Logical Relations: Type safety for STLC

Eline Bovy Tosca Klijnsma Alex van de Griendt

What is type safety?

Informally:

- Well-typed programs do not "go wrong".
- If a term is well-typed, then it won't get stuck.

What is type safety?

Informally:

- Well-typed programs do not "go wrong".
- If a term is well-typed, then it won't get stuck.

For the simply typed lambda calculus:
Type safety: If $\cdot \vdash e : \tau$ and $e \to^* e'$, then $\mathrm{Val}(e')$ or $\exists e''.e' \to e''$

What is type safety?

Informally:

- Well-typed programs do not "go wrong".
- If a term is well-typed, then it won't get stuck.

For the simply typed lambda calculus:
Type safety: If $\cdot \vdash e : \tau$ and $e \to^* e'$, then $\mathsf{Val}(e')$ or $\exists e''.e' \to e''$

$$\tau ::= \mathit{bool} \mid \tau_1 \to \tau_2$$

$$e ::= x \mid \texttt{true} \mid \texttt{false} \mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \mid \lambda x : \tau.e \mid e_1\, e_2$$

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

What is type safety?

Informally:

- Well-typed programs do not "go wrong".
- If a term is well-typed, then it won't get stuck.

For the simply typed lambda calculus:
Type safety: If $\cdot \vdash e : \tau$ and $e \to^* e'$, then $\mathrm{Val}(e')$ or $\exists e''.e' \to e''$

$$\tau ::= \mathit{bool} \mid \tau_1 \to \tau_2$$
$$e ::= x \mid \texttt{true} \mid \texttt{false} \mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \mid \lambda x : \tau.e \mid e_1\, e_2$$
$$v ::= \texttt{true} \mid \texttt{false} \mid \lambda x : \tau.e$$

Examples of stuck terms: $\texttt{true}\ \texttt{false}$, $\texttt{false}\ (\lambda x : \mathit{bool}.x)$, ...

The classical treatment

Type safety: If $\cdot \vdash e : \tau$ and $e \to^* e'$, then $\mathsf{Val}(e')$ or $\exists e''.e' \to e''$.

The classical treatment

Type safety: If $\cdot \vdash e : \tau$ and $e \to^* e'$, then $\mathsf{Val}(e')$ or $\exists e''.e' \to e''$.

Progress: If $\cdot \vdash e : \tau$, then $\mathsf{Val}(e)$ or $\exists e'.e \to e'$.

Preservation: If $\cdot \vdash e : \tau$ and $e \to e'$, then $\cdot \vdash e' : \tau$.

Logical predicates

A logical predicate is a proof method for proving program properties. Usual notation: $P_\tau(e)$, "$e : \tau$ has property P".

Logical predicates

A logical predicate is a proof method for proving program properties. Usual notation: $P_\tau(e)$, "$e : \tau$ has property P".

Proof of type safety in two phases:

(A) If e is well-typed, then it belongs to the logical predicate:

$$\cdot \vdash e : \tau \Rightarrow P_\tau(e)$$

Logical predicates

A logical predicate is a proof method for proving program properties. Usual notation: $P_\tau(e)$, "$e : \tau$ has property P".

Proof of type safety in two phases:

(A) If e is well-typed, then it belongs to the logical predicate:

$$\cdot \vdash e : \tau \Rightarrow P_\tau(e)$$

(B) If e belongs to the logical predicate, then e is type safe:

$$P_\tau(e) \Rightarrow \mathsf{safe}(e)$$

where $\mathsf{safe}(e) := \forall e'.e \to^* e' \Rightarrow \mathsf{Val}(e') \vee \exists e''.e' \to e''$.

Logical predicates

(A) $\cdot \vdash e : \tau \Rightarrow P_\tau(e)$

(B) $P_\tau(e) \Rightarrow \mathit{safe}(e)$

How do we construct $P_\tau(e)$ such that we can prove this?

Logical predicates

(A) $\cdot \vdash e : \tau \Rightarrow P_\tau(e)$

(B) $P_\tau(e) \Rightarrow \mathit{safe}(e)$

How do we construct $P_\tau(e)$ such that we can prove this?

We want the following things to hold true for expressions e in the logical predicate $P_\tau(e)$:

(1) The expression has the property we are interested in.

Logical predicates

(A) $\cdot \vdash e : \tau \Rightarrow P_\tau(e)$

(B) $P_\tau(e) \Rightarrow \mathit{safe}(e)$

How do we construct $P_\tau(e)$ such that we can prove this?

