11.5.2015 — 20 INRIA

V.4 Strong ~~termination~~ Normalization

A language is <u>strongly normalizing</u> (SN) if every term in the language terminates eventually.

Quite an exceptional property...

Amazingly our simply-typed language is SN!

<u>Theorem</u> $\emptyset \vdash e : \tau \curvearrowright \exists v \text{ val} .\ e \xrightarrow{*} v$

To simplify the proof consider

$e ::= () \mid x \mid \lambda(x{:}\tau) e \mid e\, e$

$\tau ::= \text{unit} \mid \tau \to \tau$

but the technique scales

<u>Proof attempt</u> (Wrong)

By induction on expression

- $()$ ✓
- $x \notin dom(\emptyset)$ (✓)
- $\lambda(x{:}\tau)e$ val ✓
- $e_1 e_2$

<u>Lemma</u> $e_1 \xrightarrow{*} v_1 \curvearrowright e_1 e_2 \xrightarrow{*} v_1 e_2$

$e_2 \xrightarrow{*} v_2 \curvearrowright v_1 e_2 \xrightarrow{*} v_1 v_2$

<u>Lemma</u> $e_1 \xrightarrow{*} e_2 \wedge e_2 \xrightarrow{*} e_3 \curvearrowright e_1 \xrightarrow{*} e_3$

assume $\emptyset \vdash e : \tau \rightsquigarrow \exists v$ val. $e \xrightarrow{*} v$

<u>Case $e_1 e_2$</u>

$e_1 \xrightarrow{*} v_1$ v_1 val by induction

$e_1 e_2 \xrightarrow{*} v_1 e_2$ congr.

$e_2 \xrightarrow{*} v_2$ v_2 val induction

$v_1 e_2 \xrightarrow{*} v_1 v_2$ Congr.

$\emptyset \vdash v_1 v_2 : \tau$ by type preservation

$\emptyset \vdash v_1 : \tau_1 \to \tau$ by inversion

$v_1 = \lambda(x{:}\tau_1)\, e$ canonical forms

$v_1 v_2 \mapsto e[x := v_2]$ β-reduction

$e[x := v_2] \xrightarrow{*} v_3$ "induction"

$\rightsquigarrow e_1 e_2 \xrightarrow{*} v_3$

Observation: the term can grow on reduction

INRIA

Successful path to proof:
use a <u>logical relation</u>, i.e.,
a type-indexed relation on terms.

Here: unary relation = predicate.

[idea due to Tait 1967]

<u>Def</u> logical relation $R_\tau(e)$

- $R_{unit}(e)$ iff e halts and $\emptyset \vdash e : unit$
- $R_{\tau_1 \to \tau_2}(e)$ iff e halts and $\emptyset \vdash e : \tau_1 \to \tau_2$ and $\forall e' : R_{\tau_1}(e') \leadsto R_{\tau_2}(e e')$

Def. ~~desired~~ property @ base type
then extend logically to higher type
by requiring that functions preserve the prop.

It remains to show

1. If $\emptyset \vdash e : \tau$, then $R_\tau(e)$
2. If $R_\tau(e)$, then e halts

~~Lemma~~
Proof (part 2): immediate by def.

11.5.2015 - 23

Ad part 1: need to establish that the relation is preserved under evaluation

Lemma If ~~$\emptyset \vdash$~~ $e : \tau$ and $e \mapsto e'$
then $R_\tau(e) \Leftrightarrow R_\tau(e')$

Proof ~~immediate~~ because $\mapsto$ is deterministic
~~$\emptyset \vdash$~~ [e halts iff e' halts]

Induction on $\boxed{\tau}$!

Case unit:
assume $\emptyset \vdash e : \text{unit}$ and $e \mapsto e'$
Show e halts $\wedge\ \emptyset \vdash e : \text{unit} \iff e'$ halts $\wedge\ \emptyset \vdash e' : \text{unit}$
✓

Case $\tau_1 \to \tau_2$:
Additionally we need
1. $R_{\tau_1}(e_1) \wedge R_{\tau_2}(e\,e_1) \Rightarrow R_{\tau_2}(e'e_1)$
2. $R_{\tau_1}(e_1) \wedge R_{\tau_2}(e'e_1) \Rightarrow R_{\tau_2}(e\,e_1)$

ad 1.
$\emptyset \vdash e_1 : \tau_1$ $\quad R_{\tau_1}(e_1)$
$\emptyset \vdash e : \tau_1 \to \tau_2$ $\quad$ ass.
$\curvearrowright$ $\underline{\emptyset \vdash e\,e_1 : \tau_2}$ $\quad$ typing
$e \mapsto e'$ $\curvearrowright$ $\underline{e\,e_1 \mapsto e'e_1}$
$\curvearrowright$ $R_{\tau_2}(e\,e_1) \iff R_\tau(e'e_1)$ by induction

<u>Lemma</u> If $\emptyset \vdash e : \tau$ then $R_\tau(e)$

<u>Proof</u> induction on $\emptyset \vdash e : \tau$

⋮

but for $\dfrac{x:\tau_1 \vdash e:\tau_2}{\emptyset \vdash \lambda(x:\tau_1)\, e : \tau_2}$ the IH is not applicable

generalize to

<u>Lemma</u> If $x_1:\tau_1 \dots x_n:\tau_n \vdash e:\tau$

and $R_{\tau_1}(v_1) \dots R_{\tau_n}(v_n)$

then $R_\tau(e[x_i := v_i])$

<u>Proof</u> induction of typing derivation.