HORSes: format, termination and confluence

Jean-Pierre Jouannaud
INRIA-LIAMA and Tsinghua Software Chair

Joint on-going work with
Jianqi Li
School of Software, Tsinghua University

Project CoqLF
TNList Cross-discipline Foundation 2011-9

September 28, 2011

Outline

Our interest and non-interest in lambda-calculus

- Question: is a given term SN ?
  Answer: our interest is in proving SN theorems, NOT in analyzing the control flow of a particular term to prove it is SN.
- Question: does finite development hold ?
  Answer: can be seen as a control flow pb for a restricted form of beta-reduction. I can also turn it into the SN property of all terms of a modified system. YES, I am interested.
- Question: show that simply typed lambda calculus is SN.
  Answer: YES, of course.

And, I want formal proofs and machine support.

Objectives of CoqLF

To equip Coq with libraries allowing users to specify logical systems (or programming languages) via HOAS approach and carry out meta-theoretical studies with automated tools for:

- defining typing systems declaratively as a term language, a type language, a set of typing rules and a set of computational rules (Beluga, FreshML, Cαml, UNBOUND, etc.)
- showing burocratic lemmas,
- showing type preservation,
- showing confluence,
- showing strong normalization.

This talk

In this talk, we concentrate on the rewrite rules:

- format
- confluence
- termination

First-order rules on first-order syntax and higher-order rules on higher-order syntax should be two different instances of a same mechanism, so as to have generic tools.

Nipkow's HOR

All symbols are higher-order constants. We use Krivine's style for λ-expressions on *this* slide.

$$\begin{array}{lcl} \times & : & R \to R \to R \\ \mathrm{diff} & : & (R \to R) \to R \to R \\ \sin, \cos & : & R \to R \\ F & : & R \to R \end{array}$$

Rewrite rule for differentiation:

$$\mathrm{diff}\,(\lambda x.\sin\,(F\,x))(y) \to \cos\,(F\,y) \times \mathrm{diff}\,(F)(y)$$

of which $\mathrm{diff}(\lambda x.\sin(x))(y) \to \cos(y)$ is an instance (replace F by $\lambda x.x$ and normalize).

Note: β is used both as an equation and rule.

Nipkow's HOR

All symbols are higher-order constants. We use Krivine's style for λ-expressions on *this* slide.

$$\begin{array}{lcl} \times & : & R \to R \to R \\ \mathrm{diff} & : & (R \to R) \to R \to R \\ \sin, \cos & : & R \to R \\ F & : & R \to R \end{array}$$

Rewrite rule for differentiation:

$$\mathrm{diff}\,(\lambda x.\sin\,(F\,x))(y) \to \cos\,(F\,y) \times \mathrm{diff}\,(F)(y)$$

of which $\mathrm{diff}(\lambda x.\sin(x))(y) \to \cos(y)$ is an instance (replace F by $\lambda x.x$ and normalize).

Note: β is used both as an equation and rule.

Nipkow's differentiation revisited

$$\begin{array}{lcl} \mathrm{diff} & : & (R \to R) \Rightarrow (R \to R) \\ \sin, \cos & : & R \Rightarrow R \\ F & : & \Rightarrow R \to R \\ \times & : & (R \to R) \to (R \to R) \Rightarrow (R \to R) \end{array}$$

$$\mathrm{diff}(\lambda x.\, \sin(F\ x)) \to \lambda x.\, \cos(F\ x) \times \mathrm{diff}(F)$$

Higher-order matching is still necessary.
Confluence is harder to prove.

Algebraic differentiation: choose your types!

$$
\begin{array}{lcl}
\mathrm{diff} & : & (R \to R) \Rightarrow (R \to R) \\
\sin, \cos & : & \Rightarrow R \to R \\
F & : & \Rightarrow R \to R \\
\circ, \times & : & (R \to R) \to (R \to R) \Rightarrow (R \to R)
\end{array}
$$

A rewrite rule for differentiation:

$$\mathrm{diff}(sin \circ F) \to \cos \times \mathrm{diff}(F)$$

of which $\mathrm{diff}(sin) \to \cos$ is an instance (replace F by identity and normalize w.r.t. identity rules for composition and product).