We want the following things to hold true for expressions e in the logical predicate $P_\tau(e)$:

(1) The expression has the property we are interested in.

(2) The property of interest is preserved by elimination forms.

$$\frac{\Gamma \vdash e_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1\, e_2 : \tau_2}\ \textsc{App}$$

Value and expression interpretation

(A) $\cdot \vdash e : \tau \Rightarrow P_\tau(e)$

(B) $P_\tau(e) \Rightarrow \mathit{safe}(e)$

We define the logical predicate $P_\tau(e)$ in two parts: a value interpretation $\mathcal{V}[\![-]\!]$ and an expression interpretation $\mathcal{E}[\![-]\!]$.

Value and expression interpretation

(A) $\cdot \vdash e : \tau \Rightarrow P_\tau(e)$

(B) $P_\tau(e) \Rightarrow \mathit{safe}(e)$

We define the logical predicate $P_\tau(e)$ in two parts: a value interpretation $\mathcal{V}[\![-]\!]$ and an expression interpretation $\mathcal{E}[\![-]\!]$.

Value interpretation:

$$\mathcal{V}[\![\mathit{bool}]\!] = \{\texttt{true}, \texttt{false}\}$$
$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{\lambda x : \tau_1.e \mid \forall v \in \mathcal{V}[\![\tau_1]\!].e[v/x] \in \mathcal{E}[\![\tau_2]\!]\}$$

Value and expression interpretation

(A) $\cdot \vdash e : \tau \Rightarrow P_\tau(e)$

(B) $P_\tau(e) \Rightarrow \mathit{safe}(e)$

We define the logical predicate $P_\tau(e)$ in two parts: a value interpretation $\mathcal{V}[\![-]\!]$ and an expression interpretation $\mathcal{E}[\![-]\!]$.

Value interpretation:

$$\mathcal{V}[\![\mathit{bool}]\!] = \{\texttt{true}, \texttt{false}\}$$
$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{\lambda x : \tau_1.e \mid \forall v \in \mathcal{V}[\![\tau_1]\!].e[v/x] \in \mathcal{E}[\![\tau_2]\!]\}$$

Expression interpretation:

$$\mathcal{E}[\![\tau]\!] = \{e \mid \forall e'.e \to^* e' \wedge \mathsf{irred}(e') \Rightarrow e' \in \mathcal{V}[\![\tau]\!]\}$$

where $\mathsf{irred}(e) := \nexists e'.e \to e'$.

Proof structure

(A) $\cdot \vdash e : \tau \Rightarrow e \in \mathcal{E}[\![\tau]\!]$

(B) $e \in \mathcal{E}[\![\tau]\!] \Rightarrow \mathit{safe}(e)$

Proof of (A) by induction on the typing derivation.

However, this gives a problem for the case ABS:

$$\frac{x : \tau_1 \vdash e : \tau_2}{\cdot \vdash \lambda x : \tau_1 . e : \tau_1 \to \tau_2}\ \textsc{Abs}$$

For part (A), we need to close off our term e.

Environment interpretation

(A) $\cdot \vdash e : \tau \Rightarrow e \in \mathcal{E}[\![\tau]\!]$

(B) $e \in \mathcal{E}[\![\tau]\!] \Rightarrow \mathit{safe}(e)$

Environment interpretation: What are the sets of good substitutions that we can use to close off our term e?

Environment interpretation

(A) $\cdot \vdash e : \tau \Rightarrow e \in \mathcal{E}[\![\tau]\!]$

(B) $e \in \mathcal{E}[\![\tau]\!] \Rightarrow \mathit{safe}(e)$

Environment interpretation: What are the sets of good substitutions that we can use to close off our term e?

$$\mathcal{G}[\![\cdot]\!] = \{\emptyset\}$$
$$\mathcal{G}[\![\Gamma, x : \tau]\!] = \{\gamma[x \mapsto v] \mid \gamma \in \mathcal{G}[\![\Gamma]\!] \wedge v \in \mathcal{V}[\![\tau]\!]\}$$

Environment interpretation

(A) $\cdot \vdash e : \tau \Rightarrow e \in \mathcal{E}[\![\tau]\!]$

(B) $e \in \mathcal{E}[\![\tau]\!] \Rightarrow \mathit{safe}(e)$

Environment interpretation: What are the sets of good substitutions that we can use to close off our term e?

