Strong Normalization (SN) of Simply Typed Lambda Calculus

Typed Properties

[Strong Normalization] Every simply typed term is normalising:
if $\Gamma \vdash_\lambda t : A$, then $t \in SN_\beta$.

Defining Strongly Normalizing Terms

- **Non-inductive** definition:
  $t \in SN_\beta$ iff there is no infinite β-reduction sequence starting at t.
- Equivalent **non-inductive** definition:
  $t \in SN_\beta$ iff every β-reduction sequence starting at t is finite.
- **First inductive** alternative definition of SN_β (set SN_1):
  - If t is a β-normal form, then $t \in SN_1$
  - If $\forall t' \; [(t \to_\beta t')$ implies $t' \in SN_1]$, then $t \in SN_1$
    (the first line is a special case of the second one)
- **Second inductive** alternative definition of SN_β (set SN_2):
  - $t_1, \ldots, t_n \in SN_2$ implies $x\vec{t} = x\, t_1 \; \ldots t_n \in SN_2$.
  - $t \in SN_2$ implies $\lambda x.t \in SN_2$.
  - $t\{x\backslash u\}\vec{r} \in SN_2$ and $u \in SN_2$ implies $(\lambda x.t)u\vec{r} \in SN_2$.
- All these notions are equivalent: $\boxed{t \in SN_1 \text{ iff } t \in SN_2 \text{ iff } t \in SN_\beta}$.

Definition (Measuring SN_β-terms)

Given $t \in SN_\beta$, we define the **measure** $\mu_\beta(t)$ as $\max\{n \in \mathbb{N} \mid \; t \to_\beta^n t'\}$.

Note that $t \to_\beta t'$ implies $\mu_\beta(t') < \mu_\beta(t)$, so that $t \in SN_\beta$ and $t \to_\beta t'$ implies $t' \in SN_\beta$.

Some General Remarks About SN_β-Terms

- $u \in SN_\beta$ iff $\lambda y.u \in SN_\beta$.
- $u_1, \ldots, u_n \in SN_\beta$ iff $x\ u_1 \ldots u_n \in SN_\beta$.
- In general, if $t \in SN_\beta$, then every subterm of t is also SN_β. but the converse is not true, e.g. $(\lambda xx.x)(\lambda xx.x)$.
- This is because SN_β is not stable by substitution. Example: $x\ x \in SN_\beta$, $\lambda y.y\ y \in SN_\beta$, but $(x\ x)\{x\backslash\lambda y.y\ y\} = \Delta\ \Delta \notin SN_\beta$.

First Proof of the SN property

- This first proof is due to Tait.
- Uses the **first inductive** definition of SN_β ($SN_\beta = SN_1$)
- It is based on a predicate SC characterizing *strong computable* terms.

Definition

Let t be of type $A = A_1 \to \ldots \to A_n \to \tau$. Then $t \in SC$ iff forall $u_i \in SC$ of type A_i we have $t\,\vec{u} = t\,u_1 \ldots u_n \in SN_\beta$.

The previous definition implies

1. $SC \subseteq SN_\beta$.
2. SC is closed under β (i.e. $t \in SC$ and $t \to_\beta t'$ implies $t' \in SC$).
3. $x \in SC$ for every variable x (using 1).

Lemma

If $u, u_1, \ldots, u_n$ $(n \geq 1) \in SN_\beta$ *and* $u\{x\backslash u_1\}u_2 \ldots u_n \in SN_\beta$, *then* $t = (\lambda x.u)u_1u_2 \ldots u_n \in SN_\beta$.

Proof.

By the first inductive definition of SN_β, to show $t \in SN_\beta$ it is sufficient to show that **all** the reducts of $t = (\lambda x.u)u_1 \ldots u_n$ are in $t \in SN_\beta$. By the first hypothesis of the lemma we can proceed by **induction** on $\mu_\beta(u) + \Sigma_i\, \mu_\beta(u_i)$. We reason by case analysis on the reducts of t, which are:

- $(\lambda x.u')u_1 \ldots u_n$, where $u \rightarrow\ u'$. Then $\mu_\beta(u') < \mu_\beta(u)$, we conclude by the **i.h.**
- $(\lambda x.u)u_1 \ldots u'_i \ldots u_n$, where $u_i \rightarrow\ u'_i$. Then $\mu_\beta(u'_i) < \mu_\beta(u_i)$, we conclude by the **i.h.**
- $u\{x\backslash u_1\}u_2 \ldots u_n$. We conclude by the second hypothesis.

□

Remark: The base case of the induction is when $\mu_\beta(u) + \Sigma_i\, \mu_\beta(u_i) = 0$, *i.e.* when $u, u_1, \ldots, u_n$ are β-normal forms. The only reduct of t in this case is of the form $u\{x\backslash u_1\}u_2 \ldots u_n$, as in the third item.

