Review:
stuck = bad
decidable type system
soundness
completeness

Type Soundness

If $\cdot \vdash e : T$ and $e \xrightarrow{*} e'$, then either (1) e' is a value or (2) $\exists\, e''$, $e' \to e''$.

// well typed terms never get stuck

Proof.

Corollary of Progress and Preservation.

Progress:

If $\cdot \vdash e : T$, then either (1) e is a value or (2) $\exists\, e'$, $e \to e'$.

Preservation:

If $\cdot \vdash e : T$ and $e \to e'$, then $\cdot \vdash e' : T$

□

Given these, we can prove Type Soundness.

By induction on $e \xrightarrow{*} e'$.

Base: $\emptyset$ steps

- $e' = e$
- exactly progress

Inductive: Proof for n steps. Show for n+1.

- $e \xrightarrow{n} e' \rightarrow e''$
- need either e'' value or e''' s.t. $e'' \rightarrow e'''$
- $\cdot \vdash e : \tau$ and Preservation gives $\cdot \vdash e' : \tau$

(+ induction!)
↳ easy lemma

- Progress now provides either e'' value or e''' s.t. $e'' \rightarrow e'''$

Qed.

Progress

If $\cdot \vdash e : \tau$, then either (1) e is a value or (2) $\exists e'$, $e \to e'$.

Proof.

Will use Canonical Forms Lemma:

(A) If $\cdot \vdash v : \text{int}$, then $\exists c$, $v = c$

(B) If $\cdot \vdash v : \tau_1 \to \tau_2$, then $\exists x\ e$, $v = \lambda x.\ e$

■

By induction on the derivation of $\cdot \vdash e : \tau$.

T-CONST :
- ▷ $e = c$
- ▷ c is a value

T-VAR :
- ▷ $e = x$
- ▷ $\cdot(x) = \tau$ ... impossible! Contradiction.
  - no derivation of form $\cdot \vdash e : \tau$ can end w/ T-VAR rule

T-FUN :

- ▷ $e = \lambda x.e'$
- ▷ $\lambda.e'$ is a value

T-APP :

- ▷ $e = e_1 \; e_2$
- ▷ By inversion $\exists \tau'$ s.t.
  - → $\cdot \vdash e_1 : \tau' \to \tau$
  - → $\cdot \vdash e_2 : \tau'$
- ▷ if e_1 is not a value
  - ▷ IH + $\cdot \vdash e_1 : \tau' \to \tau$ provides e_1' s.t. $e_1 \to e_1'$
  - ▷ By E-APP1, $e_1 \; e_2 \to e_1' \; e_2$
- ▷ e_1 is a value
  - ▷ if e_2 is not a value
    - ▷ IH + $\cdot \vdash e_2 : \tau'$ provides e_2' s.t. $e_2 \to e_2'$
    - ▷ By E-APP2, $e_1 \; e_2 \to e_1 \; e_2'$
  - ▷ e_2 is a value

(e_1 and e_2 values)

▷ $\cdot \vdash e_1 : T' \to T$ + Canonical Forms provides x, e' s.t. $e_1 = \lambda x.e'$

▷ By E-APPLY, $(\lambda x.e')\ e_2 \to e'[e_2/x]$

Qed.

Canonical Forms

(A) If $\cdot \vdash v : \text{int}$, then $\exists c, v = c$.

▷ $\cdot \vdash v : \text{int}$ can only be derived by T-CONST, which requires $\exists c$ s.t. $v = c$.

(B) If $\cdot \vdash V : T_1 \to T_2$ then $\exists x\ e, v = \lambda x.e$.

▷ $\cdot \vdash V : T_1 \to T_2$ can only be derived by T-FUN, which requires x, e s.t. $V = \lambda x.e$.

Preservation

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

Proof.

Will use Substitution Lemma:

If $\Gamma, x : \tau' \vdash e : \tau$ and $\Gamma \vdash e' : \tau'$,
then $\Gamma \vdash e[e'/x] : \tau$.

■

By induction on the derivation of $\bullet \vdash e : \tau$.

T-CONST: ▷ $e = c$
▷ $c \to e'$ ... impossible! Contradiction.

T-VAR: ▷ $e = x$
▷ $\bullet(x) = \tau$ ... impossible! Contradiction.

T-FUN: ▷ $e = \lambda x. e_b$
▷ $\lambda x. e_b \to e'$ ... impossible! Contradiction.

T-APP: ▷ $e = e_1\ e_2$

▷ By inversion $\exists\ T'$ s.t.