Recursors on polymorphic finite lists

$$\begin{array}{lcl} \alpha & : & * \\ \mathsf{list} & : & * \to * \\ H & : & \alpha \\ F & : & \alpha \to \alpha \\ \mathsf{nil}, T & : & \mathsf{list}(\alpha) \\ \mathsf{cons} & : & \alpha \times \mathsf{list}(\alpha) \Rightarrow \mathsf{list}(\alpha) \\ \mathsf{map} & : & \mathsf{list}(\alpha) \times (\alpha \to \alpha) \Rightarrow \mathsf{list}(\alpha) \end{array}$$

$$\begin{array}{lcl} \mathsf{map}(\mathsf{nil}, F) & \to & \mathsf{nil} \\ \mathsf{map}(\mathsf{cons}(H, T), F) & \to & \mathsf{cons}((F\ H), \mathsf{map}(T, F)) \end{array}$$

(Plain) *first-order matching* suffices, because matching is on (free) *constructor expressions*.

Idempotent Abelian groups

$$\begin{array}{lcl} \mathrm{G} & : & * \\ + & : & G \rightarrow G \Rightarrow G \\ {}^{-1} & : & G \Rightarrow G \\ 0 & : & \Rightarrow G \end{array}$$

$$\begin{array}{rlcl} Inv: & x + x^{-1} & \rightarrow & 0 \\ Z: & x + 0 & \rightarrow / = & x \\ I: & x + x & \rightarrow / = & x \\ A: & (x+y)+z & = & x+(y+z) \\ C: & x+y & = & y+x \end{array}$$

Matching is modulo ACZI on terms in normal form: Z, I are used as both equations and rules.

Manifesto

- These examples share a common structure.
- The first/higher-order characteristic is *not* relevant.
- We need a rule format
  emphasizing the structure of computation,
  and a representation of terms
  hiding their syntactic differences.
- We like a rule format
  with good operational behaviour
  (pattern-based lhs !, safe rhs ?)

Term representation

There are two different styles of representations:

- canonical representations: locally nameless (De Bruijn numbers), or locally canonically named (Sato, Sato-Pollack)
- non-canonical representations with explicit α-conversion.

Canonical reps are superior for reasonning: renaming is built-in the induction principle.

Non-canonical are superior for computing: renaming is by need.

Our choice: both !

Rules, equations and simplifiers

We distinguish three kinds of sets:

- A set of rules R used for rewriting only: differentiation or Inv,
- A set of equations E used for matching only: α-conversion or AC,
- A set of simplifiers S used for normalization and matching:
  $\beta\eta$ or I.

Normal rewriting

$$u \longrightarrow^{p}_{R_{S_E\downarrow}} (u[r\sigma]_p)\downarrow_{S_E} \quad \text{if}$$

$u = u\downarrow_{S_E}$

$u|_p =_{S\cup E} l\sigma$ for some $l \to r \in R$

$$v \longrightarrow^{p}_{S_E} v[d\theta]_p \quad \text{if}$$

$v|_p =_E g\theta$ for some $g \to d \in S$

General assumptions for normal rewriting

(a) S is Church-Rosser modulo E,

(b) $R_{S\cup E} \cup S_E$ is terminating,

(c) rules in R are S_E-normalized,

(d) $R_{S_E\downarrow}$ is Church-Rosser modulo $S \cup E$.

