Program termination in the simply-typed λ-calculus

Frédéric Blanqui

INRIA

2nd Asian-Pacific Summer School on Formal Methods
Tsinghua University, 20-27 August 2010

Outline

Untyped λ-calculus

introduced by Alonzo Church in 1932

λ-terms: $t \in \mathcal{L} = x \in \mathcal{X} \mid \lambda x.t \mid tt$

β-reduction

$\rightarrow_\beta$ is defined by induction as follows:

$$\text{(top)}\ \frac{}{(\lambda x.t)u \rightarrow_\beta t_x^u}$$

context rules:

$$\text{(abs)}\ \frac{t \rightarrow_\beta t'}{\lambda x.t \rightarrow_\beta \lambda x.t'}$$

$$\text{(app1)}\ \frac{t \rightarrow_\beta t'}{tu \rightarrow_\beta t'u}$$

$$\text{(app2)}\ \frac{u \rightarrow_\beta u'}{tu \rightarrow_\beta tu'}$$

Termination

a term t terminates
if every sequence of β-reductions starting from t is finite

i.e. there is no infinite sequence of β-reductions starting from t

$$t = t_0 \rightarrow_\beta t_1 \rightarrow_\beta t_2 \rightarrow_\beta \ldots$$

let SN be the set of terminating terms

Which λ-terms terminate ?

- $(\lambda x.\mathtt{if}\ x \geq 2\ \mathtt{then}\ t\ \mathtt{else}\ u)v$ is typable and terminates
- $(\lambda x.xx)(\lambda x.xx)$ is not typable and does not terminate
- $(\lambda x.\mathtt{if}\ x \geq 2\ \mathtt{then}\ t\ \mathtt{else}\ u)$ "$\mathtt{foo}$" is not typable and terminates

Do all simply-typed λ-terms terminate ?

Simply-typed λ-calculus *à la* Church simplified

simple types: $S \in \mathcal{T} = B \in \mathcal{B} \mid S \to S$

To remove the need for typing environments, we assume that each variable x is given a fixed type τ_x. Let τ_t be the unique type of t:

$$\text{(var)}\ \frac{}{x : \tau_x} \quad \text{(abs)}\ \frac{t : T}{\lambda x.t : \tau_x \to T} \quad \text{(app)}\ \frac{u : S \to T \quad s : S}{us : T}$$

Consequence: induction on $v : T$ is equivalent to induction on v

Preservation of typing under substitution

Definition: a substitution ρ is well-typed if, for all x, $x\rho : \tau_x$.

Lemma: if $v : V$ and ρ is well-typed, then $v\rho : V$.

Proof. By induction on v. Exercise.

Remark: In the following, we only consider well-typed terms and substitutions.

Preservation of typing under reduction

Lemma: if $v : V$ and $v \rightarrow_\beta v'$, then $v' : V$.

Proof. By induction on v. Exercise.

Outline

β-reduction

Van Daalen's proof (1980)

Tait's proof (1967)

First proof attempt

Theorem: if $v : V$, then $v \in \mathrm{SN}$.

Proof. By induction on v.

- $v \in \mathcal{X}$. Then, $v \in \mathrm{SN}$.
- $v = \lambda x.t$. By IH, $t \in \mathrm{SN}$. Thus, $v \in \mathrm{SN}$.

Application case

- $v = us$. By IH, $u, s \in \mathrm{SN}$. How to prove that $v = us \in \mathrm{SN}$?

  Remark: $v \in \mathrm{SN}$ if every reduct of v is SN.

  Possible reducts of us:

  - $u's$ with u' a reduct of u
  - us' with s' a reduct of s
  - t_x^s if $u = \lambda x.t$

  Is every possible reduct SN ?

  Since $u, s \in \mathrm{SN}$, the first two cases can be dealt with by well-founded induction on (u, s).

  For the third case, we need to strengthen the IH.

Second proof attempt

Theorem: if $v : V$ and $a : \tau_y$ are SN, then $v_y^a \in \mathrm{SN}$.