$$\mathcal{G}[\![\cdot]\!] = \{\emptyset\}$$
$$\mathcal{G}[\![\Gamma, x : \tau]\!] = \{\gamma[x \mapsto v] \mid \gamma \in \mathcal{G}[\![\Gamma]\!] \wedge v \in \mathcal{V}[\![\tau]\!]\}$$

Semantic type safety:

$$\Gamma \vDash e : \tau \ := \ \forall \gamma \in \mathcal{G}[\![\Gamma]\!].\gamma(e) \in \mathcal{E}[\![\tau]\!]$$

Proof structure

Proof structure:

(A) The fundamental property: If e is syntactically well-typed, then e is semantically well-typed:

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \vDash e : \tau$$

(B) If e is semantically well-typed, then e is type safe:

$$\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

Definition of safe(e):

$$\mathsf{safe}(e) := \forall e'.e \rightarrow^* e' \Rightarrow \mathsf{Val}(e') \vee \exists e''.e' \rightarrow e''$$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

Definition of safe(e):

$$\mathsf{safe}(e) := \forall e'.e \rightarrow^* e' \Rightarrow \mathsf{Val}(e') \vee \exists e''.e' \rightarrow e''$$

- Suppose $e \rightarrow^* e_1$
- We need to show $\mathit{Val}(e_1) \vee \exists e_2.e_1 \rightarrow e_2$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

Definition of safe(e):

$$\mathsf{safe}(e) := \forall e'.e \to^* e' \Rightarrow \mathsf{Val}(e') \vee \exists e''.e' \to e''$$

- Suppose $e \to^* e_1$
- We need to show $\mathit{Val}(e_1) \vee \exists e_2.e_1 \to e_2$
- Case distinction: $\mathit{irred}(e_1) \vee \neg \mathit{irred}(e_1)$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

Definition of safe(e):

$$\text{safe}(e) := \forall e'.e \to^* e' \Rightarrow \text{Val}(e') \vee \exists e''.e' \to e''$$

- Suppose $e \to^* e_1$
- We need to show $\mathit{Val}(e_1) \vee \exists e_2.e_1 \to e_2$
- Case distinction: $\mathit{irred}(e_1) \vee \neg \mathit{irred}(e_1)$

Definition of irred(e):

$$\text{irred}(e) := \nexists e'.e \to e'$$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\neg \mathit{irred}(e_1)$
  - Then $\neg(\nexists e_2.e_1 \rightarrow e_2) = \exists e_2.e_1 \rightarrow e_2$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\neg \mathit{irred}(e_1)$
  - Then $\neg(\nexists e_2.e_1 \rightarrow e_2) = \exists e_2.e_1 \rightarrow e_2$
  - Hence $\mathit{Val}(e_1) \vee \exists e_2.e_1 \rightarrow e_2$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\neg \mathit{irred}(e_1)$
  - Then $\neg(\nexists e_2.e_1 \rightarrow e_2) = \exists e_2.e_1 \rightarrow e_2$
  - Hence $\mathit{Val}(e_1) \vee \exists e_2.e_1 \rightarrow e_2$
- Suppose $\mathit{irred}(e_1) = \nexists e_2.e_1 \rightarrow e_2$
  - Then $\nexists e_2.e_1 \rightarrow e_2$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\neg \mathit{irred}(e_1)$
  - Then $\neg(\nexists e_2.e_1 \rightarrow e_2) = \exists e_2.e_1 \rightarrow e_2$
  - Hence $\mathit{Val}(e_1) \vee \exists e_2.e_1 \rightarrow e_2$
- Suppose $\mathit{irred}(e_1) = \nexists e_2.e_1 \rightarrow e_2$
  - Then $\nexists e_2.e_1 \rightarrow e_2$
  - By assumption we know $\cdot \vDash e : \tau$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\neg \mathit{irred}(e_1)$
  - Then $\neg(\nexists e_2.e_1 \rightarrow e_2) = \exists e_2.e_1 \rightarrow e_2$
  - Hence $\mathit{Val}(e_1) \vee \exists e_2.e_1 \rightarrow e_2$
- Suppose $\mathit{irred}(e_1) = \nexists e_2.e_1 \rightarrow e_2$
  - Then $\nexists e_2.e_1 \rightarrow e_2$
  - By assumption we know $\cdot \vDash e : \tau$

