\eta)$ $t = \lambda x.ax$, $x \notin \mathrm{FV}(a)$ and $a \geq u$

CPO (B., Jouannaud and Rubio 2014)

improve HORPO-07 by:

- fixing the conditions on $\geq_{\mathcal{T}}$
- reducing the number of type comparisons
- handling bound variables
- handling recursion on strictly positive inductive types
- handling symbols smaller than application and abstraction

$>$ is now defined as $>^{\emptyset}_{\tau}$ where:

- for any relation $>$, $t >_{\tau} u$ if $t > u$ and $\tau(t) \geq_{\mathcal{T}} \tau(u)$
- given a finite set X of variables, $>^{X}$ is defined inductively as...

Admissible type orderings

A relation $\geq_{\mathcal{T}}$ on types is admissible if:

1. $\geq_{\mathcal{T}}$ is an ordering containing $\rhd_r$, where $T \Rightarrow U \rhd_r U$
2. $>_{\mathcal{T}} \cup \rhd_l$ is well-founded, where $T \Rightarrow U \rhd_l T$
3. if $T \Rightarrow U >_{\mathcal{T}} V$ then $U >_{\mathcal{T}} V$ or, $V = T \Rightarrow U'$ and $U >_{\mathcal{T}} U'$

Example: some sub-relation of RPO
given a well-founded ordering $>_{\mathcal{S}}$ on sorts, the smallest ordering $>_{\mathcal{T}}$ containing $>_{\mathcal{S}}$ and $\rhd_r$ that is right-monotone ($U >_{\mathcal{T}} U'$ implies $T \Rightarrow U >_{\mathcal{T}} T \Rightarrow U'$) is admissible

Core CPO part 1/3

$\boxed{t = \mathsf{f}\vec{t} >^{X} u}$ if either:

($\mathcal{F}\rhd$) $t_i \geq_{\tau}^{\emptyset} u$ for some i

($\mathcal{F}>$) $u = \mathsf{g}\vec{u}$, $\mathsf{f} >_{\mathcal{F}} \mathsf{g}$ and P: $t >^{X} u_i$ for all i

($\mathcal{F}=$) $u = \mathsf{g}\vec{u}$, $\mathsf{f} \simeq_{\mathcal{F}} \mathsf{g}$, $\vec{t}\ (>_{\tau}^{\emptyset})_{\mathrm{stat(f)}}\ \vec{u}$ and P

($\mathcal{F}$@) $u = u_1 u_2$ and P

($\mathcal{F}\lambda$) $u = \lambda x.b$ and $t >^{X\cup\{x\}} b$ and $x \notin \mathrm{FV}(t)$

($\mathcal{F}\mathcal{X}$) $u \in X$

Core CPO part 2/3

$\boxed{t = t_1 t_2 >^X u}$ if either:

(@▷) $t_1 \geq^X u$ or $t_2 \geq_\tau^X u$

(@=) $u = u_1 u_2$, $\vec{t}\ (>_\tau^\emptyset)_{\mathrm{mul}}\ \vec{u}$

(@λ) $u = \lambda x.b$, $x \notin \mathrm{FV}(b)$ and $t >^X b$

(@$\mathcal{X}$) $u \in X$

(@β) $t_1 = \lambda x.a$ and $a_x^{t_2} \geq^X u$

Core CPO part 3/3

$\boxed{t = \lambda x.a >^X u}$ if either:

($\lambda\rhd$) $a \geq_\tau^X u$ and $x \notin \mathrm{FV}(u)$

($\lambda=$) $u = \lambda x.b$ and $a >^X b$

($\lambda\neq$) $u = \lambda y.b$, $\tau(x) \neq \tau(y)$, $y \notin \mathrm{FV}(b)$ and $t >^X u$

($\lambda\mathcal{X}$) $u \in X$

($\lambda\eta$) $a = vx$, $x \notin \mathrm{FV}(v)$ and $v \geq^X u$

Example with Core CPO

- $\mathsf{C} : ((\mathbb{C} \Rightarrow \mathbb{L}) \Rightarrow \mathbb{L}) \Rightarrow \mathbb{C}$
- $\mathsf{ex} : \mathbb{C} \Rightarrow \mathbb{L}$

$$\mathsf{ex}\ (\mathsf{C}\ F) \quad \rightarrow_{\mathcal{R}} \quad F\ \mathsf{ex}$$