Proof. Let $\rho = \binom{a}{y}$. By induction on v.

- $v \in \mathcal{X}$. If $v = y$, then $v\rho = a \in \mathrm{SN}$. Otherwise, $v\rho = v \in \mathrm{SN}$.
- $v = \lambda x.t$. Wlog we can assume that $x \neq y$ and $x \notin \mathrm{FV}(a)$. Thus, $v\rho = \lambda x.t\rho$. By IH, $t\rho \in \mathrm{SN}$. Thus, $v\rho \in \mathrm{SN}$.

Application case: $v = uss_1 \dots s_n$

- $u = x \neq y$. Then, $v\rho = xs\rho s_1\rho \dots s_n\rho \in \mathrm{SN}$ since, by IH, $s\rho, s_1\rho, \dots, s_n\rho \in \mathrm{SN}$.
- $u = y$. Then, $v\rho = as\rho s_1\rho \dots s_n\rho$. If $v\rho \notin \mathrm{SN}$, then $a \rightarrow_\beta^* \lambda x.b$ and $b_x^{s\rho} s_1\rho \dots s_n\rho \notin \mathrm{SN}$. How to conclude ?
  Remark 1: $\tau_y = \tau_x \rightarrow \tau_z$
  Remark 2: for $c = b_x^{s\rho}$, we have $\tau_x < \tau_y$
  Remark 3: $b_x^{s\rho} s_1\rho \dots s_n\rho = (zs_1\rho \dots s_n\rho)_z^c$ and $\tau_z < \tau_y$
- $u = \lambda x.t$. $v\rho = (\lambda x.t\rho)s\rho s_1\rho \dots s_n\rho$. We prove that $v\rho \in \mathrm{SN}$ by well-founded induction on $(t, s, s_1, \dots, s_n)$. Reducts of $v\rho$:
  - Reduction in $t\rho, s\rho, s_1\rho, \dots, s_n\rho$: IH.
  - Otherwise, the reduct is $t\rho_x^{s\rho} s_1\rho \dots s_n\rho$. How to conclude ?
    Remark 4: $t\rho_x^{s\rho} s_1\rho \dots s_n\rho = (t_x^s s_1 \dots s_n)\rho \leftarrow_\beta v\rho$ and $v \in \mathrm{SN}$

Final proof (Diederik Van Daalen, 1980)

Theorem: if $v : V$ and $a : \tau_y$ are SN, then $v_y^a \in \mathrm{SN}$.

Proof. Let $\rho = \binom{a}{y}$. By induction on (τ_y, v) using $\rightarrow_\beta \cup \rhd$ as well-founded ordering on v.

- $v \in \mathcal{X}$. If $v = y$, then $v\rho = a \in \mathrm{SN}$. Otherwise, $v\rho = v \in \mathrm{SN}$.
- $v = \lambda x.t$. By IH, $t\rho \in \mathrm{SN}$. Thus, $v\rho \in \mathrm{SN}$.
- $v = us\vec{s}$. By IH, $s\rho, \vec{s}\rho \in \mathrm{SN}$.
  - $u = x \neq y$. Then, $v\rho = xs\rho\vec{s}\rho \in \mathrm{SN}$ since, by IH, $s\rho, \vec{s}\rho \in \mathrm{SN}$.
  - $u = y$. Then, $v\rho = as\rho\vec{s}\rho$. If $v\rho \notin \mathrm{SN}$, then $a \rightarrow_\beta^* \lambda x.b$ and $b_x^{s\rho}\vec{s}\rho = (z\vec{s}\rho)_z^{b_x^{s\rho}} \notin \mathrm{SN}$. Since $b \in \mathrm{SN}$ and $\tau_x < \tau_y$, by IH, $b_x^{s\rho} \in \mathrm{SN}$. Since $z\vec{s}\rho \in \mathrm{SN}$ and $\tau_z < \tau_y$, by IH, $b_x^{s\rho}\vec{s}\rho \in \mathrm{SN}$.
  - $u = \lambda x.t$. $v\rho = (\lambda x.t\rho)s\rho\vec{s}\rho$. Reducts of $v\rho$:
    - Reduction in $t\rho, s\rho, \vec{s}\rho$: IH.
    - Otherwise, the reduct is $t\rho_x^{s\rho}\vec{s}\rho = (t_x^s\vec{s})\rho$. We have $t_x^s\vec{s} \in \mathrm{SN}$ since it is a reduct of $v \in \mathrm{SN}$. Thus, by IH, $t\rho_x^{s\rho}\vec{s}\rho \in \mathrm{SN}$.

