CS 365 — Programming Language Concepts

Type correctness proofs

Apr 16, 2008

Type rules

$$\Gamma \vdash \texttt{true} : \mathsf{bool} \tag{1}$$

$$\Gamma \vdash \texttt{false} : \mathsf{bool} \tag{2}$$

$$\Gamma \vdash x : \Gamma(x) \tag{3}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{bool} \qquad \Gamma \vdash e_2 : \tau \qquad \Gamma \vdash e_3 : \tau}{\Gamma \vdash \ \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3 : \tau} \tag{4}$$

$$\frac{\Gamma \cup \{(x_1, \tau_1)\} \vdash e : \tau_2}{\Gamma \vdash \mathsf{fn}(x) \Rightarrow e : \tau_1 \rightarrow \tau_2} \tag{5}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_1 \rightarrow \tau_2}{\Gamma \vdash e_1(e_2) : \tau_2} \tag{6}$$

Definitions

An expression e is $well\text{-}typed$ in a type system if there exists an environment Γ and type τ such that the judgment $\Gamma \vdash e : \tau$ can be proven by the type rules.

A type system is $sound$ if well-typed programs cannot cause type errors.

An expression is $closed$ if it has no free variables.

A $program$ is a closed expression.

A $value$ is a closed abstraction or a boolean constant. We will use the variable v to range over values.

Semantic rules

$$(\mathsf{fn}(x) \Rightarrow e)(v) \longrightarrow [v/x]e \tag{7}$$

$$\text{if } v \text{ then } e_2 \text{ else } e_3 \longrightarrow \begin{cases} e_2 & \text{if } v = \texttt{true} \\ e_3 & \text{otherwise} \end{cases} \tag{8}$$

$$\frac{e_1 \longrightarrow e_1'}{e_1(e_2) \longrightarrow e_1'(e_2)} \tag{9}$$

$$\frac{e_2 \longrightarrow e_2'}{v_1(e_2) \longrightarrow v_1(e_2')} \tag{10}$$

$$\frac{e_1 \longrightarrow e_1'}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \longrightarrow \text{ if } e_1' \text{ then } e_2 \text{ else } e_3} \tag{11}$$

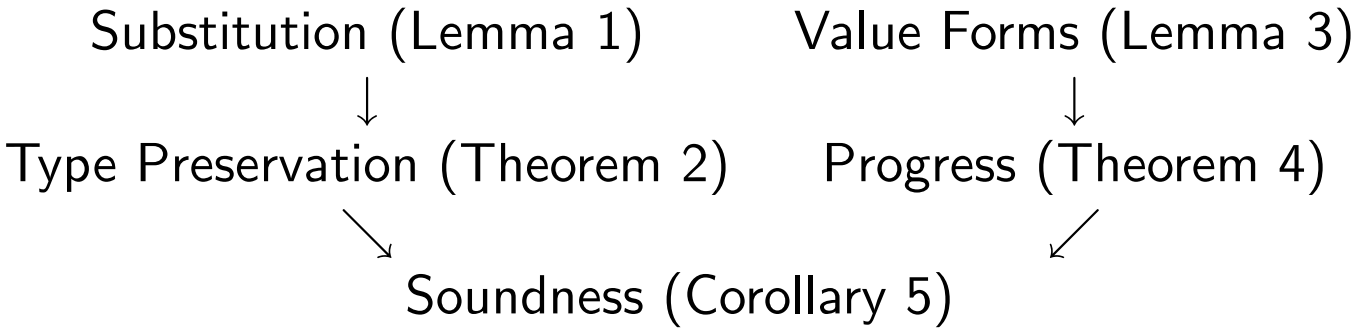
Claim

An expression e is $stuck$ if it is not a value and there does not exist an e' such that $e \longrightarrow e'$.

An expression $goes\ wrong$ if it evaluates to a stuck expression.

Claim: The BoolEm type system is sound. That is, well-typed BoolEm programs cannot go wrong.

Proof Outline.



Theorems

Lemma 1. [Substitution.] *If* $\Gamma \cup \{(x, \tau')\} \vdash e : \tau$ *and* $\Gamma \vdash v : \tau'$, *then* $\Gamma \vdash [v/x]e : \tau$.

Theorem 2. [Type Preservation.] *If* $\Gamma \vdash e : \tau$ *and* $e \longrightarrow e'$, *then* $\Gamma \vdash e' : \tau$.

Lemma 3. [Value Forms.] *If* $\Gamma \vdash v$: *bool, then* v *is in the form* `true` *or* `false`. *If* $\Gamma \vdash v : \tau_1 \rightarrow \tau_2$, *then* v *is in the form* fn$(x) \Rightarrow e$.

Theorem 4. [Progress.] *If* e *is a closed expression and* $\Gamma \vdash e : \tau$, *then either* e *is a value or there exists an* e' *such that* $e \longrightarrow e'$.

Corollary 5. [Soundness] *Well-typed programs cannot go wrong.*