- $\mathsf{ex}\ (\mathsf{C}\ F) >_{\tau}^{\emptyset} F\ \mathsf{ex}$
  because $\tau(\mathsf{ex}\ (\mathsf{C}\ F)) = \tau(F\ \mathsf{ex})$ and, after $(@\rhd)$:
- $\mathsf{C}\ F >_{\tau}^{\emptyset} F\ \mathsf{ex}$, because
  $\tau(\mathsf{C}\ F) \geq_{\mathcal{T}} \tau(F\ \mathsf{ex})$ if one takes $\mathbb{C} \geq_{\mathcal{T}} \mathbb{L}$ and, after $(\mathcal{F}@)$:
- $\mathsf{C}\ F : \mathbb{C} >^{\emptyset} F : (\mathbb{C} \Rightarrow \mathbb{L}) \Rightarrow \mathbb{L}$ after $(\mathcal{F}\rhd)$
- $\mathsf{C}\ F : \mathbb{C} > \mathsf{ex} : \mathbb{C} \Rightarrow \mathbb{L}$ after $(\mathcal{F}>)$ if one takes $\mathsf{C} >_{\mathcal{F}} \mathsf{ex}$

Tightness of Core CPO part 1/3

$\boxed{t = \mathrm{f}\vec{t} >^X u}$ if either:

($\mathcal{F}\rhd$) $t_i \geq_\tau^\emptyset u$ for some i
replacing $\geq_\tau^\emptyset$ by $\geq_\tau^X$ or $\geq$ leads to non-termination

($\mathcal{F}>$) $u = \mathrm{g}\vec{u}$, $\mathrm{f} >_\mathcal{F} \mathrm{g}$ and P

($\mathcal{F}=$) $u = \mathrm{g}\vec{u}$, $\mathrm{f} \simeq_\mathcal{F} \mathrm{g}$, $\vec{t}\ (>_\tau^\emptyset)_{\mathrm{stat(f)}}\ \vec{u}$ and P
replacing $>_\tau^\emptyset$ by $>_\tau^X$ or $>$ leads to non-termination

($\mathcal{F}$@) $u = u_1 u_2$ and P
replacing $>^X$ by $(>^X)^+$ leads to non-termination

($\mathcal{F}\lambda$) $u = \lambda x.b$ and $t >^{X\cup\{x\}}$ and $x \notin \mathrm{FV}(t)$

($\mathcal{F}\mathcal{X}$) $u \in X$

Tightness of Core CPO part 2/3

$\boxed{t = t_1 t_2 >^X u}$ if either:

($@\rhd$) $t_1 \geq^X u$ or $t_2 \geq^X_\tau u$
replacing $\geq^X_\tau$ by $\geq^X$ leads to non-termination

($@=$) $u = u_1 u_2$, $\vec{t}(>^\emptyset_\tau)_{\mathrm{mul}}\vec{u}$
replacing $>^\emptyset_\tau$ by $>^X_\tau$ or $>$ leads to non-termination

($@\lambda$) $u = \lambda x.b$, $x \notin \mathrm{FV}(b)$ and $t >^X b$
replacing $>^X$ by $>^{X \cup \{x\}}$ leads to non-termination

($@\mathcal{X}$) $u \in X$

($@\beta$) $t_1 = \lambda x.a$ and $a_x^{t_2} \geq^X u$

Tightness of Core CPO part 3/3

$\boxed{t = \lambda x.a >^X u}$ if either:

($\lambda\rhd$) $a \geq_\tau^X u$ and $x \notin \mathrm{FV}(u)$
replacing $\geq_\tau^X$ by $\geq^X$ leads to non-termination

($\lambda=$) $u = \lambda x.b$ and $a >^X b$

($\lambda\neq$) $u = \lambda y.b$, $\tau(x) \neq \tau(y)$, $y \notin \mathrm{FV}(b)$ and $t >^X u$
replacing $>^X$ by $>^{X \cup \{y\}}$ or
removing the condition $\tau(x) \neq \tau(y)$ leads to non-termination

($\lambda\mathcal{X}$) $u \in X$

($\lambda\eta$) $a = vx$, $x \notin \mathrm{FV}(v)$ and $v \geq^X u$

Handling strictly positive inductive types

$\boxed{t = \mathsf{f}\vec{t} >^X u}$ if either:

...

