Andrew Appel
based on STLC chapter of Pierce's *ProgrammingLanguage Foundations*

SIMPLY TYPED LAMBDA CALCULUS

SYNTAX

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

$\tau ::= \text{Bool} \mid \tau_1 \to \tau_2$

Values

$v ::= \lambda x{:}\tau.e \mid \text{true} \mid \text{false}$

Substitution

$x[e/x] = e$

$y[e/x] = y \qquad y \neq x$

$(\lambda x{:}\tau.e_1)[e/x] = \lambda x{:}\tau.e_1$

$(\lambda y{:}\tau.e_1)[e/x] = \lambda y{:}\tau.(e_1[e/x]) \qquad y \neq x \text{ and } y \notin fv(e)$

$(\lambda y{:}\tau.e_1)[e/x] = \lambda z{:}\tau.(e_1[z/y][e/x] \qquad y \neq x \text{ and } z \notin fv(e, e_1)$

$\text{true}[e/x] = \text{true}$

$\text{false}[e/x] = \text{false}$

$\text{if } e_1 \text{ then } e_2 \text{ else } e_3\,[e/x] = \text{if } e_1[e/x] \text{ then } e_2[e/x] \text{ else } e_3[e/x]$

SMALL STEP OPERATIONAL SEMANTICS

$$\frac{\text{value } v_2}{(\lambda x{:}\tau.e_1)\,v_2 \mapsto e_1[v_2/x]} \qquad \frac{e_1 \mapsto e_1'}{e_1 e_2 \mapsto e_1' e_2} \qquad \frac{\text{value } v_1 \quad e_2 \mapsto e_2'}{v_1 e_2 \mapsto v_1 e_2'}$$

$$\frac{}{\text{if true then } e_2 \text{ else } e_3 \mapsto e_2} \qquad \frac{e_1 \mapsto e_1'}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \mapsto \text{if } e_1' \text{ then } e_2 \text{ else } e_3}$$

$$\frac{}{\text{if false then } e_2 \text{ else } e_3 \mapsto e_3}$$

TYPE SYSTEM

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau} \qquad \frac{\Gamma, x{:}\tau_1 \vdash e{:}\,\tau_2}{\Gamma \vdash (\lambda x{:}\tau_1.\,e) : \tau_2} \qquad \frac{\Gamma \vdash e_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1 e_2 : \tau_2}$$

$$\frac{}{\Gamma \vdash \text{true} : \text{Bool}} \qquad \frac{}{\Gamma \vdash \text{false} : \text{Bool}} \qquad \frac{\Gamma \vdash e_1 : \text{Bool} \quad \Gamma \vdash e_2 : \tau \quad \Gamma \vdash e_3 : \tau}{\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau}$$

FREE VARIABLES

$$\frac{}{x \in fv(x)} \qquad \frac{x \in fv(e_1)}{x \in fv(e_1 e_2)} \qquad \frac{x \in fv(e_2)}{x \in fv(e_1 e_2)} \qquad \frac{x \in fv(e) \quad x \neq y}{x \in fv(\lambda y{:}\tau.e)} \qquad \frac{x \in fv(e_1)}{x \in fv(\text{if } e_1 \text{ then} \ldots)}$$

etc.

STLC page 2

CANONICAL FORMS $\vdash e : Bool \rightarrow value\ e \rightarrow e \equiv true \vee e \equiv false$

$\vdash e : \tau_1 \rightarrow \tau_2 \rightarrow value\ e \rightarrow \exists x, e'.\ e \equiv \lambda x : \tau_1 . e'$

PROGRESS $\vdash e : \tau \rightarrow value\ e \vee \exists e'.\ e \mapsto e'$

by induction on $\vdash e : \tau$

CLOSED $e\ closed := fv(e) = \{\}$ (that is, $\exists x.\ x \in fv(e)$)

FREE IN CONTEXT: $x \in fv(e) \rightarrow \Gamma \vdash e : \tau \rightarrow \exists \tau'.\ \Gamma(x) = \tau'$

by induction on $x \in fv(e)$

CONTEXT INVARIANCE: $\Gamma \vdash e : \tau \rightarrow (\forall x.\ x \in fv(e) \rightarrow \Gamma(x) = \Gamma'(x)) \rightarrow \Gamma' \vdash e : \tau$

by induction on $\Gamma \vdash e : \tau$

SUBSTITUTION PRESERVES TYPING $\Gamma, x : \tau' \vdash e : \tau \rightarrow \vdash v : \tau' \rightarrow \Gamma \vdash e[v/x] : \tau$

by induction on e

PRESERVATION $\vdash e : \tau \rightarrow e \mapsto e' \rightarrow \vdash e' : \tau$

by induction on $\vdash e : \tau$

SAFETY $safe\ e :\equiv \forall e'.\ e \mapsto^* e' \rightarrow value\ e' \vee \exists e''.\ e' \mapsto e''$

