Termination of λΠ-Calculus Modulo Rewriting using Size-Change Principle

Work in progress

Guillaume Genestier
Joint work with Frédéric Blanqui

Thursday, July 19th 2018

Contents

Typing rule for application :

$$\frac{\Gamma \vdash t : \forall(x : A).B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\,u : B\left[{}^{u}/_{x}\right]}$$

```
ℕ : TYPE
0 : ℕ
s : ℕ ⇒ ℕ

A : TYPE

List : ℕ ⇒ TYPE
nil  : List 0
cons : ∀(n:ℕ).A ⇒ List n ⇒ List (s n)
```

Rewrite rules

Conversion :

$$\frac{\Gamma \vdash t : A \qquad A \longleftrightarrow^* B \qquad \Gamma \vdash B : \mathtt{TYPE}}{\Gamma \vdash t : B}$$

```
ℕ : TYPE
0 : ℕ
s : ℕ ⇒ ℕ

+ : ℕ ⇒ ℕ ⇒ ℕ

      0 + n ⟶ n
  (s m) + n ⟶ s (m + n)
(m + n) + p ⟶ m + (n + p)
```

λΠ-calculus modulo rewriting

Higher-order rewrite rules

```
map: (A ⇒ A) ⇒ ∀(n:ℕ). List n ⇒ List n

map f 0 nil             ⟶ nil
map f _ (cons n x l)    ⟶ cons n (f x) (map f n l)
```

Type-level rewrite rules

```
NArrows : ℕ ⇒ TYPE

NArrows 0       ⟶ ℕ
NArrows (s n)   ⟶ ℕ ⇒ (NArrows n)
```

Our goal

We want to prove strong normalization of $\longrightarrow_\beta \cup \longrightarrow_\mathcal{R}$ for *typable* terms in the $\lambda\Pi$-calculus modulo $\longleftrightarrow^*_{\beta\mathcal{R}}$

Contents

Wahlstedt's criterion, 2007

For $\mathcal{R}$ orthogonal

1) No infinite DP-chains $\Rightarrow$ Weak normalization of typable terms,
2) Size-Change Termination $\Rightarrow$ No infinite DP-chains.

Our contributions

For $\mathcal{R}$ ~~orthogonal~~ confluent

1) No infinite DP-chains $\Rightarrow$ ~~Weak~~ Strong normalization of typable terms,
2) Size-Change Termination $\Rightarrow$ No infinite DP-chains.

Definition (Dependency pair)

$(f\ \bar{p}, g\ \bar{u})$ *is a dependency pair if* $f\ \bar{p} \longrightarrow C[g\ \bar{u}]$ *is a rewrite rule.*

Definition (Instanciated call)

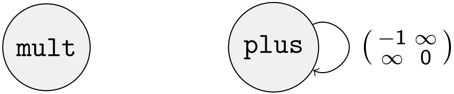
$f\ \bar{t} \tilde{>} g\ \bar{v}$ *if:*

- $(f\ \bar{p}, g\ \bar{u})$ *is a dependency pair,*
- $\bar{t} \longrightarrow^* \bar{p}\sigma$,
- $\bar{u}\sigma = \bar{v}$.

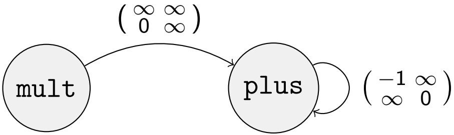
$\tilde{>}$ is well-founded = no infinite DP-chains

Size-Change Termination [Lee, Jones, Ben Amram, 2001]

(s m) + n ⟶ s (m + n)	
(s m) × n ⟶ n + (m × n)	

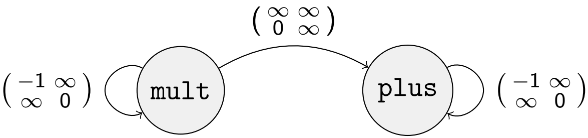


`(s m) + n` $\longrightarrow$ `s (m + n)`	$\begin{array}{c} \\ s\,m \\ n \end{array} \begin{array}{c} \begin{array}{cc} m & n \end{array} \\ \begin{pmatrix} -1 & \infty \\ \infty & 0 \end{pmatrix} \end{array}$
`(s m)` $\times$ `n` $\longrightarrow$ `n + (m` $\times$ `n)`	