$(\mathcal{F}\rhd)$ $t_i \unrhd_b^s \unrhd_a \geq_\tau u$ for some i

$(\mathcal{F}=)$ $u = \mathsf{g}\vec{u}$, $\mathsf{f} \simeq_\mathcal{F} \mathsf{g}$, $\vec{t}\ (>_\tau^\emptyset \cup \rhd_@^X \geq_\tau^\emptyset)_{\mathrm{stat(f)}}\ \vec{u}$ and P

$\unrhd_b^s$ and $\unrhd_a$ are restricted subterm relations

$\rhd_@^X$ is Coquand' structurally smaller relation (1992)

they all depend on the types of symbols
(e.g. $\mathsf{f}\vec{t} : \mathbb{B} \rhd_a t_i : T_i$ only if $\mathbb{B}$ occurs only positively in T_i)

Handling strictly positive inductive types

$$\Sigma X; P \quad \rightarrow_{\mathcal{R}} \quad \Sigma(\lambda d.Xd; P)$$

- $\Sigma X; P >^{\emptyset}_{\tau} \Sigma(\lambda d.Xd; P)$ by $(\mathcal{F}>)$ if one takes $; >_{\mathcal{F}} \Sigma$ because:
- $\Sigma X; P >^{\emptyset} \lambda d.Xd; P$ by $(\mathcal{F}\lambda)$ because:
- $\Sigma X; P >^{\{d\}} Xd; P$ by $(\mathcal{F}=)$ because:
- $\Sigma X \rhd^{\{d\}}_{@} Xd$

Handling "small" symbols: $\mathcal{F} = \mathcal{F}_b \uplus \mathcal{F}_s$

$\boxed{t = t_1 t_2 >^X u}$ if either: ...

($@\mathcal{F}_s$) $u = \mathsf{g}\vec{u}$, $\mathsf{g} \in \mathcal{F}_s$ and P_τ: $t >^X_\tau u_i$ for all i

$\boxed{t = \lambda x.a >^X u}$ if either: ...

($@\mathcal{F}_s$) $u = \mathsf{g}\vec{u}$, $\mathsf{g} \in \mathcal{F}_s$ and P_τ

$\boxed{t = \mathsf{f}\vec{t} >^X u}$ with $\mathsf{f} \in \mathcal{F}_s$ if either:

($\mathcal{F}_s\triangleright$) $t_i \geq^\emptyset_\tau u$ for some i

($\mathcal{F}_s>$) $u = \mathsf{g}\vec{u}$, $\mathsf{g} \in \mathcal{F}_s$, $\mathsf{f} >_\mathcal{F} \mathsf{g}$ and P_τ

($\mathcal{F}_s=$) $u = \mathsf{g}\vec{u}$, $\mathsf{g} \in \mathcal{F}_s$, $\mathsf{f} \simeq_\mathcal{F} \mathsf{g}$, $\vec{t}\ (>^\emptyset_\tau \cup \triangleright^X_@ \trianglerighteq^\emptyset_\tau)_{\mathrm{stat(f)}}\ \vec{u}$ and P_τ

($\mathcal{F}_s@$) $u = u_1 u_2$ and P_τ

($\mathcal{F}_s\mathcal{X}$) $u \in X$

A few words on the termination proof - Part 1/3

The termination of $>^{\emptyset}_{\tau}$ is proved by extending the technique of Tait (1967) and Girard (1972):

1) we interpret every sort $\mathbb{B}$ by some set of terms $[\![\mathbb{B}]\!]$

- the interpretation of arrow types is fixed:
  $[\![U \Rightarrow V]\!] = \{t \in \mathcal{T} | \forall u \in [\![U]\!], tu \in [\![V]\!]\}$
- a term $t : T$ is *computable* if $t \in [\![T]\!]$

A few words on the termination proof - Part 2/3

2) we explicit conditions under which a set $[\![T]\!]$ satisfies:

(comp-sn) the elements of $[\![T]\!]$ are strongly normalizing wrt $>^{\emptyset}_{\tau}$
(comp-red) every $>^{\emptyset}_{\tau}$-reduct of $t \in [\![T]\!]$ is computable
mp-neutral) $t \in [\![T]\!]$ if $t : T$ is *neutral* and every $>^{\emptyset}_{\tau}$-reduct of t is computable
(comp-lam) $\lambda x.a \in [\![T]\!]$ if $T = U \Rightarrow V$ and, for every comp. $u : U$, a^u_x is comp.
omp-small) $\mathsf{f}\vec{t} \in [\![T]\!]$ if $\mathsf{f}\vec{t} : T$, $\mathsf{f} \in \mathcal{F}_s$ and $\vec{t}$ are computable

Examples:

