# An Introduction to Logical Relations

## Chapters 1 & 2

Tom van Bussel

May 2, 2017

# Logical Relations

Better name: Type Indexed Inductive relations

| Logical predicates | Logical relations |
|---|---|
| (Unary) | (Binary) |
| $P_\tau(e)$ | $R_\tau(e_1, e_2)$ |
| -Strong normalization | -Program equivalence |
| -Type safety | |

# Notation

### Types

$\tau ::= \mathit{bool} \mid \tau \to \tau$

### Expressions

$\mathsf{e} ::= x \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ e\ \mathsf{then}\ e\ \mathsf{else}\ e \mid \lambda x : \tau.\ e \mid \mathsf{e}\ \mathsf{e}$

### Values

$\mathsf{v} ::= \mathsf{true} \mid \mathsf{false} \mid \lambda x : \tau.\ e$

### Typing contexts

$\Gamma ::= \bullet \mid \Gamma, x : \tau$

# Typing Rules

$$\frac{}{\Gamma \vdash \mathit{false} : \mathit{bool}} \; \mathrm{T-False}$$

$$\frac{}{\Gamma \vdash \mathit{true} : \mathit{bool}} \; \mathrm{T-True}$$

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau} \; \mathrm{T-Var}$$

$$\frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x : \tau_1 . e : \tau_1 \rightarrow \tau_2} \; \mathrm{T-Abs}$$

$$\frac{\Gamma \vdash e_1 : \tau_2 \rightarrow \tau_1 \quad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1 e_2 : \tau_1} \; \mathrm{T-App}$$

$$\frac{\Gamma \vdash e : \mathit{bool} \quad \Gamma \vdash e_1 : \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \; \text{if } e \text{ then } e_1 \text{ else } e_2 : \tau} \; \mathrm{T-If}$$

# Evaluation

## Evaluation contexts

$E ::= [] \mid \text{if } E \text{ then } e \text{ else } e \mid e\ E \mid E\ v$

## Rules

- if true then $e_1$ else $e_2 \mapsto e_1$
- if false then $e_1$ else $e_2 \mapsto e_2$
- $(\lambda x : \tau . e) v \mapsto e[v/x]$

$$\frac{e \mapsto e'}{E[e] \mapsto E[e']}$$

- 

# Strong Normalization

## Notation

- $e \Downarrow v \iff e \mapsto^* v$
- $e \Downarrow \iff \exists v . e \Downarrow v$

## Theorem (Strong Normalization)

*If* $\bullet \vdash e : \tau$ *then* $e \Downarrow$.

## Proof by induction on structure of derivation does not work

$$\frac{\Gamma \vdash e_1 : \tau_2 \to \tau_1 \quad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1 e_2 : \tau_1} \; \mathrm{T-App}$$

# Logical Predicate for Strongly Normalizing Expressions

$SN_{bool}(e)$ iff

- $\bullet \vdash e : bool$ and
- $e \Downarrow$.

$SN_{\tau_1 \to \tau_2}(e)$ iff

- $\bullet \vdash e : \tau_1 \to \tau_2$,
- $e \Downarrow$, and
- $\forall e'. SN_{\tau_1}(e') \implies SN_{\tau_2}(e\ e')$.

# Structure of proof

(a) $\bullet \vdash e : \tau \implies SN_\tau(e)$

(b) $SN_\tau(e) \implies e \Downarrow$

# Substitutions

$$\frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x : \tau_1 . e : \tau_1 \to \tau_2} \ \mathrm{T-Abs}$$

## Substitution

$\emptyset(e) = e$

$\gamma[x \mapsto v](e) = \gamma(e[v/x])$

$$\gamma \models \Gamma \iff dom(\gamma) = dom(\Gamma) \land \forall x \in dom(\Gamma). SN_{\Gamma(x)}(\gamma(x))$$

## Generalized (a)

If $\Gamma \vdash e : \tau$ and $\gamma \models \Gamma$ then $SN_\tau(\gamma(e))$

# Lemmas

**Lemma (Substitution lemma)**

*If* $\Gamma \vdash e : \tau$ *and* $\gamma \models \Gamma$ *then* $\bullet \vdash \gamma(e) : \tau$.

