Announcements

• Proof-writing workshop tomorrow night, 7pm, in Gates 310!
Type “Completeness”?

Are all well-behaved programs well-typed?
Normalization

The simply-typed lambda calculus enjoys a remarkable property:

Every well-typed program terminates.
Simply-Typed Lambda Calculus

Syntax

expressions \( e ::= x \mid \lambda x : \tau. \ e \mid e_1 \ e_2 \mid () \)

values \( v ::= \lambda x : \tau. \ e \mid () \)

types \( \tau ::= \text{unit} \mid \tau_1 \to \tau_2 \)
## Simply-Typed Lambda Calculus

### Syntax

- **expressions**
  \[ e ::= x \mid \lambda x : \tau. e \mid e_1 e_2 \mid () \]

- **values**
  \[ v ::= \lambda x : \tau. e \mid () \]

- **types**
  \[ \tau ::= \textsf{unit} \mid \tau_1 \rightarrow \tau_2 \]

### Dynamic Semantics

- **Environment**
  \[ E ::= [] \mid E \ e \mid \nu E \]

- **Substitution**
  \[
  \begin{align*}
  & e \rightarrow e' \\
  & E[e] \rightarrow E[e'] \\
  & (\lambda x : \tau. e) \nu \rightarrow e[\nu/x]
  \end{align*}
  \]
Simply-Typed Lambda Calculus

Static Semantics

\[ \Gamma \vdash () : \text{unit} \]

\[ \Gamma(x) = \tau \quad \text{T-VAR} \]

\[ \Gamma \vdash x : \tau \]

\[ \Gamma, x : \tau \vdash e : \tau' \quad \text{T-Abs} \]

\[ \Gamma \vdash \lambda x : \tau. e : \tau \rightarrow \tau' \]

\[ \Gamma \vdash e_1 : \tau \rightarrow \tau' \quad \Gamma \vdash e_2 : \tau \quad \text{T-App} \]

\[ \Gamma \vdash e_1 \ e_2 : \tau' \]
Lemma (Inversion)

- If $\Gamma \vdash x : \tau$ then $\Gamma(x) = \tau$
- If $\Gamma \vdash \lambda x : \tau_1. e : \tau$ then $\tau = \tau_1 \rightarrow \tau_2$ and $\Gamma, x : \tau_1 \vdash e : \tau_2$.
- If $\Gamma \vdash e_1 e_2 : \tau$ then $\Gamma \vdash e_1 : \tau' \rightarrow \tau$ and $\Gamma \vdash e_2 : \tau'$. 
Supporting Lemmas

**Lemma (Inversion)**

- If $\Gamma \vdash x : \tau$ then $\Gamma(x) = \tau$
- If $\Gamma \vdash \lambda x : \tau_1. e : \tau$ then $\tau = \tau_1 \rightarrow \tau_2$ and $\Gamma, x : \tau_1 \vdash e : \tau_2$.
- If $\Gamma \vdash e_1 e_2 : \tau$ then $\Gamma \vdash e_1 : \tau' \rightarrow \tau$ and $\Gamma \vdash e_2 : \tau'$.

**Lemma (Canonical Forms)**

- If $\Gamma \vdash v : \text{unit}$ then $v = ()$
- If $\Gamma \vdash v : \tau_1 \rightarrow \tau_2$ then $v = \lambda x : \tau_1. e$ and $\Gamma, x : \tau_1 \vdash e : \tau_2$. 
Theorem (Normalization)

If $\vdash e : \tau$ then there exists a value $v$ such that $e \rightarrow^* v$. 
Logical Relations

Idea: define a set with the following properties:

- At base types the set contains all expressions satisfying some property.
- At function types, the set contains all expressions such that the property is preserved whenever we apply the function to an argument of appropriate type that is also in the set.
Logical Relations

Idea: define a set with the following properties:

- At base types the set contains all expressions satisfying some property.
- At function types, the set contains all expressions such that the property is preserved whenever we apply the function to an argument of appropriate type that is also in the set.

In our setting, the property will concern normalization...
Definition (Logical Relation)

- $R_{\text{unit}}(e)$ iff $\vdash e : \text{unit}$ and $e$ halts.
- $R_{\tau_1 \rightarrow \tau_2}(e)$ iff $\vdash e : \tau_1 \rightarrow \tau_2$ and $e$ halts, and for every $e'$ such that $R_{\tau_1}(e')$ we have $R_{\tau_2}(e \ e')$. 
Supporting Lemmas

Lemma

If $R_\tau(e)$ then $e$ halts.
Supporting Lemmas

Lemma
If $R_\tau(e)$ then $e$ halts.

Lemma
If $\vdash e : \tau$ and $e \rightarrow e'$ then $R_\tau(e)$ iff $R_\tau(e')$. 
Supporting Lemmas

Lemma

If $R_\tau(e)$ then $e$ halts.

Lemma

If $\vdash e : \tau$ and $e \rightarrow e'$ then $R_\tau(e)$ iff $R_\tau(e')$.

Lemma (Goal)

If $\vdash e : \tau$ then $R_\tau(e)$
Main Lemma

Lemma (Goal – Strengthened)

If

- $x_1 : \tau_1, \ldots, x_k : \tau_k \vdash e : \tau$,
- $v_1$ through $v_k$ are values such that $\vdash v_1 : \tau_1$ through $\vdash v_k : \tau_k$, and
- $R_{\tau_1}(v_1)$ through $R_{\tau_k}(v_k)$,

then $R_{\tau}(e[v_1/x_1] \ldots [v_k/x_k])$. 