1. $[\![U \Rightarrow V]\!]$ satisfies (comp-sn) if
   $[\![U]\!]$ satisfies (comp-neutral) and $[\![V]\!]$ satisfies (comp-sn)
2. $[\![U]\!]$ satisfies (comp-small) if $[\![U]\!]$ satisfies (comp-neutral),
   for every $U' <_{\mathcal{T}} U$, $[\![U']\!]$ satisfies (comp-small),
   for every small $\mathsf{f} : \vec{T} \Rightarrow U$, $[\![\vec{T}]\!]$ satisfies (comp-sn) and (comp-red)

A few words on the termination proof - Part 3/3

3) we prove that, for every type T, $[\![T]\!]$ satisfies all the computability properties

To break cyclic dependencies in conditions, we assume that for every small f : $\vec{T} \Rightarrow U$ with $U = \vec{U} \Rightarrow \mathbb{B}$:

1. every sort occurring in $\vec{T}$ is $\leq_{\mathcal{T}} \mathbb{B}$
2. and either:
   - $\vec{U}$ is empty and $\mathbb{B}$ has no *unsafe occurrences* in every T_i
   - $\vec{U}$ is not empty and every $T_i \leq_{\mathcal{T}} U$

small symbols are used for proving the termination of an extension of CPO to dependent types (Jouannaud and Li 2013)

Conclusion

- CPO is a new powerful extension of HORPO
- difficult to improve without giving up Tait-Girard's technique
- Prolog implementation available on Albert Rubio's web page
- details to appear in Logical Methods in Computer Science

Tightness of core CPO - Example 1/2

in $(\mathcal{F}\triangleright) t_i \geq_\tau^\emptyset u$ for some i, replace $\geq_\tau^\emptyset$ by $\geq_\tau^X$

with $\mathsf{a} : o >_\mathcal{F} \mathsf{f} : o \Rightarrow o >_\mathcal{F} \gamma : o \Rightarrow o \Rightarrow o$:

- $\mathsf{fa} >_\tau^\emptyset (\lambda x.\mathsf{f}x)\mathsf{a}$, because $\tau(\mathsf{fa}) = \tau((\lambda x.\mathsf{f}x)\mathsf{a})$, $(\mathcal{F}@)$ and:
  - $\mathsf{fa} >^\emptyset \mathsf{a}$, because $(\mathcal{F}\triangleright)$ and $\mathsf{a} \geq_\tau^\emptyset \mathsf{a}$
  - $\mathsf{fa} >^\emptyset \lambda x.\mathsf{f}x$, because $(\mathcal{F}\lambda)$ and:
  - $\mathsf{fa} >^{\{x\}} \mathsf{f}x$, because $(\mathcal{F}\triangleright)$ and:
  - $\mathsf{a} >_\tau^{\{x\}} \mathsf{f}x$, because $\tau(\mathsf{a}) = \tau(\mathsf{f}x)$, $(\mathcal{F}>)$ and:
  - $\mathsf{a} >^{\{x\}} x$, because $(\mathcal{F}\mathcal{X})$
- $(\lambda x.\mathsf{f}x)\mathsf{a} >_\tau^\emptyset \mathsf{fa}$, because $\tau((\lambda x.\mathsf{f}x)\mathsf{a}) = \tau(\mathsf{fa})$ and $(@\beta)$

Tightness of core CPO - Example 2/2

in $(\mathcal{F}\rhd) t_i \geq^{\emptyset}_{\tau} u$ for some i, replace $\geq^{\emptyset}_{\tau}$ by $\geq^{\emptyset}$

with $\mathsf{a} : o >_{\mathcal{F}} \mathsf{f} : o \Rightarrow o >_{\mathcal{F}} \gamma : o \Rightarrow o \Rightarrow o$:

- $\mathsf{fa} >^{\emptyset}_{\tau} (\lambda x.\mathsf{f}x)\mathsf{a}$, because $\tau(\mathsf{fa}) = \tau((\lambda x.\mathsf{f}x)\mathsf{a})$, $(\mathcal{F}@)$ and:
  - $\mathsf{fa} >^{\emptyset} \mathsf{a}$, because $(\mathcal{F}\rhd)$ and $\mathsf{a} \geq^{\emptyset} \mathsf{a}$
  - $\mathsf{fa} >^{\emptyset} \lambda x.\mathsf{f}x$, because $(\mathcal{F}\rhd)$ and:
  - $\mathsf{a} >^{\emptyset} \lambda x.\mathsf{f}x$, because $(\mathcal{F}\lambda)$ and:
  - $\mathsf{a} >^{\{x\}} \mathsf{f}x$, because $(\mathcal{F}>)$ and:
  - $\mathsf{a} >^{\{x\}} x$, because $(\mathcal{F}\mathcal{X})$
- $(\lambda x.\mathsf{f}x)\mathsf{a} >^{\emptyset}_{\tau} \mathsf{fa}$, because $\tau((\lambda x.\mathsf{f}x)\mathsf{a}) = \tau(\mathsf{fa})$ and $(@\beta)$