$$u \longrightarrow^{p}_{R_{S_E\downarrow}} (u[r\sigma]_p)\downarrow_{S_E} \quad \text{if}$$

$$u = u\downarrow_{S_E}$$

$$u|_p =_{S\cup E} l\sigma \text{ for some } l \to r \in R$$

$$v \longrightarrow^{p}_{S_E} v[d\theta]_p \quad \text{if}$$

$$v|_p =_E g\theta \text{ for some } g \to d \in S$$

General assumptions for normal rewriting

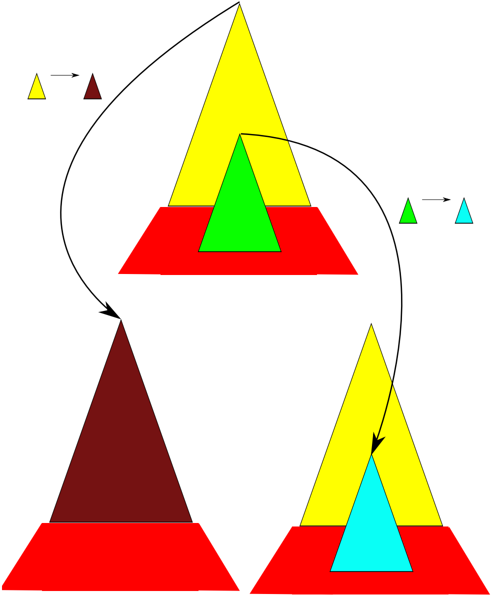
(a) S is Church-Rosser modulo E,

(b) $R_{S\cup E} \cup S_E$ is terminating,

(c) rules in R are S_E-normalized,

(d) $R_{S_E\downarrow}$ is Church-Rosser modulo $S \cup E$.

Confluence analysis: critical pairs



Coherence analysis: extension rules

Assume

$$\dfrac{\begin{array}{rrcll} \text{a rule} & l & \rightarrow & r & \\ \text{an equation} & g & = & d & \\ \text{such that} & (g|_p)\sigma & = & l\sigma & \text{for mgu } \sigma \end{array}}{\begin{array}{rrcl} \text{Extension rule} & g[l]_p & \rightarrow & g[r]_p \end{array}}$$

Assume $+$ is AC and consider rule $a + b \rightarrow r$.

$(a + b) + c$ is not in normal form

$a + (c + b)$ is in normal form

Both rewrite to $r + c$ with $a + b + x \rightarrow r + x$.

Coherence analysis: extension rules

Assume

$$\frac{\begin{array}{rrcll} \text{a rule} & l & \to & r & \\ \text{an equation} & g & = & d & \\ \text{such that} & (g|_p)\sigma & = & l\sigma & \text{for mgu } \sigma \end{array}}{\begin{array}{rrcl} \text{Extension rule} & g[l]_p & \to & g[r]_p \end{array}}$$

Assume $+$ is AC and consider rule $a + b \to r$.

$(a + b) + c$ is not in normal form

$a + (c + b)$ is in normal form

Both rewrite to $r + c$ with $a + b + x \to r + x$.

Checking (d)

Theorem

Let R, S, E satisfy:
properties (a), (b), and (c),
(S', S'') is a splitting of S, i.e., $S_E{\downarrow} = S'_E{\downarrow}\ S''_E{\downarrow}$.
Then, normal rewriting is Church-Rosser (d) if

(i) R is closed wrt normalized $E \cup S$-extensions
(ii) R is closed under forward pairs with S'',
(iii) $S'{\downarrow}$ shallow critical pairs in $SCP_E(R, S')$ are strongly E-joinable
(iv) $S'{\downarrow}$ critical pairs in $CP_{SE}(R)$ are E-joinable,

α-conversion: nothing to do.

η is used as a reduction: no forward pairs; for $app(l, x) \to r$ with $x \notin \mathcal{V}ar(l)$, add $l \to \lambda x.r$

η is used as an expansion: no extension; for $app(l, x) \to r$ with $x \notin \mathcal{V}ar(l)$, add $l \to \lambda x.r$

β is used as a reduction
when the rules in R are of base type, their lefthand side cannot unify with an abstraction.
non-base type case:
For each rule $\lambda x.l \to r$, add $l \to app(r, x)\downarrow$.

no shallow critical pairs:
Since rules are in β-normal form, no subterm of a rule can unify with a β-redex.

Nipkow's rewriting and variations

α-conversion: nothing to do.