Definition of semantic type safety:

$$\Gamma \vDash e : \tau \;:=\; \forall \gamma \in \mathcal{G}[\![\Gamma]\!].\gamma(e) \in \mathcal{E}[\![\tau]\!]$$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\mathit{irred}(e_1) = \nexists e_2.e_1 \rightarrow e_2$
  - Then $\nexists e_2.e_1 \rightarrow e_2$
  - By assumption we know $\cdot \vDash e : \tau$
  - So, by semantic type safety, $e \in \mathcal{E}[\![\tau]\!]$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\mathit{irred}(e_1) = \nexists e_2.e_1 \rightarrow e_2$
  - Then $\nexists e_2.e_1 \rightarrow e_2$
  - By assumption we know $\cdot \vDash e : \tau$
  - So, by semantic type safety, $e \in \mathcal{E}[\![\tau]\!]$

Definition of $\mathcal{E}[\![\tau]\!]$:

$$\mathcal{E}[\![\tau]\!] = \{e \mid \forall e'.e \rightarrow^* e' \wedge \mathsf{irred}(e') \Rightarrow e' \in \mathcal{V}[\![\tau]\!]\}$$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\mathit{irred}(e_1) = \nexists e_2.e_1 \rightarrow e_2$
  - Then $\nexists e_2.e_1 \rightarrow e_2$
  - By assumption we know $\cdot \vDash e : \tau$
  - So, by semantic type safety, $e \in \mathcal{E}[\![\tau]\!]$

Definition of $\mathcal{E}[\![\tau]\!]$:

$$\mathcal{E}[\![\tau]\!] = \{e \mid \forall e'.e \rightarrow^* e' \wedge \mathsf{irred}(e') \Rightarrow e' \in \mathcal{V}[\![\tau]\!]\}$$

- So $\forall e'.e \rightarrow^* e' \wedge \mathsf{irred}(e') \Rightarrow e' \in \mathcal{V}[\![\tau]\!]$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\mathit{irred}(e_1) = \nexists e_2.e_1 \rightarrow e_2$
  - Then $\nexists e_2.e_1 \rightarrow e_2$
  - By assumption we know $\cdot \vDash e : \tau$
  - So, by semantic type safety, $e \in \mathcal{E}[\![\tau]\!]$

Definition of $\mathcal{E}[\![\tau]\!]$:

$$\mathcal{E}[\![\tau]\!] = \{e \mid \forall e'.e \rightarrow^* e' \wedge \mathsf{irred}(e') \Rightarrow e' \in \mathcal{V}[\![\tau]\!]\}$$

- So $\forall e'.e \rightarrow^* e' \wedge \mathsf{irred}(e') \Rightarrow e' \in \mathcal{V}[\![\tau]\!]$
- Then $e_1 \in \mathcal{V}[\![\tau]\!]$, hence $\mathit{Val}(e_1) \vee \exists e_2.e_1 \rightarrow e_2$

Proof of part B

(B) $\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$

- Suppose $\mathit{irred}(e_1) = \nexists e_2.e_1 \rightarrow e_2$
  - Then $\nexists e_2.e_1 \rightarrow e_2$
  - By assumption we know $\cdot \vDash e : \tau$
  - So, by semantic type safety, $e \in \mathcal{E}[\![\tau]\!]$

Definition of $\mathcal{E}[\![\tau]\!]$:

$$\mathcal{E}[\![\tau]\!] = \{e \mid \forall e'.e \rightarrow^* e' \wedge \mathsf{irred}(e') \Rightarrow e' \in \mathcal{V}[\![\tau]\!]\}$$

  - So $\forall e'.e \rightarrow^* e' \wedge \mathsf{irred}(e') \Rightarrow e' \in \mathcal{V}[\![\tau]\!]$
  - Then $e_1 \in \mathcal{V}[\![\tau]\!]$, hence $\mathit{Val}(e_1) \vee \exists e_2.e_1 \rightarrow e_2$
- Hence $\mathit{safe}(e)$

Proof of part A

The fundamental property

(A) $\Gamma \vdash e : \tau \rightarrow \Gamma \models e : \tau$

- To prove part A we need a helpful lemma called the *Substitution Lemma*

Substitution Lemma

- Let e be a syntactically well-typed form
- Let v be a closed value
- Let γ be a substitution that maps variables to closed values
- Let x be a variable not in the domain of γ

Then:

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

Substitution Lemma

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

Definition of subsitution:

$$\emptyset(e) = e$$

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

Substitution Lemma

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

Definition of subsitution:

$$\emptyset(e) = e$$

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

By induction on the size γ.