Lemma

Let t be a typed term and let σ be a type preserving substitution mapping all the free variables of t to terms in SC. Then $t\sigma \in SC$.

Proof.

We proceed by **induction** on the typed term t.

- $t = x$. Then $x\sigma = \sigma(x) \in SC$ by the second hypothesis.
- $t = uv$. Then $v\sigma$ in SC by the **i.h.** Consider $r_i \in SC$ so that $v\sigma, r_1, \ldots, r_n \in SC$. Then $(uv)\sigma\vec{r} = u\sigma\ v\sigma\ \vec{r} \in SN_\beta$ by definition of $u\sigma \in SC$, which also holds by the **i.h**.
- $t = \lambda x.u$, then $(\lambda x.u)\sigma =_\alpha \lambda x.u\sigma$. Since $\sigma \cup \{x\backslash x\}$ verifies the second hypothesis of the lemma, then by the **i.h.** $u(\sigma \cup \{x\backslash x\}) = u\sigma \in SC$. To show $\lambda x.u\sigma \in SC$ we consider $r_1, \ldots, r_n \in SC$ and we show $(\lambda x.u\sigma)r_1 \ldots r_n \in SN_\beta$. This follows from the previous lemma since
  1. $u\sigma \in SN_\beta$: since $u\sigma \in SC$ and $SC \subseteq SN_\beta$.
  2. $r_1, \ldots, r_n \in SN_\beta$: since $r_1, \ldots, r_n \in SC$ and $SC \subseteq SN_\beta$.
  3. $(u\sigma)\{x\backslash r_1\}r_2 \ldots r_n \in SN_\beta$: since $(u\sigma)\{x\backslash r_1\} = u(\sigma \cup \{x\backslash r_1\})$ and $\sigma \cup \{x\backslash r_1\}$ verifies the second hypothesis of the lemma, then $(u\sigma)\{x\backslash r_1\} \in SC$ holds by the **i.h.**, and thus $(u\sigma)\{x\backslash r_1\}r_2 \ldots r_n \in SN_\beta$ holds by definition of SC.

$\square$

Lemma

Every typed term is in SC.

Proof.

Using the previous lemma with the identity substitution $\mathbf{id}$ defined by $\mathbf{id}(x) = x$ for all x. Remark that $\mathbf{id}$ is a type preserving substitution and maps variables to variables, which are terms in SC as previously remarked. $\square$

Theorem

Every typed term is in SN_β.

Proof.

Using the previous lemma and the fact the $SC \subseteq SN_\beta$. □

Second proof of the SN property

- Can be found in Femke van Raamsdonk's Thesis.
- Uses the **second inductive** definition of SN_β ($SN_\beta = SN_2$)

1. Define Λ_A (terms of type A) inductively:
   - If x is a variable of type A, then $x \in \Lambda_A$.
   - If $t \in \Lambda_C$ and x is a variable of type B, then $\lambda x.t \in \Lambda_{B \to C}$.
   - If $t \in \Lambda_{B \to A}$ and $u \in \Lambda_B$, then $t\ u \in \Lambda_A$.
2. Define $SN_A := SN_2 \cap \Lambda_A$.
3. Define $X \Rightarrow Y := \{t \mid \forall u.(u \in X \text{ implies } tu \in Y)\}$.
4. Show $\Lambda_{A \to B} = \Lambda_A \Rightarrow \Lambda_B$.
5. Show $SN_A \Rightarrow SN_B \subseteq SN_{A \to B}$ (easy).
6. If $u \in SN_{A_1} \Rightarrow SN_{A_2} \Rightarrow \ldots \Rightarrow SN_{A_m}$ with A_m a base type and $t \in SN_B$, then $t\{x \backslash u\} \in SN_B$ (induction on SN using 5).
7. Show $SN_{A \to B} \subseteq SN_A \Rightarrow SN_B$ (using 6).
8. Show that $\Lambda_A \subseteq SN_A$ (by induction using 7).
9. Since $SN_A \subseteq SN_2 = SN_\beta$ we conclude.

Third Proof of the SN property

- This first proof is due to Gandy, later rediscovered by René David.
- Uses the **first inductive** definition of SN_β ($SN_\beta = SN_1$)

Lemma

If t and u are typed and belong to SN_β, then $t\{x\backslash u\} \in SN_\beta$.

Proof.

By induction on $\langle type(u), \mu_\beta(t), size(t)\rangle$, using the standard lexicographic order. We reason by case analysis on t.