Accessible subterms

First, we assume every $\mathsf{f} : \vec{T} \Rightarrow \mathbb{B}$ equipped with a set $\mathrm{Acc}(\mathsf{f}) \subseteq \{1, \ldots, |\vec{T}|\}$ such that $i \in \mathrm{Acc}(\mathsf{f})$ only if:

- every sort occurring in T_i is $\leq \mathbb{B}$
- $\mathbb{B}$ occurs only positively in T_i (wrt $\Rightarrow$)

- $t \trianglerighteq_b^s u$ if $t \trianglerighteq u$, $\mathrm{FV}(u) \subseteq \mathrm{FV}(t)$ and $\tau(u)$ is a basic sort $\mathbb{B}$, i.e.:
  - for all $T <_{\mathcal{T}} \mathbb{B}$, T is a basic sort
  - for all $\mathsf{f} : \vec{U} \Rightarrow \mathbb{B}$ and $i \in \mathrm{Acc}(\mathsf{f})$, $U_i = \mathbb{B}$ or U_i is a basic sort

- $t \triangleright_a u$ if there are $\mathsf{f} : \vec{T} \Rightarrow \mathbb{B}$, $\vec{t}$ and $i \in \mathrm{Acc}(\mathsf{f})$ such that: $t = \mathsf{f}\vec{t}$ and $t_i \trianglerighteq_a u$

- $t \triangleright_{@}^{X} u$ if there are $\mathbb{B}$, v and $\vec{x}$ such that $t : \mathbb{B}$, $u : \mathbb{B}$, $u = v\vec{x}$, $t \triangleright_a v$, $\vec{x} \in X$ and $\mathbb{B}$ doesn't occur in $\tau(\vec{x})$

Unsafe occurrences of a sort $\mathbb{A}$ in a type T: $\mathrm{SPos}_{\mathbb{A}}(T)$

- $\mathrm{SPos}_{\mathbb{A}}(\mathbb{B}) = \mathrm{NPos}_{\mathbb{A}}(\mathbb{B}) = \mathrm{LPos}_{\mathbb{A}}(\mathbb{B}) = \emptyset$ whatever $\mathbb{A}$ and $\mathbb{B}$ are
- $\mathrm{CPos}_{\mathbb{A}}(\mathbb{A}) = \{\varepsilon\}$
- $\mathrm{CPos}_{\mathbb{A}}(\mathbb{B}) = \emptyset$ if $\mathbb{B} \neq \mathbb{A}$

- $\mathrm{SPos}_{\mathbb{A}}(U \to V) = 1 \cdot \mathrm{NPos}_{\mathbb{A}}(U) + 2 \cdot \mathrm{SPos}_{\mathbb{A}}(V)$
- $\mathrm{NPos}_{\mathbb{A}}(U \to V)$
  $= \mathrm{CPos}_{\mathbb{A}}(U \to V) = 1 \cdot \mathrm{SPos}_{\mathbb{A}}(U) + 2 \cdot (\mathrm{LPos}_{\mathbb{A}}(V) + \mathrm{CPos}_{\mathbb{A}}(V))$
- $\mathrm{LPos}_{\mathbb{A}}(U \to V) = \mathrm{NPos}_{\mathbb{A}}(U \Rightarrow V) + 1 \cdot \mathrm{NPos}_{\mathbb{A}}(U)$

Unsafe occurrences of a sort $\mathbb{A}$ in a type T: $\mathrm{SPos}_{\mathbb{A}}(T)$

Examples of safe types T, i.e. with $\mathrm{SPos}_{\mathbb{A}}(T) = \emptyset$:

- $o(T) \leq 1$
- $T = \vec{U} \Rightarrow \mathbb{A}$ and $\mathbb{A}$ doesn't occur in $\vec{U}$ (e.g. Coq types)
- $o(T) = 2$ and $\mathbb{A}$ occurs only positively in T

Example of unsafe type: $(\mathbb{B} \Rightarrow (\mathbb{B} \Rightarrow \mathbb{A}) \Rightarrow \mathbb{B}) \Rightarrow \mathbb{A}$