$\rightarrow$ $\bullet \vdash e_1 : T' \rightarrow T$

$\rightarrow$ $\bullet \vdash e_2 : T'$

▷ 3 cases for derivation of $e_1\ e_2 \rightarrow e'$

E-APP1: ▷ $e' = e_1'\ e_2$ and $e_1 \rightarrow e_1'$

▷ IH + $\bullet \vdash e_1 : T' \rightarrow T$ + $e_1 \rightarrow e_1'$ provides

$\bullet \vdash e_1' : T' \rightarrow T$

▷ By T-APP w/ $\bullet \vdash e_1' : T' \rightarrow T$ +

$\bullet \vdash e_2 : T'$, we get

$\bullet \vdash e_1'\ e_2 : T$

E-APP2: ▷ $e' = e_1\ e_2'$ and $e_2 \rightarrow e_2'$

▷ IH + $\bullet \vdash e_2 : T'$ + $e_2 \rightarrow e_2'$ provides

$\bullet \vdash e_2' : T'$

▷ By T-APP w/ $\bullet \vdash e_1 : T' \rightarrow T$ +

$\bullet \vdash e_2' : T'$ we get

$\bullet \vdash e_1\ e_2' : T$

E-APPLY :

▷ $e_1 = \lambda x.\, e_b$

▷ $e' = e_b[e_2/x]$

▷ By inversion on $\cdot \vdash e_1 : T' \to T$, we get $\cdot, x : T' \vdash e_b : T$

▷ By Substitution Lemma w/

- $\cdot, x : T' \vdash e_b : T$ ∧
- $\cdot \vdash e_2 : T'$ provides
- $\cdot \vdash e_b[e_2/x] : T$

Qed.

Substitution

If $\Gamma, x : T' \vdash e : T$ and $\Gamma \vdash e' : T'$,
then $\Gamma \vdash e[e'/x] : T$.

Proof.

Will use Weakening:

If $\Gamma \vdash e : T$ and $x \notin Dom(\Gamma)$, then
$\Gamma, x : T' \vdash e : T$.

Will use Exchange:

If $\Gamma, x : T_1, y : T_2 \vdash e : T$ and $x \neq y$, then
$\Gamma, y : T_2, x : T_1 \vdash e : T$.

■

By induction on the derivation of
$\Gamma, x : T' \vdash e : T$.

T-const: ▷ $e = c$ and $T = int$
▷ $c[e'/x] = c$
▷ By T-const, $\Gamma \vdash c : int$

T-VAR : ▷ $e = y$ and $T = (\Gamma, x : T')(y)$

▷ $\Gamma, x : T' \vdash y : T$

▷ if $x \neq y$

▷ $y[e'/x] = y$

▷ since $x \neq y$, $\Gamma(y) = T$

▷ By T-VAR, $\Gamma \vdash y : T$

▷ if $x = y$

▷ $y[e'/x] = e'$

▷ since $(\Gamma, x : T')(x) = T$, $T' = T$

▷ By assumption, $\Gamma \vdash e' : T'$,

so $\Gamma \vdash e' : T$

$\downarrow$

$y[e'/x]$

T-APP : ▷ $e = e_1\ e_2$

▷ $e[e'/x] = (e_1[e'/x])\ (e_2[e'/x])$

▷ $\Gamma, x : T' \vdash e_1\ e_2 : T$ from assumption

▷ inverting yields some T_1 s.t.

▷ $\Gamma, x : T' \vdash e_1 : T_1 \to T$

▷ $\Gamma, x : T' \vdash e_2 : T_1$

▷ IH + $\Gamma, x : T' \vdash e_1 : T_1 \to T$ provides

$$\Gamma \vdash e_1[e'/x] : T_1 \to T$$

▷ IH + $\Gamma, x : T' \vdash e_2 : T_1$ provides

$$\Gamma \vdash e_2[e'/x] : T_1$$

▷ By T-APP $\Gamma \vdash (e_1[e'/x])(e_2[e'/x]) : T$

▷ By defn of subs

$$\Gamma \vdash (e_1\ e_2)[e'/x] : T$$

T-FUN: ▷ $e = \lambda y . e_b$

~~$e[e'/x] = e$~~

▷ pick $z \neq x$ and $z \notin Dom(\Gamma)$

▷ α convert $\lambda y . e_b$ to $\lambda z . e_c$

▷ $e[e'/x] = (\lambda z . e_c)[e'/x]$

$= \lambda z . e_c[e'/x]$

▷ $\Gamma, x : T' \vdash \lambda z . e_c : T_1 \to T_2 = T$

▷ inverting yields

$$\Gamma, x : T', z : T_1 \vdash e_c : T_2$$

▷ By Exchange $\Gamma, z : T_1, x : T' \vdash e_c\ T_2$

$\square$

▷ By Weakening: $\Gamma, z : T_1 \vdash e' : T'$

▷ By IH:

$$\Gamma, z : T_1 \vdash e_c[e'/x] : T_2$$

▷ By T-Fun

$$\Gamma \vdash \lambda z . e_c[e'/x] : T_1 \to T_2$$

▷ By defn subst

$$\Gamma \vdash (\lambda z. e_c)[e'/x] : T_1 \to T_2$$

Q.e.d.