**Lemma (Preservation lemma)**

*Suppose* $\bullet \vdash e : \tau$ *and* $e \mapsto e'$ *then*

1. *if* $SN_\tau(e')$, *then* $SN_\tau(e)$
2. *if* $SN_\tau(e)$, *then* $SN_\tau(e')$

# Proof

Proof by induction on the derivation of $\Gamma \vdash e : \tau$.

## Case

$$\frac{}{\Gamma \vdash \mathit{true} : \mathit{bool}} \text{ T} - \text{True}$$

Given $\gamma \models \Gamma$ we need to show that $SN_{bool}(\gamma(\mathit{true}))$.

## Case

$$\frac{}{\Gamma \vdash \mathit{false} : \mathit{bool}} \text{ T} - \text{False}$$

This case is similar to the case of true.

# Proof

## Case

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau}\ \mathrm{T-Var}$$

This case follows from $\gamma \models \Gamma$ and well-typedness of $x$:

- $x$ is well-typed, so $x \in \mathit{dom}(\Gamma)$
- From $\gamma \models \Gamma$ we get $SN_{\Gamma(x)}(\gamma(x))$
- $x$ is well-typed, so $\Gamma(x) = \tau$.

# Proof

## Case

$$\frac{\Gamma \vdash e_1 : \tau_2 \to \tau_1 \quad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1 e_2 : \tau_1} \; \mathrm{T-App}$$

Need to show that $SN_\tau(\gamma(e_1\ e_2))$, which is equivalent to $SN_\tau(\gamma(e_1)\ \gamma(e_2))$.

By the induction hypothesis we have:

(1) $SN_{\tau_2 \to \tau_1}(\gamma(e_1))$

(2) $SN_{\tau_2}(\gamma(e_2))$

By (1) we have $\forall e'.SN_{\tau_2}(e') \implies SN_{\tau_1}(\gamma(e_1)e')$.
Combined with (2) this gives us the result.

# Proof

## Case

$$\frac{\Gamma \vdash e : bool \quad \Gamma \vdash e_1 : \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \text{ if } e \text{ then } e_1 \text{ else } e_2 : \tau} \text{ T} - \text{If}$$

By the induction hypothesis we have:

(1) $SN_{bool}(e)$

(2) $SN_\tau(e_1)$

(3) $SN_\tau(e_2)$

By (1) $e \Downarrow v$. Two cases $v = \text{true}$ or $v = \text{false}$.

- in first case: $\text{if } e \text{ then } e_1 \text{ else } e_2 \mapsto e_1$.
- in second case: $\text{if } e \text{ then } e_1 \text{ else } e_2 \mapsto e_2$.

So by the preservation lemma $SN_\tau(\text{if } e \text{ then } e_1 \text{ else } e_2)$.

# Proof

## Case

$$\frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x : \tau_1 . e : \tau_1 \rightarrow \tau_2} \ \mathrm{T-Abs}$$

Given $\gamma \models \Gamma$, need to show:

1. $\bullet \vdash \lambda x : \tau_1 . \gamma(e) : \tau_1 \rightarrow \tau_2$
2. $\lambda x : \tau_1 . \gamma(e) \Downarrow$
3. $\forall e' . SN_{\tau_1}(e') \implies SN_{\tau_2}((\lambda x : \tau_1 . \gamma(e))\ e')$

Induction hypothesis:
$\Gamma, x : \tau_1 \vdash e : \tau_2 \ \wedge \ \gamma' \models \Gamma, x : \tau_1 \implies SN_{\tau_2}(\gamma'(e))$

$(\lambda x : \tau_1 . \gamma(e))\ e' \mapsto^* \gamma[x \mapsto v'](e)$