- $\gamma = \emptyset$:

$$[x \mapsto v]e = e[v/x]$$

Substitution Lemma

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

- $\gamma = \gamma'[y \mapsto v']$ $(x \neq y)$:

Substitution Lemma

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

- $\gamma = \gamma'[y \mapsto v']$ $(x \neq y)$:

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

Substitution Lemma

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

- $\gamma = \gamma'[y \mapsto v']$ $(x \neq y)$:

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

$$\gamma[x \mapsto v]e \quad = \gamma'[y \mapsto v'][x \mapsto v]e$$

Substitution Lemma

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

- $\gamma = \gamma'[y \mapsto v']$ $(x \neq y)$:

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

$$\begin{aligned} \gamma[x \mapsto v]e &= \gamma'[y \mapsto v'][x \mapsto v]e \\ &= \gamma'[x \mapsto v][y \mapsto v']e \end{aligned}$$

Substitution Lemma

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

- $\gamma = \gamma'[y \mapsto v']$ $(x \neq y)$:

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

$$\begin{aligned} \gamma[x \mapsto v]e &= \gamma'[y \mapsto v'][x \mapsto v]e \\ &= \gamma'[x \mapsto v][y \mapsto v']e \\ &= \gamma'[x \mapsto v](e[v'/y]) \end{aligned}$$

Substitution Lemma

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

- $\gamma = \gamma'[y \mapsto v']$ $(x \neq y)$:

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

$$\begin{aligned}
\gamma[x \mapsto v]e &= \gamma'[y \mapsto v'][x \mapsto v]e \\
&= \gamma'[x \mapsto v][y \mapsto v']e \\
&= \gamma'[x \mapsto v](e[v'/y]) \\
&\overset{\text{IH}}{=} \gamma'(e[v'/y])[v/x]
\end{aligned}$$

Substitution Lemma

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

- $\gamma = \gamma'[y \mapsto v']$ $(x \neq y)$:

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

$$\begin{aligned}
\gamma[x \mapsto v]e &= \gamma'[y \mapsto v'][x \mapsto v]e \\
&= \gamma'[x \mapsto v][y \mapsto v']e \\
&= \gamma'[x \mapsto v](e[v'/y]) \\
&\overset{\text{IH}}{=} \gamma'(e[v'/y])[v/x] \\
&= \gamma'[y \mapsto v'](e)[v/x]
\end{aligned}$$

Substitution Lemma

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

- $\gamma = \gamma'[y \mapsto v']$ $(x \neq y)$:

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

$$\begin{aligned}
\gamma[x \mapsto v]e &= \gamma'[y \mapsto v'][x \mapsto v]e \\
&= \gamma'[x \mapsto v][y \mapsto v']e \\
&= \gamma'[x \mapsto v](e[v'/y]) \\
&\overset{\text{IH}}{=} \gamma'(e[v'/y])[v/x] \\
&= \gamma'[y \mapsto v'](e)[v/x] \\
&= \gamma(e)[v/x]
\end{aligned}$$

Fundamental Property

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \models e : \tau$$

By induction on the typing judgment:

Fundamental Property

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \models e : \tau$$

By induction on the typing judgment:

`False`: $\Gamma \vdash \texttt{false} : \mathit{bool} \Rightarrow \Gamma \models \texttt{false} : \mathit{bool}$

Fundamental Property

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \models e : \tau$$

By induction on the typing judgment:

False: $\Gamma \vdash \texttt{false} : \mathit{bool} \Rightarrow \Gamma \models \texttt{false} : \mathit{bool}$

True: $\Gamma \vdash \texttt{true} : \mathit{bool} \Rightarrow \Gamma \models \texttt{true} : \mathit{bool}$

Fundamental Property

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \models e : \tau$$

By induction on the typing judgment:

False: $\Gamma \vdash \texttt{false} : \mathit{bool} \Rightarrow \Gamma \models \texttt{false} : \mathit{bool}$

True: $\Gamma \vdash \texttt{true} : \mathit{bool} \Rightarrow \Gamma \models \texttt{true} : \mathit{bool}$

If: $\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau \Rightarrow$
$\Gamma \models \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau$

Fundamental Property

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \models e : \tau$$

By induction on the typing judgment:

False: $\Gamma \vdash \texttt{false} : \mathit{bool} \Rightarrow \Gamma \models \texttt{false} : \mathit{bool}$

