

# SIMPLY TYPED LAMBDA CALCULUS

SYNTAX	$e ::= x \mid \lambda x:\tau. e \mid e_1 e_2 \mid \text{true} \mid \text{false} \mid \text{if } e_1, \text{ then } e_2 \text{ else } e_3$ $\tau ::= \text{Bool} \mid \tau_1 \rightarrow \tau_2$
Values	$v ::= \lambda x:\tau. e \mid \text{true} \mid \text{false}$
Substitution	$x[e/x] = e$ $y[e/x] = y \quad y \neq x$ $(\lambda x:\tau. e_1)[e/x] = \lambda x:\tau. e_1$ $(\lambda y:\tau. e_1)[e/x] = \lambda y:\tau. (e_1[e/x]) \quad y \neq x \text{ and } y \notin \text{fv}(e)$ $(\lambda y:\tau. e_1)[e/x] = \lambda z:\tau. (e_1[z/y][e/x]) \quad y \neq x \text{ and } z \notin \text{fv}(e, e_1)$ $\text{true}[e/x] = \text{true}$ $\text{false}[e/x] = \text{false}$ $\text{if } e_1, \text{ then } e_2 \text{ else } e_3 [e/x] = \text{if } e_1[e/x], \text{ then } e_2[e/x] \text{ else } e_3[e/x]$
SMALL STEP	$\frac{\text{value } v_2}{(\lambda x:\tau. e_1) v_2 \mapsto e_1[v_2/x]} \quad \frac{e_1 \mapsto e_1'}{e_1 e_2 \mapsto e_1' e_2} \quad \frac{\text{value } v_1 \quad e_2 \mapsto e_2'}{v_1 e_2 \mapsto v_1 e_2'}$
OPERATIONAL SEMANTICS	$\frac{}{\text{if true then } e_2 \text{ else } e_3 \mapsto e_2} \quad \frac{e_1 \mapsto e_1'}{\text{if } e_1, \text{ then } e_2 \text{ else } e_3 \mapsto \text{if } e_1', \text{ then } e_2 \text{ else } e_3}$
TYPE SYSTEM	$\frac{\Gamma(x) = \tau}{\Gamma \vdash x:\tau} \quad \frac{\Gamma, x:\tau_1 \vdash \tau_2}{\Gamma \vdash (\lambda x:\tau_1. e):\tau_2} \quad \frac{\Gamma \vdash e_1:\tau_1 \rightarrow \tau_2 \quad \Gamma \vdash e_2:\tau_1}{\Gamma \vdash e_1 e_2:\tau_2}$ $\frac{}{\Gamma \vdash \text{true}:\text{Bool}} \quad \frac{}{\Gamma \vdash \text{false}:\text{Bool}} \quad \frac{\Gamma \vdash e_1:\text{Bool} \quad \Gamma \vdash e_2:\tau \quad \Gamma \vdash e_3:\tau}{\Gamma \vdash \text{if } e_1, \text{ then } e_2 \text{ else } e_3:\tau}$
FREE VARIABLES	$\frac{}{x \in \text{fv}(x)} \quad \frac{x \in \text{fv}(e_1)}{x \in \text{fv}(e_1 e_2)} \quad \frac{x \in \text{fv}(e_2)}{x \in \text{fv}(e_1 e_2)} \quad \frac{x \in \text{fv}(e) \quad x \neq y}{x \in \text{fv}(\lambda y:\tau. e)} \quad \frac{x \in \text{fv}(e_1) \quad x \in \text{fv}(e_2) \quad x \in \text{fv}(e_3)}{x \in \text{fv}(\text{if } e_1, \text{ then } e_2 \text{ else } e_3)}$ <p style="text-align: right;">etc.</p>

# STLC page 2

CANONICAL  
FORMS

$$\vdash e: \text{Bool} \rightarrow \text{value } e \rightarrow e \equiv \text{true} \vee e \equiv \text{false}$$

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

PROGRESS

$$\vdash e: \tau \rightarrow \text{value } e \vee \exists e'. e \mapsto e'$$

by induction on  $\vdash e: \tau$

CLOSED

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

FREE IN CONTEXT:

$$x \in \text{fv}(e) \rightarrow \Gamma \vdash e: \tau \rightarrow \exists \tau'. \Gamma(x) = \tau'$$

by induction on  $x \in \text{fv}(e)$

CONTEXT INVARIANCE:

$$\Gamma \vdash e: \tau \rightarrow \left( \forall x. x \in \text{fv}(e) \rightarrow \Gamma(x) = \Gamma'(x) \right) \rightarrow \Gamma' \vdash e: \tau$$

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

$$\text{safe } e := \forall e'. e \mapsto^* e' \rightarrow \text{value } e' \vee \exists e''. e' \mapsto e''$$

SOUNDNESS  
(of type system)

$$\vdash e: \tau \rightarrow \text{safe } e$$

# SUBTYPING

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

$$\frac{\Gamma \vdash e_i : \sigma \quad \sigma <: \tau}{\Gamma \vdash e_i : \tau} \text{Subsumption}$$

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

$$\frac{\sigma_1 <: \tau_1 \quad \sigma_2 <: \tau_2}{(\sigma_1 \times \sigma_2) <: (\tau_1 \times \tau_2)} \quad \frac{\tau_1 <: \sigma_1 \quad \sigma_2 <: \tau_2}{\sigma_1 \rightarrow \sigma_2 <: \tau_1 \rightarrow \tau_2}$$

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

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

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

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

## INVERSION LEMMAS

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

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

## CANONICAL FORMS

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

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

PROGRESS ... same as in STLC

## INVERSION LEMMAS FOR SUBTYPING

$$\Gamma \vdash \lambda x : \tau. e_2 : \tau \longrightarrow \exists \tau_2. \tau_1 \rightarrow \tau_2 <: \tau \wedge \Gamma, x : \tau_1 \vdash e_2 : \tau_2$$

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

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

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

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

# SUBTYPING, PAGE 2

"Combination" inversion lemma:

$$\vdash (\lambda x:\sigma_1. e_2) : \tau_1 \rightarrow \tau_2 \rightarrow \tau_1 <: \sigma_1 \wedge x:\sigma_1 \vdash e_2 : \tau_2$$

CONTEXT INVARIANCE same as in STLC

SUBSTITUTION PRESERVES TYPING same as in STLC

PRESERVATION

" " " "