η is used as a reduction: no forward pairs;
for $app(l, x) \to r$ with $x \notin \mathcal{V}ar(l)$, add $l \to \lambda x.r$

η is used as an expansion: no extension;
for $app(l, x) \to r$ with $x \notin \mathcal{V}ar(l)$, add $l \to \lambda x.r$

β is used as a reduction
when the rules in R are of base type, their lefthand side cannot unify with an abstraction.
non-base type case:
For each rule $\lambda x.l \to r$, add $l \to app(r, x) \downarrow$.

no shallow critical pairs:
Since rules are in β-normal form, no subterm of a rule can unify with a β-redex.

Nipkow's rewriting and variations

α-conversion: nothing to do.

η is used as a reduction: no forward pairs;
for $app(l, x) \to r$ with $x \notin \mathcal{V}ar(l)$, add $l \to \lambda x.r$

η is used as an expansion: no extension;
for $app(l, x) \to r$ with $x \notin \mathcal{V}ar(l)$, add $l \to \lambda x.r$

β is used as a reduction
when the rules in R are of base type, their lefthand side cannot unify with an abstraction.
non-base type case:
For each rule $\lambda x.l \to r$, add $l \to app(r, x) \downarrow$.

no shallow critical pairs:
Since rules are in β-normal form, no subterm of a rule can unify with a β-redex.

α-conversion: nothing to do.

η is used as a reduction: no forward pairs;
for $app(l, x) \to r$ with $x \notin \mathcal{V}ar(l)$, add $l \to \lambda x.r$

η is used as an expansion: no extension;
for $app(l, x) \to r$ with $x \notin \mathcal{V}ar(l)$, add $l \to \lambda x.r$

β is used as a reduction
when the rules in R are of base type, their lefthand side cannot unify with an abstraction.
non-base type case:
For each rule $\lambda x.l \to r$, add $l \to app(r, x) \downarrow$.

no shallow critical pairs:
Since rules are in β-normal form, no subterm of a rule can unify with a β-redex.

Rule format

Minimize the amount of computations:

Rules in R of the form:

$F(\bar{l}) \to r$

where F is a graded higher-order constant.

Make rewriting and unification feasible:

$F(\bar{l})$ is a pattern in the sense of Miller.

Type of rules is arbitrary.

Termination proofs

- Requirement: powerful but easy to use.
- Challenge: a painless version of Girard's "reducibility candidates"
- Approach:
  1. define well-founded orderings on the abstract syntax of terms and types ;
  2. use Girard's method to prove their well-foundedness ;
  3. incorporate semantic termination arguments to strengthen these orderings.

CPO: $s \succ^X t$ iff

Case 1: $s = f(\overline{s})$ with $f \in \mathcal{FS}$ and $t \in X$ or

1. $u : \theta \succeq_{\mathcal{T_S}} t : \tau$ for some u such that $u : \theta \in \overline{s}$
2. $t = g(\overline{t})$ with $f >_{\mathcal{F}} g \in \mathcal{FS} \cup \{@\}$ and $s \succ^X \overline{t}$
3. $t = g(\overline{t}),\ f =_{\mathcal{F}} g \in \mathcal{F}$ and $s \succ^X \overline{t}, \overline{s}(\succ_{\mathcal{T_S}})_{stat_f} \overline{t}$
4. $t = \lambda x.u$ with $x \notin X$ and $f(\overline{s}) \succ^{X \cup \{x\}} u$

Case 2: $s = @(v, w)$ and

1. $t = @(u, r)$ and $(v, w)(\succ^X_{\mathcal{T_S}})_{mul}(u, r)$
2. $v = \lambda x.u$ and $u\{x \mapsto w\} \succ^X t$

Case 3: $s = \lambda x : \alpha.u$ and

1. $t = \lambda x : \beta.v,\ x \notin X,\ \alpha \simeq \beta$ and $u \succ^{X \cup \{x\}} v$
2. $u = @(v, x),\ x \notin \mathcal{V}ar(v)$ and $v \succ^X t$

Case 4: $s = u \rightarrow v$ and $v \succeq t$ or $t = u' \rightarrow v'$ and $\{u, v\} \succ_{lex} \{u', v'\}$