True: $\Gamma \vdash \texttt{true} : \mathit{bool} \Rightarrow \Gamma \models \texttt{true} : \mathit{bool}$

If: $\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau \Rightarrow$
$\Gamma \models \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau$

Abs: $\Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2 \Rightarrow$
$\Gamma \models \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$

Fundamental Property

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \models e : \tau$$

By induction on the typing judgment:

False: $\Gamma \vdash \texttt{false} : \mathit{bool} \Rightarrow \Gamma \models \texttt{false} : \mathit{bool}$

True: $\Gamma \vdash \texttt{true} : \mathit{bool} \Rightarrow \Gamma \models \texttt{true} : \mathit{bool}$

If: $\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau \Rightarrow$
$\Gamma \models \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau$

Abs: $\Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2 \Rightarrow$
$\Gamma \models \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$

App: $\Gamma \vdash e_1 e_2 : \tau \Rightarrow \Gamma \models e_1 e_2 : \tau$

Fundamental Property

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \models e : \tau$$

By induction on the typing judgment:

False: $\Gamma \vdash \texttt{false} : \mathit{bool} \Rightarrow \Gamma \models \texttt{false} : \mathit{bool}$

True: $\Gamma \vdash \texttt{true} : \mathit{bool} \Rightarrow \Gamma \models \texttt{true} : \mathit{bool}$

If: $\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau \Rightarrow$
$\Gamma \models \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau$

Abs: $\Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2 \Rightarrow$
$\Gamma \models \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$

App: $\Gamma \vdash e_1 e_2 : \tau \Rightarrow \Gamma \models e_1 e_2 : \tau$

Var: $\Gamma \vdash x : \tau \Rightarrow \Gamma \models x : \tau$

False

$$\Gamma \vdash \texttt{false} : \mathit{bool} \Rightarrow \Gamma \models \texttt{false} : \mathit{bool}$$

- `false` contains no variables, so all subsitutions $\gamma \in \mathcal{G}[\![\Gamma]\!]$ must leave `false` intact:

$$\gamma(\texttt{false}) = \texttt{false}$$

$$\Gamma \models e : \tau \iff \forall \gamma \in \mathcal{G}[\![\Gamma]\!].\gamma(e) \in \mathcal{E}[\![\tau]\!]$$

False

$$\Gamma \vdash \texttt{false} : \mathit{bool} \Rightarrow \Gamma \models \texttt{false} : \mathit{bool}$$

- `false` contains no variables, so all subsitutions $\gamma \in \mathcal{G}[\![\Gamma]\!]$ must leave `false` intact:

$$\gamma(\texttt{false}) = \texttt{false}$$

- We know that $\texttt{false} \in \mathcal{E}[\![\mathit{bool}]\!]$ since it is an irreducible value in $\mathcal{V}[\![\mathit{bool}]\!]$

$$\Gamma \models e : \tau \iff \forall \gamma \in \mathcal{G}[\![\Gamma]\!].\gamma(e) \in \mathcal{E}[\![\tau]\!]$$

Abs

$$\Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$$
$$\Rightarrow \quad \Gamma \models \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$$

Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and write $F = \lambda x : \tau_1.\gamma(e)$. To show:

$$F \in \mathcal{E}[\![\tau_1 \to \tau_2]\!]$$

Definition $\mathcal{V}[\![\tau_1 \to \tau_2]\!]$:

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{\lambda x : \tau_1.e \mid \forall v \in \mathcal{V}[\![\tau_1]\!].e[v/x] \in \mathcal{E}[\![\tau_2]\!]\}$$

Abs

$$\Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$$
$$\Rightarrow \quad \Gamma \models \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$$

Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and write $F = \lambda x : \tau_1.\gamma(e)$. To show:

$$F \in \mathcal{E}[\![\tau_1 \to \tau_2]\!]$$

- Suppose $F \to^* e'$ with irred(e')

Definition $\mathcal{V}[\![\tau_1 \to \tau_2]\!]$:

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{\lambda x : \tau_1.e \mid \forall v \in \mathcal{V}[\![\tau_1]\!].e[v/x] \in \mathcal{E}[\![\tau_2]\!]\}$$

Abs

$$\Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$$
$$\Rightarrow \quad \Gamma \models \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$$

Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and write $F = \lambda x : \tau_1.\gamma(e)$. To show:

$$F \in \mathcal{E}[\![\tau_1 \to \tau_2]\!]$$

- Suppose $F \to^* e'$ with irred(e')
- F is a value, hence irreducible, and hence $F = e'$

Definition $\mathcal{V}[\![\tau_1 \to \tau_2]\!]$:

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{\lambda x : \tau_1.e \mid \forall v \in \mathcal{V}[\![\tau_1]\!].e[v/x] \in \mathcal{E}[\![\tau_2]\!]\}$$

Abs

$$\begin{array}{rl} & \Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2 \\ \Rightarrow & \Gamma \models \lambda x : \tau_1.\ e : \tau_1 \to \tau_2 \end{array}$$

Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and write $F = \lambda x : \tau_1.\gamma(e)$. To show:

$$F \in \mathcal{E}[\![\tau_1 \to \tau_2]\!]$$

- Suppose $F \to^* e'$ with irred(e')
- F is a value, hence irreducible, and hence $F = e'$
- To show: $F \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]$

Definition $\mathcal{V}[\![\tau_1 \to \tau_2]\!]$:

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{\lambda x : \tau_1.e \mid \forall v \in \mathcal{V}[\![\tau_1]\!].e[v/x] \in \mathcal{E}[\![\tau_2]\!]\}$$

Abs

$$\Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$$
$$\Rightarrow \quad \Gamma \models \lambda x : \tau_1.\ e : \tau_1 \to \tau_2$$

Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and write $F = \lambda x : \tau_1.\gamma(e)$. To show:

$$F \in \mathcal{E}[\![\tau_1 \to \tau_2]\!]$$

- Suppose $F \to^* e'$ with irred(e')
- F is a value, hence irreducible, and hence $F = e'$
- To show: $F \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]$

Definition $\mathcal{V}[\![\tau_1 \to \tau_2]\!]$:

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{\lambda x : \tau_1.e \mid \forall v \in \mathcal{V}[\![\tau_1]\!].e[v/x] \in \mathcal{E}[\![\tau_2]\!]\}$$

- This means: for $v \in \mathcal{V}[\![\tau_1]\!]$ we must show $\gamma(e)[v/x] \in \mathcal{E}[\![\tau_2]\!]$

Abs

$$v \in \mathcal{V}[\![\tau_1]\!] \Rightarrow \gamma(e)[v/x] \in \mathcal{E}[\![\tau_2]\!]$$

Induction hypothesis:

$$\Gamma, x : \tau_1 \models e : \tau_2$$

- Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and $v \in \mathcal{V}[\![\tau_2]\!]$

Abs

$$v \in \mathcal{V}[\![\tau_1]\!] \Rightarrow \gamma(e)[v/x] \in \mathcal{E}[\![\tau_2]\!]$$

Induction hypothesis:

$$\Gamma, x : \tau_1 \models e : \tau_2$$

- Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and $v \in \mathcal{V}[\![\tau_2]\!]$
- Instantiate the induction hypothesis with $\gamma[x \mapsto v]$ to get: $\gamma[x \mapsto v] \in \mathcal{G}[\![\Gamma, x : \tau_1]\!]$

Abs

$$v \in \mathcal{V}[\![\tau_1]\!] \Rightarrow \gamma(e)[v/x] \in \mathcal{E}[\![\tau_2]\!]$$

Induction hypothesis:

$$\Gamma, x : \tau_1 \models e : \tau_2$$

- Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and $v \in \mathcal{V}[\![\tau_2]\!]$
- Instantiate the induction hypothesis with $\gamma[x \mapsto v]$ to get: $\gamma[x \mapsto v] \in \mathcal{G}[\![\Gamma, x : \tau_1]\!]$
- $\Rightarrow \gamma[x \mapsto v](e) \in \mathcal{E}[\![\tau_2]\!]$

Abs

$$v \in \mathcal{V}[\![\tau_1]\!] \Rightarrow \gamma(e)[v/x] \in \mathcal{E}[\![\tau_2]\!]$$

Induction hypothesis:

$$\Gamma, x : \tau_1 \models e : \tau_2$$

- Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$ and $v \in \mathcal{V}[\![\tau_2]\!]$
- Instantiate the induction hypothesis with $\gamma[x \mapsto v]$ to get: $\gamma[x \mapsto v] \in \mathcal{G}[\![\Gamma, x : \tau_1]\!]$
- $\Rightarrow \gamma[x \mapsto v](e) \in \mathcal{E}[\![\tau_2]\!]$
- $\Rightarrow \gamma(e)[v/x] \in \mathcal{E}[\![\tau_2]\!]$ by the substitution lemma

