Type Soundness

Intuitively, the type system should rule out eventually stuck expressions.

Recall that e is eventually stuck if $e \xrightarrow{*} e'$ and e' is stuck.

Theorem (Type Soundness) If $\vdash e : T$ then either e's evaluation does not terminate or there exists v such that $e \xrightarrow{*} v$ and $\vdash v : T$.

- $\vdash e : T$ is shorthand for $\{\} \vdash e : T$

The simply-typed lambda calculus actually satisfies a stronger version of the theorem. Every expression in the simply-typed lambda calculus terminates!

Soundness = Progress + Preservation

Theorem (Type Soundness) If $\vdash e : T$ then either e's evaluation does not terminate or there exists v such that $e \xrightarrow{*} v$ and $\vdash v : T$.

A well-typed term e is not stuck:

Theorem (Progress): If $\vdash e : T$, then either e is a value or there exists e' such that $e \longrightarrow e'$.

The $\longrightarrow$ relation preserves typing:

Theorem (Type Preservation): If $\Gamma \vdash e : T$ and $e \longrightarrow e'$, then $\Gamma \vdash e' : T$.

Preliminaries

The typing rules make implicit assumptions about the values of various types. For example, T-If assumes that true and false are the only values of type Bool:

$$\frac{\Gamma \vdash e_1 : \text{Bool} \quad \Gamma \vdash e_2 : T \quad \Gamma \vdash e_3 : T}{\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : T} \text{(T-If)}$$

The Canonical Forms Lemma validates these assumptions, which are needed by the Progress and Type Preservation proofs.

Lemma (Canonical Forms):

1. If $\Gamma \vdash v : \text{Bool}$, then v is either true or false.
2. If $\Gamma \vdash v : T_1 \rightarrow T_2$, then v has the form $\lambda x : T_1.e$.

Proof "by inspection" of the typing rules.

Preliminaries (cont.)

To prove Type Preservation, we need to reason about the type correctness of the substitution function.

Well-typed substitution produces well-typed terms:

Lemma (Substitution): If $\Gamma \cup \{x : T\} \vdash e' : T'$ and $\Gamma \vdash v : T$, then $\Gamma \vdash [x \mapsto v]e' : T'$.

Proof by strong induction on the depth of the derivation tree for $\Gamma \cup \{x : T\} \vdash e' : T'$

Straightforward but somewhat tedious. See the handout for the details.

Proving Progress and Type Preservation

Theorem (Progress): If $\vdash e : T$, then either e is a value or there exists e' such that $e \longrightarrow e'$.

- Proof by strong induction on the depth of the derivation tree for $\vdash e : T$.
- Case analysis of the last rule used in the derivation.
- Inductive hypothesis allows us to assume the theorem for sub-expressions of e.

Theorem (Type Preservation): If $\Gamma \vdash e : T$ and $e \longrightarrow e'$, then $\Gamma \vdash e' : T$.

- Proof by strong induction on the depth of the derivation tree for $\Gamma \vdash e : T$.
- Case analysis of the last rule used in the derivation.
- Sub-case analysis of the last rule used in the derivation of $e \longrightarrow e'$.
- Inductive hypothesis allows us to assume the theorem for sub-expressions of e.

Simply-Typed Lambda Calculus

Syntax

$$e ::= x \mid \lambda x : T.e \mid e_1\ e_2 \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3$$

$$T ::= \mathsf{Bool} \mid T_1 \to T_2 \qquad v ::= \lambda x : T.e \mid \mathsf{true} \mid \mathsf{false}$$

Operational Semantics

$$\frac{}{(\lambda x : T.e)v \longrightarrow [x \mapsto v]e}\ \text{(E-AppRed)} \qquad \frac{e_1 \longrightarrow e_1'}{e_1\ e_2 \longrightarrow e_1'\ e_2}\ \text{(E-App1)} \qquad \frac{e \longrightarrow e'}{v\ e \longrightarrow v\ e'}\ \text{(E-App2)}$$

$$\frac{}{\mathsf{if\ true\ then}\ e_2\ \mathsf{else}\ e_3 \longrightarrow e_2}\ \text{(E-IfTrue)} \qquad \frac{}{\mathsf{if\ false\ then}\ e_2\ \mathsf{else}\ e_3 \longrightarrow e_3}\ \text{(E-IfFalse)}$$

$$\frac{e_1 \longrightarrow e_1'}{\mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3 \longrightarrow \mathsf{if}\ e_1'\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3}\ \text{(E-If)}$$

Typing Rules

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T}\ \text{(T-Var)} \qquad \frac{\Gamma \cup \{x : T_1\} \vdash e : T_2}{\Gamma \vdash (\lambda x : T_1.e) : T_1 \to T_2}\ \text{(T-Abs)} \qquad \frac{\Gamma \vdash e_1 : T_2 \to T \quad \Gamma \vdash e_2 : T_2}{\Gamma \vdash e_1\ e_2 : T}\ \text{(T-App)}$$

$$\frac{}{\Gamma \vdash \mathsf{true} : \mathsf{Bool}}\ \text{(T-True)} \qquad \frac{}{\Gamma \vdash \mathsf{false} : \mathsf{Bool}}\ \text{(T-False)}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{Bool} \quad \Gamma \vdash e_2 : T \quad \Gamma \vdash e_3 : T}{\Gamma \vdash \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3 : T}\ \text{(T-If)}$$