Case 5: $s = *$ and $t = *$

Case 1: $s = f(\overline{s})$ with $f \in \mathcal{FS}$ and $t \in X$ or

1. $u : \theta \succeq_{\mathcal{T_S}} t : \tau$ for some u such that $u : \theta \in \overline{s}$
2. $t = g(\overline{t})$ with $f >_{\mathcal{F}} g \in \mathcal{FS} \cup \{@\}$ and $s \succ^X \overline{t}$
3. $t = g(\overline{t}),\ f =_{\mathcal{F}} g \in \mathcal{F}$ and $s \succ^X \overline{t}, \overline{s}(\succ_{\mathcal{T_S}})_{stat_f} \overline{t}$
4. $t = \lambda x.u$ with $x \notin X$ and $f(\overline{s}) \succ^{X \cup \{x\}} u$

Case 2: $s = @(v, w)$ and

1. $t = @(u, r)$ and $(v, w)(\succ^X_{\mathcal{T_S}})_{mul}(u, r)$
2. $v = \lambda x.u$ and $u\{x \mapsto w\} \succ^X t$

Case 3: $s = \lambda x : \alpha.u$ and

1. $t = \lambda x : \beta.v,\ x \notin X,\ \alpha \simeq \beta$ and $u \succ^{X \cup \{x\}} v$
2. $u = @(v, x),\ x \notin \mathcal{V}ar(v)$ and $v \succ^X t$

Case 4: $s = u \to v$ and $v \succeq t$ or $t = u' \to v'$ and $\{u, v\} \succ_{lex} \{u', v'\}$

Case 5: $s = *$ and $t = *$

CPO: $s \succ^X t$ iff

Case 1: $s = f(\overline{s})$ with $f \in \mathcal{FS}$ and $t \in X$ or

1. $u : \theta \succeq_{\mathcal{T_S}} t : \tau$ for some u such that $u : \theta \in \overline{s}$
2. $t = g(\overline{t})$ with $f >_{\mathcal{F}} g \in \mathcal{FS} \cup \{@\}$ and $s \succ^X \overline{t}$
3. $t = g(\overline{t})$, $f =_{\mathcal{F}} g \in \mathcal{F}$ and $s \succ^X \overline{t}$, $\overline{s}(\succ_{\mathcal{T_S}})_{stat_f}\overline{t}$
4. $t = \lambda x.u$ with $x \notin X$ and $f(\overline{s}) \succ^{X \cup \{x\}} u$

Case 2: $s = @(v, w)$ and

1. $t = @(u, r)$ and $(v, w)(\succ^X_{\mathcal{T_S}})_{mul}(u, r)$
2. $v = \lambda x.u$ and $u\{x \mapsto w\} \succ^X t$

Case 3: $s = \lambda x : \alpha.u$ and

1. $t = \lambda x : \beta.v$, $x \notin X$, $\alpha \simeq \beta$ and $u \succ^{X \cup \{x\}} v$
2. $u = @(v, x)$, $x \notin \mathcal{V}ar(v)$ and $v \succ^X t$

Case 4: $s = u \to v$ and $v \succeq t$ or $t = u' \to v'$ and $\{u, v\} \succ_{lex} \{u', v'\}$

Case 5: $s = *$ and $t = *$

CPO: $s \succ^X t$ iff

Case 1: $s = f(\overline{s})$ with $f \in \mathcal{FS}$ and $t \in X$ or

1. $u : \theta \succeq_{\mathcal{T_S}} t : \tau$ for some u such that $u : \theta \in \overline{s}$
2. $t = g(\overline{t})$ with $f >_{\mathcal{F}} g \in \mathcal{FS} \cup \{@\}$ and $s \succ^X \overline{t}$
3. $t = g(\overline{t})$, $f =_{\mathcal{F}} g \in \mathcal{F}$ and $s \succ^X \overline{t}$, $\overline{s}(\succ_{\mathcal{T_S}})_{stat_f} \overline{t}$
4. $t = \lambda x.u$ with $x \notin X$ and $f(\overline{s}) \succ^{X \cup \{x\}} u$