- $t = z$. If $t = x$, then $x\{x\backslash u\} = u \in SN_\beta$ by hypothesis, whereas $t = z \neq x$ implies $z\{x\backslash u\} = z$ which is trivially in SN_β.
- $t = z\ c_1 \ldots c_n$ ($z \neq x$). By the i.h. on c_i ($type(u)$ is equal, $\mu_\beta(\_)$ decreases and $size(\_)$ strictly decreases.).
- $t = x\ c_1 \ldots c_n$. By the i.h. $C_i = c_i\{x\backslash u\} \in SN_\beta$. It is sufficient to show that all the reducts of $T = t\{x\backslash u\} = u\ C_1 \ldots C_n$ are in SN_β. We reason by induction on $\mu_\beta(u) + \Sigma_i\ \mu_\beta(C_i)$. The reducts of T are:
  - $u'\ C_1 \ldots C_n$, where $u \to u'$. Apply the i.h.
  - $u\ C_1 \ldots C'_i \ldots C_n$, where $C_i \to C'_i$. Apply the i.h.
  - $v\{y\backslash C_1\}C_2 \ldots C_n$, where $u = \lambda y.v$. But $v\{y\backslash C_1\}C_2 \ldots C_n = (zC_2 \ldots C_n)\{z\backslash v\{y\backslash C_1\}\}$ and $type(v\{y\backslash C_1\}) < type(u)$. We thus conclude by the i.h. since $zC_2 \ldots C_n$ and $v\{y\backslash C_1\}$ are typed and belong to SN_β by the i.h.

□

- $t = \lambda y.v$. By the i.h. on v ($type(u)$ and $\mu_\beta(\_)$ are equal, $size(\_)$ strictly decreases).
- $t = (\lambda y.b)\ c_1 \ldots c_n$. By the i.h. $B = b\{x\backslash u\}$, and $C_i = c_i\{x\backslash u\}$ are in SN_β. It is sufficient to show that all the reducts of $T = t\{x\backslash u\} = (\lambda y.B)\ C_1 \ldots C_n$ are in SN_β. We proceed by induction on $\mu_\beta(u) + \mu_\beta(B) + \Sigma_i\ \mu_\beta(C_i)$. The reducts of T are:
  - $(\lambda y.B')\ C_1 \ldots C_n$, where $B \to B'$. Apply the i.h.
  - $(\lambda y.B)\ C_1 \ldots C'_i \ldots C_n$, where $C_i \to C'_i$. Apply the i.h.
  - $B\{y\backslash C_1\}C_2 \ldots C_n$. But $B\{y\backslash C_1\}C_2 \ldots C_n = (b\{y\backslash c_1\}c_2 \ldots c_n)\{x\backslash u\}$ and $\mu_\beta(b\{y\backslash c_1\}c_2 \ldots c_n) < \mu_\beta(t)$. Thus $B\{y\backslash C_1\}C_2 \ldots C_n \in SN_\beta$ by the i.h.

Remark: The base case ⟨ base type, 0, 1 ⟩ necessarily corresponds to a variable of base type, which is a particular case of the already detailed case $t = z$.

Theorem

If t is typable, then $t \in SN_\beta$.

Proof.

By induction on the typed term t.

- Case $t = x$ is trivial.
- Case $t = \lambda y.u$ holds by the i.h.
- For the case $t = u\ v$, we use the fact that $t = (z\ v)\{z \backslash u\}$, where z is a fresh variable, and then apply previous lemma (verification of the hypothesis is easy).

□

Fourth proof of the SN property

See for example Gandy's proof by Alexandre Miquel.
A combinatorial proof of strong normalisation for the simply typed lambda-calculus.
http://www.pps.univ-paris-diderot.fr/~miquel/publis/snlam.pdf

Strong Normalization of Girard's System F

Reducibility Candidates

- **Remind the relation $\to_F$:**

$$\begin{array}{lcl} (\lambda x : A.t)\, u & \to & t\{x\backslash u\} \\ (\Lambda \alpha t)[A] & \to & t\{\alpha\backslash A\} \end{array}$$

- **Strongly Normalizing Terms:** $t \in SN_F$ iff there is no infinite $\to_F$ reduction sequence starting at t.
- **Neutral Terms:** Terms that are not abstractions.

Definition

A reducibility candidate of type A is a set $\mathcal{R}$ of terms of type A such that

(CR1) If $t \in \mathcal{R}$, then $t \in SN_F$

(CR2) If $t \in \mathcal{R}$ and $t \to_F t'$, then $t' \in \mathcal{R}$

(CR3) If t is a neutral term and ($t \to_F t'$ implies $t' \in \mathcal{R}$), then $t \in \mathcal{R}$.

Definition

If $\mathcal{R}$ and $\mathcal{S}$ are reducibility candidates of type A and B respectively, then $\mathcal{R} \Rightarrow \mathcal{S}$ is a set of terms of type $A \to B$ defined by

$$t \in \mathcal{R} \Rightarrow \mathcal{S} \text{ iff } \forall u.(u \in \mathcal{R} \text{ implies } tu \in \mathcal{S})$$

Remarks

- A consequence of $(CR3)$: If t is a neutral and F-normal term, then $t \in \mathcal{R}$.
- $\mathcal{R}$ of type A is never empty, it contains at least the variables of type A.
- The set $\{t \in SN_F$ and t of type $A\}$ is a reducibility candidate.
- The set $\mathcal{R} \Rightarrow \mathcal{S}$ is a reducibility candidate.