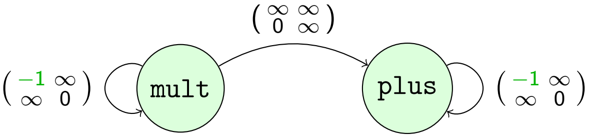


`(s m) + n` $\longrightarrow$ `s (m + n)`	$\begin{array}{cc} & \begin{array}{cc} m & n \end{array} \\ \begin{array}{c} s\,m \\ n \end{array} & \begin{pmatrix} -1 & \infty \\ \infty & 0 \end{pmatrix} \end{array}$
`(s m)` $\times$ `n` $\longrightarrow$ `n + (m` $\times$ `n)`	$\begin{array}{cc} & \begin{array}{cc} n & m \times n \end{array} \\ \begin{array}{c} s\,m \\ n \end{array} & \begin{pmatrix} \infty & \infty \\ 0 & \infty \end{pmatrix} \end{array}$

Size-Change Termination [Lee, Jones, Ben Amram, 2001]



`(s m) + n` $\longrightarrow$ `s (m + n)`	$\begin{array}{c} \\ s\,m \\ n \end{array}\begin{array}{c} \begin{array}{cc} m & n \end{array} \\ \begin{pmatrix} -1 & \infty \\ \infty & 0 \end{pmatrix} \end{array}$
`(s m)` $\times$ `n` $\longrightarrow$ `n + (m` $\times$ `n)`	$\begin{array}{c} \\ s\,m \\ n \end{array}\begin{array}{c} \begin{array}{cc} n & m \times n \end{array} \\ \begin{pmatrix} \infty & \infty \\ 0 & \infty \end{pmatrix} \end{array}$ $\begin{array}{c} \\ s\,m \\ n \end{array}\begin{array}{c} \begin{array}{cc} m & n \end{array} \\ \begin{pmatrix} -1 & \infty \\ \infty & 0 \end{pmatrix} \end{array}$



`(s m) + n` $\longrightarrow$ `s (m + n)`	$\begin{array}{c c} & \begin{array}{cc} m & n \end{array} \\ \begin{array}{c} s\,m \\ n \end{array} & \begin{pmatrix} -1 & \infty \\ \infty & 0 \end{pmatrix} \end{array}$
`(s m) × n` $\longrightarrow$ `n + (m × n)`	$\begin{array}{c c} & \begin{array}{cc} n & m \times n \end{array} \\ \begin{array}{c} s\,m \\ n \end{array} & \begin{pmatrix} \infty & \infty \\ 0 & \infty \end{pmatrix} \end{array}$ $\begin{array}{c c} & \begin{array}{cc} m & n \end{array} \\ \begin{array}{c} s\,m \\ n \end{array} & \begin{pmatrix} -1 & \infty \\ \infty & 0 \end{pmatrix} \end{array}$

Orthogonality is a restriction

Associativity and distributivity

$$(x + y) + z \longrightarrow x + (y + z)$$
$$(x + y) \times z \longrightarrow (x \times z) + (y \times z)$$

Signed integers

$\mathbb{Z}$: TYPE

$0 : \mathbb{Z}$
$s : \mathbb{Z} \Rightarrow \mathbb{Z}$
$p : \mathbb{Z} \Rightarrow \mathbb{Z}$

$s\ (p\ x) \longrightarrow x$
$p\ (s\ x) \longrightarrow x$

Contents

Computability closure

Given a well-founded precedence $\preceq$.

Blanqui, 2005

Typable terms are SN, if for all $f\,\bar{l} \longrightarrow r \in \mathcal{R}$, $r \in CC(f\,\bar{l})$

$CC(f\,\bar{l})$ is the set of typable terms, where function application is restricted as follows:

$$\frac{g \prec f \vee (g \simeq f \wedge \bar{m} <_{mul} \bar{l}) \qquad \bar{m} \in CC(f\,\bar{l})}{g\,\bar{m} \in CC(f\,\bar{l})}$$

Property

$CC(f\,\bar{l}) \subseteq \mathrm{SN}$ *whenever* $\bar{l}$ *are strongly normalizing.*

It requires a strict decrease at each call.