Case 2: $s = @(v, w)$ and

1. $t = @(u, r)$ and $(v, w)(\succ^X_{\mathcal{T_S}})_{mul}(u, r)$
2. $v = \lambda x.u$ and $u\{x \mapsto w\} \succ^X t$

Case 3: $s = \lambda x : \alpha.u$ and

1. $t = \lambda x : \beta.v$, $x \notin X$, $\alpha \simeq \beta$ and $u \succ^{X \cup \{x\}} v$
2. $u = @(v, x)$, $x \notin \mathcal{V}ar(v)$ and $v \succ^X t$

Case 4: $s = u \to v$ and $v \succeq t$ or $t = u' \to v'$ and $\{u, v\} \succ_{lex} \{u', v'\}$

Case 5: $s = *$ and $t = *$

CPO: $s \succ^X t$ iff

Case 1: $s = f(\overline{s})$ with $f \in \mathcal{FS}$ and $t \in X$ or

1. $u : \theta \succeq_{\mathcal{T}_{\mathcal{S}}} t : \tau$ for some u such that $u : \theta \in \overline{s}$
2. $t = g(\overline{t})$ with $f >_{\mathcal{F}} g \in \mathcal{FS} \cup \{@\}$ and $s \succ^X \overline{t}$
3. $t = g(\overline{t}),\ f =_{\mathcal{F}} g \in \mathcal{F}$ and $s \succ^X \overline{t}, \overline{s}(\succ_{\mathcal{T}_{\mathcal{S}}})_{stat_f} \overline{t}$
4. $t = \lambda x.u$ with $x \notin X$ and $f(\overline{s}) \succ^{X \cup \{x\}} u$

Case 2: $s = @(v, w)$ and

1. $t = @(u, r)$ and $(v, w)(\succ^X_{\mathcal{T}_{\mathcal{S}}})_{mul}(u, r)$
2. $v = \lambda x.u$ and $u\{x \mapsto w\} \succ^X t$

Case 3: $s = \lambda x : \alpha.u$ and

1. $t = \lambda x : \beta.v,\ x \notin X,\ \alpha \simeq \beta$ and $u \succ^{X \cup \{x\}} v$
2. $u = @(v, x),\ x \notin \mathcal{V}ar(v)$ and $v \succ^X t$

Case 4: $s = u \to v$ and $v \succeq t$ or $t = u' \to v'$ and $\{u, v\} \succ_{lex} \{u', v'\}$

Case 5: $s = *$ and $t = *$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H, T), F) \to \mathrm{cons}((F\ H), \mathrm{map}(T, F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H), \mathrm{map}(T, F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ @(F, H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H, T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ \mathrm{map}(T, F)$
Subgoal 21:	$\{\mathrm{cons}(H, T), F\}(\succ_{\mathcal{T}_S})_{mul}\{T, F\}$
Subgoal 211:	$\mathrm{cons}(H, T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$ $\square$

$$\mathrm{map}(\mathrm{cons}(H, T), F) \to \mathrm{cons}((F\ H), \mathrm{map}(T, F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H), \mathrm{map}(T, F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ @(F, H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H, T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ \mathrm{map}(T, F)$
Subgoal 21:	$\{\mathrm{cons}(H, T), F\}(\succ_{\mathcal{T}_S})_{mul}\{T, F\}$
Subgoal 211:	$\mathrm{cons}(H, T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

$\square$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \to \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

$$\mathrm{map}(\mathrm{cons}(H, T), F) \to \mathrm{cons}((F\ H), \mathrm{map}(T, F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H), \mathrm{map}(T, F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ @(F, H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H, T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H, T), F) \succ \mathrm{map}(T, F)$
Subgoal 21:	$\{\mathrm{cons}(H, T), F\}(\succ_{\mathcal{T}_S})_{mul}\{T, F\}$
Subgoal 211:	$\mathrm{cons}(H, T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