Definition

Let T be a type where $\texttt{tfv}(T) \subseteq \vec{\alpha}$. We write $T\{\vec{\alpha}\backslash\vec{A}\}$ for the simultaneous substitution of $\vec{\alpha}$ by $\vec{A}$. Given $\vec{\mathcal{R}}$ a sequence of reducibility candidates, we define a set $\texttt{RED}_T(\vec{\alpha}, \vec{\mathcal{R}})$ of terms of type $T\{\vec{\alpha}\backslash\vec{A}\}$.

- If $T = \alpha_i$, then $\texttt{RED}_{\alpha_i}(\vec{\alpha}, \vec{\mathcal{R}})$ is $\mathcal{R}_i$
- If $T = A \to B$, then $\texttt{RED}_{A\to B}(\vec{\alpha}, \vec{\mathcal{R}})$ is $\texttt{RED}_A(\vec{\alpha}, \vec{\mathcal{R}}) \Rightarrow \texttt{RED}_B(\vec{\alpha}, \vec{\mathcal{R}})$
- If $T = \forall\gamma.B$, then $\texttt{RED}_{\forall\gamma.B}(\vec{\alpha}, \vec{\mathcal{R}})$ is the set of terms t of type $T\{\vec{\alpha}\backslash\vec{A}\}$ such that for every type C and reducibility candidate $\mathcal{S}$ of this type, then $t[C] \in \texttt{RED}_B(\vec{\alpha}\gamma, \vec{\mathcal{R}}\mathcal{S})$

Properties

Lemma

$\mathrm{RED}_T(\vec{\alpha}, \vec{\mathcal{R}})$ *is a reducibility candidate of type* $T\{\vec{\alpha}\backslash\vec{A}\}$.

Lemma

$\mathrm{RED}_{T\{\gamma\backslash B\}}(\vec{\alpha}, \vec{\mathcal{R}}) = \mathrm{RED}_T(\vec{\alpha}\gamma, \vec{\mathcal{R}}\,\mathrm{RED}_B(\vec{\alpha}, \vec{\mathcal{R}}))$.

Lemma

If for every type B *and candidate* $\mathcal{S}$, $t\{\gamma\backslash B\} \in \mathrm{RED}_A(\vec{\alpha}\gamma, \vec{\mathcal{R}}\mathcal{S})$, *then* $\Lambda\gamma t \in \mathrm{RED}_{\forall\gamma.A}(\vec{\alpha}, \vec{\mathcal{R}})$.

Lemma

If $t \in \mathrm{RED}_{\forall\gamma.A}(\vec{\alpha}, \vec{\mathcal{R}})$, $t[B] \in \mathrm{RED}_{A\{\gamma\backslash B\}}(\vec{\alpha}, \vec{\mathcal{R}})$. *for every type* B.

Reducible Terms

Definition

A term t of type A is reducible if $t \in \mathtt{RED}_A(\vec{\alpha}, S\vec{N})$ where $\vec{\alpha} = \alpha_1 \ldots \alpha_n$ are the free type variables of A, and $S\vec{N}$ is $SN_1 \ldots SN_n$, where SN_i is the set of terms of SN_F of type α_i.

Final Theorem

Theorem

All typed terms of system F are reducible.

Corollary (by CR1)

Corollary

All typed terms of system F are in SN_F.

Lemma

Let t be a term of type A. Suppose $\mathtt{fv}(t) \subseteq \{x_1, \ldots, x_n\}$ and x_i is of type B_i. Suppose $\mathtt{tfv}(A, B_1, \ldots, B_n) \subseteq \{\alpha_1, \ldots, \alpha_m\}$. If $\{\mathcal{R}_1, \ldots, \mathcal{R}_m\}$ are reducibility candidates of types $\{C_1, \ldots, C_m\}$, and $v_1, \ldots, v_n$ are terms of types $B_1\{\vec{\alpha}\backslash\vec{C}\}, \ldots, B_n\{\vec{\alpha}\backslash\vec{C}\}$ which are in $\mathtt{RED}_{B_1}(\vec{\alpha}, \vec{\vec{\mathcal{R}}}), \ldots, \mathtt{RED}_{B_n}(\vec{\alpha}, \vec{\vec{\mathcal{R}}})$ resp., then $t\{\vec{\alpha}\backslash\vec{C}\}\{\vec{x}\backslash\vec{v}\} \in \mathtt{RED}_A(\vec{\alpha}, \vec{\mathcal{R}})$.