Mutually recursive

```
filter : (A ⇒ Bool) ⇒ List ⇒ List

filter f (cons x l) ⟶ bis (f x) f x l

bis true  f x l ⟶ cons x (filter f l)
bis false f x l ⟶ filter f l
```

Contents

Unifying both

Theorem

$\longrightarrow_\beta \cup \longrightarrow_\mathcal{R}$ *is SN on typable term modulo* $\longleftrightarrow^*_{\beta\mathcal{R}}$ *if:*

- $\longrightarrow_\beta \cup \longrightarrow_\mathcal{R}$ *is confluent (if there are type-level rules only),*
- $\longrightarrow_\beta \cup \longrightarrow_\mathcal{R}$ *preserves typing,*
- $\mathcal{R}$ *satisfies SCT,*
- *Plain Function Passing,*
- *the right-hand side of every rewrite rule is typable.*

Proof sketch

The proof relies on an adaptation of Girard's reducibility candidates to rewriting [Girard, 1972] [Jouannaud, Okada, 1997] :
We interpret every type T by a set of terms $I(T)$ satisfying Girard's conditions.

Definition

- *A signature is valid if for all* $f : T$, *we have* $f \in I(T)$.
- $t : T$ *is computable if* $t \in I(T)$.

Steps

1. $\mathcal{R}$ respects SCT $\Rightarrow$ no infinite chains,
2. No infinite chains $\Rightarrow$ signature is valid,
3. Signature is valid $\Rightarrow$ typable terms are computable.

Defining an interpretation for types

$\mathcal{R}_f(\mathbb{T}, \mathcal{P}(\mathbb{T}))$ set of partial functions from terms to sets of terms.
Interpretation is defined by the least fixpoint of $\mathcal{F}$ on the strictly inductive poset $\mathcal{R}_f(\mathbb{T}, \mathcal{P}(\mathbb{T}))$.

$$\mathcal{F} : \mathcal{R}_f(\mathbb{T}, \mathcal{P}(\mathbb{T})) \to \mathcal{R}_f(\mathbb{T}, \mathcal{P}(\mathbb{T}))$$

- $\mathcal{F}$ is increasing,
- For all I, $\mathrm{dom}(\mathcal{F}(I))$ is a reducibility candidate,
- For all I, T, $\mathcal{F}(I)(T)$ is a reducibility candidate.

Requirement

If $T \longrightarrow^* \forall(x : A).B$ then
$I(T) = \left\{ t \middle| \text{ for all } a \in I(A), t\,a \in I(B\left[{}^{a}/_{x}\right]) \right\}$

Example

```
p (s x) ⟶ x
s (p x) ⟶ x

f : ℤ ⇒ ℤ
g : ℤ ⇒ ℤ

f x ⟶ g x

g 0     ⟶ 0
g (s x) ⟶ f x
g (p x) ⟶ f x
```

Implementation

Implemented in the type-checker *Dedukti*.

Promising results

On 2017 Termination Problem Data Base, in the Higher-Order category, 75 files are proved terminating, including

- recursor of Gödel's system T,
- filter on lists.

https://github.com/Deducteam/Dedukti/tree/sizechange

Future work

Change order used in SCT [Thiemann, Giesl, 2005] [Coquand, 1992]

```
Ord : TYPE

  O : Ord
  S : Ord ⇒ Ord
lim : (ℕ ⇒ Ord) ⇒ Ord

ordrec : A⇒(Ord⇒A⇒A)⇒((ℕ⇒Ord)⇒(ℕ⇒A)⇒A)⇒Ord⇒A

ordrec x y z (lim f) ⟶ z f (λn.ordrec x y z (f n))
```

Future work

Higher-order matching

```
D (λx.sin (f x)) ⟶ mult (D (λx.f x)) (λx.cos (f x))
```

Local growth [Hyvernat, 2013]

```
f x           ⟶ g (s x)
g (s 0)       ⟶ 0
g (s (s x))   ⟶ f x
```