Direct proof (Diederik Van Daalen, 1980)

- nice proof: created redexes have abstractions of decreasing types
- but we do not know how to extend it to richer type theories yet

From left to right: husband of Henriëtte, Jan van Hoek, Diederik van Daalen, Bert Jutting, Ids Zandleven, Roel de Vrijer, prof de Bruijn.

Outline

β-reduction

Van Daalen's proof (1980)

Tait's proof (1967)

William Walker Tait's approach (1967)

Idea: strengthen the induction hypothesis again

Find a property P on well-typed terms such that:

- if $P(v)$, then $v \in \mathrm{SN}$
- if $P(u : S \to T)$ and $P(s : S)$, then $P(us)$
- if $P(u)$ and $P(s)$, then $P(u_x^s)$
- $P(x)$ holds for every variable x

William Walker Tait's approach (1967)

$u : V$ is computable if:

- either $V \in \mathcal{B}$ and $u \in \mathrm{SN}$
- or $V = S \to T$ and, for all computable $s : S$, us is computable

this provides an inductive interpretation of types:

- $[\![B]\!] = \{u : B \mid u \in \mathrm{SN}\}$
  is the set of computable terms of type B
- $[\![S \to T]\!] = \{u : S \to T \mid \forall s \in [\![S]\!], us \in [\![T]\!]\}$
  is the set of computable terms of type $S \to T$

a substitution ρ is computable if, for all x, $x\rho \in [\![\tau_x]\!]$

Computability, variables and termination

Let X be the set of terminating terms of the form $xs_1 \ldots s_n$ $(n \geq 0)$

Lemma: For all type V, $\mathrm{X} \subseteq_{(1)} [\![V]\!] \subseteq_{(2)} \mathrm{SN}$.

Proof. By induction on V.

- $V \in \mathcal{B}$.
  (1) Let $v \in \mathrm{X}$. Since $\mathrm{X} \subseteq \mathrm{SN}$, $v \in \mathrm{SN}$. Thus, $v \in [\![V]\!]$.
  (2) Let $v \in [\![V]\!]$. Then, $v \in \mathrm{SN}$.
- $V = S \to T$.
  (1) Let $v = xs_1 \ldots s_n \in \mathrm{X}$ and $s_{n+1} \in [\![S]\!]$. By IH2, $s_{n+1} \in \mathrm{SN}$. Thus, $xs_1 \ldots s_{n+1} \in \mathrm{SN}$. By IH2, $xs_1 \ldots s_{n+1} \in [\![T]\!]$. Thus, $v \in [\![V]\!]$.
  (2) Let $v \in [\![V]\!]$. By IH1, there is $x \in [\![S]\!]$. Thus, $vx \in [\![T]\!]$. By IH2, $vx \in \mathrm{SN}$. Thus, $v \in \mathrm{SN}$.

Tait's approach

Lemma: If $v : V$ and ρ is computable, then $v\rho \in [\![V]\!]$.

Proof. By induction on $v : V$.

- $v \in \mathcal{X}$. We have $v\rho \in [\![V]\!]$, since ρ is computable.
- $v = us$. We have $u : S \to V$ and $s : S$. By IH, $u\rho \in [\![S \to T]\!]$ and $s\rho \in [\![S]\!]$. Thus, $v\rho \in [\![V]\!]$.