Proof structure:

(A) The fundamental property: If e is syntactically well-typed, then e is semantically well-typed:

$$\Gamma \vdash e : \tau \Rightarrow \Gamma \vDash e : \tau$$

(B) If e is semantically well-typed, then e is type safe:

$$\cdot \vDash e : \tau \Rightarrow \mathit{safe}(e)$$

We now have $\cdot \vdash e : \tau \Rightarrow \mathsf{safe}(e)$

Extending the language

Pairs

$$\tau := \ldots \mid \tau \times \tau$$

$$e := \ldots \mid \langle e, e \rangle \mid \mathsf{fst}\ \ e \mid \mathsf{snd}\ \ e$$

Extending the language

Pairs

$$\tau := \ldots \mid \tau \times \tau$$

$$e := \ldots \mid \langle e, e \rangle \mid \mathsf{fst}\ \ e \mid \mathsf{snd}\ \ e$$

$$\mathcal{V}[\![\tau_1 \times \tau_2]\!] = \{\langle v_1, v_2 \rangle \mid v_1 \in \mathcal{V}[\![\tau_1]\!] \wedge v_2 \in \mathcal{V}[\![\tau_2]\!]\}$$

Extending the language

Pairs

$$\tau := \ldots \mid \tau \times \tau$$

$$e := \ldots \mid \langle e, e \rangle \mid \mathsf{fst}\ \ e \mid \mathsf{snd}\ \ e$$

$$\mathcal{V}[\![\tau_1 \times \tau_2]\!] = \{\langle v_1, v_2 \rangle \mid v_1 \in \mathcal{V}[\![\tau_1]\!] \wedge v_2 \in \mathcal{V}[\![\tau_2]\!]\}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \mathsf{fst}\, e : \tau_1}\ \textsc{Fst} \qquad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \mathsf{snd}\, e : \tau_2}\ \textsc{Snd}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2}\ \textsc{Pair}$$

Extending the language

Sums

$$\tau := \ldots \mid \tau + \tau$$

$$e := \ldots \mid \mathsf{inl}\, e \mid \mathsf{inr}\, e \mid \mathsf{case}\ e\ \mathsf{of}\ \mathsf{inl}\, x \Rightarrow e_1;\ \mathsf{inr}\, x \Rightarrow e_2$$

Extending the language

Sums

$$\tau := \ldots \mid \tau + \tau$$

$$e := \ldots \mid \mathsf{inl}\, e \mid \mathsf{inr}\, e \mid \mathsf{case}\ e\ \mathsf{of}\ \mathsf{inl}\, x \Rightarrow e_1;\ \mathsf{inr}\, x \Rightarrow e_2$$

$$\mathcal{V}[\![\tau_1 + \tau_2]\!] = \{\mathsf{inl}\, v \mid v \in \mathcal{V}[\![\tau_1]\!]\} \cup \{\mathsf{inr}\, v \mid v \in \mathcal{V}[\![\tau_2]\!]\}$$

Extending the language

Sums

$$\tau := \ldots \mid \tau + \tau$$

$$e := \ldots \mid \mathsf{inl}\, e \mid \mathsf{inr}\, e \mid \mathsf{case}\ e\ \mathsf{of}\ \mathsf{inl}\, x \Rightarrow e_1;\ \mathsf{inr}\, x \Rightarrow e_2$$

$$\mathcal{V}[\![\tau_1 + \tau_2]\!] = \{\mathsf{inl}\, v \mid v \in \mathcal{V}[\![\tau_1]\!]\} \cup \{\mathsf{inr}\, v \mid v \in \mathcal{V}[\![\tau_2]\!]\}$$

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \mathsf{inl}\, e : \tau_1 + \tau_2}\ \textsc{Inl} \qquad \frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \mathsf{inr}\, e : \tau_1 + \tau_2}\ \textsc{Inr}$$

$$\frac{\Gamma \vdash e : \tau_1 + \tau_2 \quad \Gamma, x_1 : \tau_1 \vdash e_1 : \tau \quad \Gamma, x_2 : \tau_2 \vdash e_2 : \tau}{\Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ \mathsf{inl}\, x_1 \Rightarrow e_1; \mathsf{inr}\, x_2 \Rightarrow e_2 : \tau}\ \textsc{Case}$$