□

recursor on lists

$$map(cons(H,T),F) \rightarrow cons((F\ H), map(T,F))$$

Goal:	$map(cons(H,T),F) \succ_{\mathcal{T}_S} cons((F\ H), map(T,F))$
Subgoal 1:	$map(cons(H,T),F) \succ @(F,H)$
Subgoal 11:	$map(cons(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$map(cons(H,T),F) \succeq H$
Subgoal 121:	$cons(H,T) : list(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$list(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$map(cons(H,T),F) \succ map(T,F)$
Subgoal 21:	$\{cons(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$cons(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

$\square$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \rightarrow \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{\mathit{mul}}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

$$\mathsf{map}(\mathsf{cons}(H,T),F) \to \mathsf{cons}((F\ H),\mathsf{map}(T,F))$$

Goal:	$\mathsf{map}(\mathsf{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathsf{cons}((F\ H),\mathsf{map}(T,F))$
Subgoal 1:	$\mathsf{map}(\mathsf{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathsf{map}(\mathsf{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathsf{map}(\mathsf{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathsf{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathsf{map}(\mathsf{cons}(H,T),F) \succ \mathsf{map}(T,F)$
Subgoal 21:	$\{\mathsf{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathsf{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \rightarrow \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

recursor on lists

$$\text{map}(\text{cons}(H,T),F) \to \text{cons}((F\ H),\text{map}(T,F))$$

Goal:	$\text{map}(\text{cons}(H,T),F) \succ_{\mathcal{T}_S} \text{cons}((F\ H),\text{map}(T,F))$
Subgoal 1:	$\text{map}(\text{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\text{map}(\text{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\text{map}(\text{cons}(H,T),F) \succeq H$
Subgoal 121:	$\text{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\text{map}(\text{cons}(H,T),F) \succ \text{map}(T,F)$
Subgoal 21:	$\{\text{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\text{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

$\square$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \to \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \to \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \rightarrow \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \to \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

$\square$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \to \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

recursor on lists

$$\mathrm{map}(\mathrm{cons}(H,T),F) \to \mathrm{cons}((F\ H),\mathrm{map}(T,F))$$

Goal:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ_{\mathcal{T}_S} \mathrm{cons}((F\ H),\mathrm{map}(T,F))$
Subgoal 1:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ @(F,H)$
Subgoal 11:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ F$
Subgoal 111:	$F \succeq_{\mathcal{T}_S} F$
Subgoal 12:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succeq H$
Subgoal 121:	$\mathrm{cons}(H,T) : \mathit{list}(\alpha) \succeq_{\mathcal{T}_S} H : \alpha$
Subgoal 1211:	$\mathit{list}(\alpha) : * \succeq_{\mathcal{T}_S} \alpha : *$
Subgoal 12111:	$* \succeq *$
Subgoal 12112:	$\alpha \succeq_{\mathcal{T}_S} \alpha$
Subgoal 1212:	$H \succeq_{\mathcal{T}_S} H$
Subgoal 2:	$\mathrm{map}(\mathrm{cons}(H,T),F) \succ \mathrm{map}(T,F)$
Subgoal 21:	$\{\mathrm{cons}(H,T),F\}(\succ_{\mathcal{T}_S})_{mul}\{T,F\}$
Subgoal 211:	$\mathrm{cons}(H,T) \succ_{\mathcal{T}_S} T$
Subgoal 211:	$T \succ_{\mathcal{T}_S} T$

$\square$

Size changing principle

Here is how we prove Neil's (first-order) example (RPO would be enough here):

$$
\begin{array}{rcl}
f(o, y) & \to & y \\
f(Sx, y) & \to & g(y, y, 0) \\
g(su, v, 0) & \to & f(u, v) \\
g(u, Sv, Sx) & \to & g(u, v, s^3(x))
\end{array}
$$

use RPO with

$f \equiv g > S > 0$

f, g lexicographic

Neil's higher-order example can be proved as well, with CPO this time.

Further problems

- Implementation
- Confluence result satisfactory but need for experiments
- Order is the weak piece:
  currently restricted to ML-like polymorphism.
  CPO can be defined for true polymorphic types, but no proof of well-foundedness (yet).
  dependent types: same.
  semantic information (not hard)