Abstraction case

- $v = \lambda x.t$. Let $s_0 = v\rho$, $S_1 = \tau_x$ and assume that $\tau_t = S_2 \rightarrow \ldots \rightarrow S_n \rightarrow B \in \mathcal{B}$. Let $s_1 \in [\![S_1]\!], \ldots, s_n \in [\![S_n]\!]$.

  Possible reducts of $s_0 s_1 \ldots s_n$:

  - $t\rho_x^{s_1} s_2 \ldots s_n \in \mathrm{SN}$ by IH
  - $s_0 \ldots s_i' \ldots s_n$ with s_i' a reduct of s_i

  Is every possible reduct SN ?

  Since each $s_i \in \mathrm{SN}$, the second case can be dealt with by well-founded induction on $(s_0, \ldots, s_n)$ if computability is preserved by reduction (the IH applies only if s_i' is computable).

Computability is preserved by reduction

Lemma: If $v \in [\![V]\!]$ and $v \to_\beta v'$, then $v' \in [\![V]\!]$.

Proof. By induction on V.

- $V \in \mathcal{B}$. Then, $v \in \mathrm{SN}$ and $v' \in \mathrm{SN}$. Thus, $v' \in [\![V]\!]$.
- $V = S \to T$. Let $s \in [\![S]\!]$. Then, $vs \in [\![T]\!]$. Since $vs \to_\beta v's$, by IH, $v's \in [\![T]\!]$. Thus, $v' \in [\![V]\!]$.

Final proof

Lemma: If $v : V$ and ρ is computable, then $v\rho \in [\![V]\!]$.

Proof. By induction 1 on $v : V$.

- $v \in \mathcal{X}$. We have $v\rho \in [\![V]\!]$, since ρ is computable.
- $v = us$. We have $u : S \to V$ and $s : S$. By IH, $u\rho \in [\![S \to T]\!]$ and $s\rho \in [\![S]\!]$. Thus, $v\rho \in [\![V]\!]$.
- $v = \lambda x.t$. Let $s_0 = v\rho$, $S_1 = \tau_x$ and assume that $\tau_t = S_2 \to \ldots \to S_n \to B$. Let $s_1 \in [\![S_1]\!], \ldots, s_n \in [\![S_n]\!]$. We then prove that $s_0 s_1 \ldots s_n \in \mathrm{SN}$ by well-founded induction 2 on $(s_0, \ldots, s_n)$. Possible reducts:
  - $t\rho_x^{s_1} s_2 \ldots s_n$ is SN by IH1.
  - $s_0 \ldots s_i' \ldots s_n$ with s_i' a reduct of s_i is SN by IH2.

Consequences

Lemma: If $v : V$ and ρ is computable, then $v\rho \in [\![V]\!]$.

Corollary: If $v : V$, then $v \in \mathrm{SN}$.

Proof. Since $[\![V]\!] \subseteq \mathrm{SN}$ and the identity substitution is computable.

Corollary: Every simply-typed λ-term has a unique β-normal form.

Proof. By termination, every term has at least one normal form. By confluence, every term has at most one normal form.

Corollary: β-equivalence is decidable.

Proof. Check that the β-normal forms are α-equivalent.

What if we add constants and δ-rules ?

Take for instance the constants:

- $c : (T \to T) \to T$
- $p : T \to (T \to T)$

and the δ-rule:

- $p(cx) \to x$

Do well-typed terms using p and c terminate ?
Let $\omega = \lambda x^T.pxx : T \to T$.
Then, $\omega(c\omega) \to_\beta p(c\omega)(c\omega) \to_\delta w(c\omega) \to_\beta \dots$!

Constants and rules introduce relations on types:

- p maps every element of T to a map from T to T. Ok.
- c maps every map from T to T to an element of T. Strange.
- $p(cx) \to x$ means that T is in bijection with the set of functions from T to T! This is possible only if $T = \emptyset$ (Cantor theorem).