CS 4110

Programming Languages & Logics

Lecture 20
Simply-Typed Lambda Calculus

Simply-Typed Lambda Calculus

Syntax

expressions	$e ::= x \mid \lambda x\!:\!\tau.\, e \mid e_1\; e_2 \mid n \mid e_1 + e_2 \mid ()$
values	$v ::= \lambda x\!:\!\tau.\, e \mid n \mid ()$
types	$\tau ::= \textbf{int} \mid \textbf{unit} \mid \tau_1 \to \tau_2$

Simply-Typed Lambda Calculus

Syntax

expressions	$e ::= x \mid \lambda x\!:\!\tau.\, e \mid e_1\, e_2 \mid n \mid e_1 + e_2 \mid ()$
values	$v ::= \lambda x\!:\!\tau.\, e \mid n \mid ()$
types	$\tau ::= \textbf{int} \mid \textbf{unit} \mid \tau_1 \to \tau_2$

Dynamic Semantics

$$E ::= [\cdot] \mid E\, e \mid v\, E \mid E + e \mid v + E$$

$$\frac{e \to e'}{E[e] \to E[e']} \qquad \frac{}{(\lambda x\!:\!\tau.\, e)\, v \to e\{v/x\}} \qquad \frac{n = n_1 + n_2}{n_1 + n_2 \to n}$$

Simply-Typed Lambda Calculus

Static Semantics

Simply-Typed Lambda Calculus

Static Semantics

$$\frac{}{\Gamma \vdash n : \textbf{int}}\ \textsc{T-Int}$$

Simply-Typed Lambda Calculus

Static Semantics

$$\frac{}{\Gamma \vdash n : \textbf{int}}\ \textsc{T-Int} \qquad \frac{}{\Gamma \vdash () : \textbf{unit}}\ \textsc{T-Unit}$$

Simply-Typed Lambda Calculus

Static Semantics

$$\frac{}{\Gamma \vdash n : \textbf{int}}\ \textsc{T-Int} \qquad \frac{}{\Gamma \vdash () : \textbf{unit}}\ \textsc{T-Unit}$$

$$\frac{\Gamma \vdash e_1 : \textbf{int} \quad \Gamma \vdash e_2 : \textbf{int}}{\Gamma \vdash e_1 + e_2 : \textbf{int}}\ \textsc{T-Add}$$

Simply-Typed Lambda Calculus

Static Semantics

$$\frac{}{\Gamma \vdash n : \textbf{int}}\ \text{T-INT} \qquad \frac{}{\Gamma \vdash () : \textbf{unit}}\ \text{T-UNIT}$$

$$\frac{\Gamma \vdash e_1 : \textbf{int} \quad \Gamma \vdash e_2 : \textbf{int}}{\Gamma \vdash e_1 + e_2 : \textbf{int}}\ \text{T-ADD}$$

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau}\ \text{T-VAR}$$

Simply-Typed Lambda Calculus

Static Semantics

$$\frac{}{\Gamma \vdash n : \textbf{int}} \ \textsc{T-Int} \qquad \frac{}{\Gamma \vdash () : \textbf{unit}} \ \textsc{T-Unit}$$

$$\frac{\Gamma \vdash e_1 : \textbf{int} \quad \Gamma \vdash e_2 : \textbf{int}}{\Gamma \vdash e_1 + e_2 : \textbf{int}} \ \textsc{T-Add}$$

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau} \ \textsc{T-Var} \qquad \frac{\Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x : \tau.\, e : \tau \to \tau'} \ \textsc{T-Abs}$$

Simply-Typed Lambda Calculus

Static Semantics

$$\frac{}{\Gamma \vdash n : \textbf{int}}\ \textsc{T-Int} \qquad \frac{}{\Gamma \vdash () : \textbf{unit}}\ \textsc{T-Unit}$$

$$\frac{\Gamma \vdash e_1 : \textbf{int} \quad \Gamma \vdash e_2 : \textbf{int}}{\Gamma \vdash e_1 + e_2 : \textbf{int}}\ \textsc{T-Add}$$

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau}\ \textsc{T-Var} \qquad \frac{\Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x : \tau.\, e : \tau \rightarrow \tau'}\ \textsc{T-Abs}$$

$$\frac{\Gamma \vdash e_1 : \tau \rightarrow \tau' \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1\ e_2 : \tau'}\ \textsc{T-App}$$

Properties

Theorem (Type soundness)

If $\vdash e : \tau$ *and* $e \to^* e'$ *and* $e' \not\to$ *then* e' *is a value and* $\vdash e' : \tau$.

Properties

Theorem (Type soundness)

If $\vdash e : \tau$ *and* $e \to^* e'$ *and* $e' \not\to$ *then* e' *is a value and* $\vdash e' : \tau$.

Lemma (Preservation)

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

Properties

Theorem (Type soundness)

If $\vdash e : \tau$ *and* $e \rightarrow^* e'$ *and* $e' \not\rightarrow$ *then* e' *is a value and* $\vdash e' : \tau$.

Lemma (Preservation)

If $\vdash e : \tau$ *and* $e \rightarrow e'$ *then* $\vdash e' : \tau$.

Lemma (Progress)

If $\vdash e : \tau$ *then either* e *is a value or there exists an* e' *such that* $e \rightarrow e'$.

Extra Lemmas for Preservation

Lemma (Substitution)

If $x : \tau' \vdash e : \tau$ *and* $\vdash v : \tau'$ *then* $\vdash e\{v/x\} : \tau$.

Lemma (Context)

If $\vdash E[e] : \tau$ *and* $\vdash e : \tau'$ *and* $\vdash e' : \tau'$ *then* $\vdash E[e'] : \tau$.

Extra Lemma for Progress

Lemma (Canonical Forms)

If $\vdash v : \tau$*, then*

1. *If* τ *is* ***int****, then v is a constant, i.e., some c.*
2. *If* τ *is* $\tau_1 \to \tau_2$*, then v is an abstraction, i.e.,* $\lambda x : \tau_1 .\, e$ *for some x and e.*