SOUNDNESS (of type system) $\vdash e : \tau \rightarrow safe\ e$

Andrew Appel
based on "Sub" chapter of Pierce's *Programming Language Foundations*

SUBTYPING

$$\frac{\Gamma \vdash e_1 : \sigma \qquad \sigma <: \tau}{\Gamma \vdash e_1 : \tau}\ \text{Subsumption}$$

$$\cancel{\tau_1 <: \tau_2\ \tau} \qquad \frac{\tau_1 <: \tau_2 \qquad \tau_2 <: \tau_3}{\tau_1 <: \tau_3}\ \text{trans} \qquad \frac{}{\tau <: \tau}\ \text{refl}$$

$$\frac{\sigma_1 <: \tau_1 \qquad \sigma_2 <: \tau_2}{(\sigma_1 \times \sigma_2) <: (\tau_1 \times \tau_2)} \qquad \frac{\tau_1 <: \sigma_1 \qquad \sigma_2 <: \tau_2}{\sigma_1 \to \sigma_2 <: \tau_1 \to \tau_2}$$

$$\frac{n > m}{\{i_1 : \tau_1, \dots, i_m : \tau_m, i_{m+1} : \tau_{m+1}, \dots, i_n : \tau_n\} <: \{i_1 : \tau_1, \dots, i_m : \tau_m\}}\ \text{Width}$$

$$\frac{\sigma_1 <: \tau_1 \quad \cdots \quad \sigma_n <: \tau_n}{\{i_1 : \sigma_1, \dots, i_n : \sigma_n\} <: \{i_1 : \tau_1, \dots, i_n : \tau_n\}}\ \text{Depth}$$

$$\frac{\pi \text{ is a permutation on } 1..n}{\{i_1 : \sigma_1, \dots, i_n : \sigma_n\} <: \{i_{\pi(1)} : \sigma_{\pi(1)}, \dots\ i_{\pi(n)} : \sigma_{\pi(n)}\}}\ \text{Perm}$$

$$\frac{}{\tau <: \text{Top}}$$

INVERSION LEMMAS

$\tau <: \text{Bool} \longrightarrow \tau = \text{Bool}$

$\sigma <: \tau_1 \to \tau_2 \longrightarrow \exists \sigma_1, \sigma_2.\ \sigma = \sigma_1 \to \sigma_2 \wedge \tau_1 <: \sigma_1 \wedge \tau_2 <: \sigma_2$

CANONICAL FORMS

$\Gamma \vdash e : \tau_1 \to \tau_2 \to \text{value } e \to \exists x, \tau, e_2.\ e = \lambda x{:}\tau.\, e_2$

$\Gamma \vdash e : \text{Bool} \to \text{value } e \to e = \text{true} \vee e = \text{false}$

PROGRESS ... same as in STLC

INVERSION LEMMAS FOR SUBTYPING

$\Gamma \vdash \lambda x{:}\tau_1.\, e_2 : \tau \to \exists \tau_2.\ \tau_1 \to \tau_2 <: \tau \wedge \Gamma, x{:}\tau_1 \vdash e_2 : \tau_2$

$\Gamma \vdash x : \tau \to \exists \sigma.\ \Gamma(x) = \sigma \wedge \sigma <: \tau$

$\Gamma \vdash e_1\, e_2 : \tau \to \exists \tau_1.\ \Gamma \vdash e_1 : \tau_1 \to \tau_2 \wedge \Gamma \vdash e_2 : \tau_1$

$\Gamma \vdash \text{true} : \tau \to \text{Bool} <: \tau$

$\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau \to \Gamma \vdash e_1 : \text{Bool} \wedge \Gamma \vdash e_2 : \tau \wedge \Gamma \vdash e_3 : \tau$

SUBTYPING, PAGE 2

"Combination" inversion lemma:

$$\vdash (\lambda x : \sigma_1 . e_2) : \tau_1 \to \tau_2 \;\longrightarrow\; \tau_1 <: \sigma_1 \;\wedge\; x : \sigma_1 \vdash e_2 : \tau_2$$

CONTEXT INVARIANCE same as in STLC
SUBSTITUTION PRESERVES TYPING same as in STLC
PRESERVATION " " " "