BL: Booleans with Let-Binding
CPSC 509: Programming Language Principles

Ronald Garcia*

14 October 2015

Syntax

$$t \in \textsc{Term}, \quad b \in \mathbb{B}, \quad x \in \textsc{Var},$$
$$t ::= \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ t\ \mathsf{then}\ t\ \mathsf{else}\ t \mid x \mid \mathsf{let}\ x = t\ \mathsf{in}\ t$$
$$b ::= \mathsf{true} \mid \mathsf{false}$$

Semantics

$$v \in \textsc{Value}, \quad r \in \textsc{Redex}, \quad E \in \textsc{ECtxt}, \quad p \in \textsc{Pgm} = \{\, t \in \textsc{Term} \mid FV(t) = \emptyset \,\}, \quad o \in \textsc{Obs} = \mathbb{B}$$
$$v ::= b$$
$$E ::= \square \mid E[\mathsf{if}\ \square\ \mathsf{then}\ t\ \mathsf{else}\ t] \mid E[\mathsf{let}\ x = \square\ \mathsf{in}\ t]$$
$$r ::= \mathsf{if}\ v\ \mathsf{then}\ t\ \mathsf{else}\ t \mid \mathsf{let}\ x = v\ \mathsf{in}\ t$$

Free Variables

$FV : \textsc{Term} \to \mathcal{P}(\textsc{Var})$

$$FV(b) = \emptyset$$
$$FV(\mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3) = FV(t_1) \cup FV(t_2) \cup FV(t_3)$$
$$FV(x) = \{\, x \,\}$$
$$FV(\mathsf{let}\ x = t_1\ \mathsf{in}\ t_2) = FV(t_1) \cup (FV(t_2) \setminus \{\, x \,\})$$

Naïve Substitution

$[x \Rrightarrow t] : \textsc{Term} \to \textsc{Term}$

$$[x \Rrightarrow t]b = b$$
$$[x \Rrightarrow t]x = t$$
$$[x \Rrightarrow t]x_0 = x_0 \text{ if } x \neq x_0$$
$$[x \Rrightarrow t](\mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3) = \mathsf{if}\ [x \Rrightarrow t]t_1\ \mathsf{then}\ [x \Rrightarrow t]t_2\ \mathsf{else}\ [x \Rrightarrow t]t_3$$
$$[x \Rrightarrow t](\mathsf{let}\ x = t_1\ \mathsf{in}\ t_2) = \mathsf{let}\ x = [x \Rrightarrow t]t_1\ \mathsf{in}\ t_2$$
$$[x \Rrightarrow t](\mathsf{let}\ x_0 = t_1\ \mathsf{in}\ t_2) = \mathsf{let}\ x_0 = [x \Rrightarrow t]t_1\ \mathsf{in}\ [x \Rrightarrow t]t_2 \text{ if } x \neq x_0$$

$\boxed{\rightsquigarrow\ \subseteq \textsc{Redex} \times \textsc{Term}}$ **Notions of Reduction**

$$\mathsf{if\ true\ then}\ t_2\ \mathsf{else}\ t_3 \rightsquigarrow t_2$$
$$\mathsf{if\ false\ then}\ t_2\ \mathsf{else}\ t_3 \rightsquigarrow t_3$$
$$\mathsf{let}\ x = v\ \mathsf{in}\ t \rightsquigarrow [x \Rrightarrow v]t$$

$\boxed{\longrightarrow\ \subseteq \textsc{Term} \times \textsc{Term}}$ **Single-Step Reduction**

$$\frac{t \rightsquigarrow t'}{E[t] \longrightarrow E[t']}$$

$$eval_{BL} : \textsc{Pgm} \to \textsc{Obs}$$
$$eval_{BL}(t) = b \text{ if } t \longrightarrow^* b$$

Proposition 1 (Principle of Definition by Primitive Recursion on $t \in \textsc{Term}$)**.** *Let S be a set and $s_t, s_f \in S$ be two elements,*

$$H_{if} : \textsc{Term} \times \textsc{Term} \times \textsc{Term} \times S \times S \times S \to S$$
$$H_{let} : \textsc{Var} \times \textsc{Term} \times \textsc{Term} \times S \times S \to S$$
$$H_{var} : \textsc{Var} \to S$$

be functions.

Then there exists a unique function

$$F : \textsc{Term} \to S$$

such that

1. $F(\textit{true}) = s_t$;
2. $F(\textit{false}) = s_f$;
3. $F(x) = H_{var}(x)$.
4. $F(\textit{if}\ t_1\ \textit{then}\ t_2\ \textit{else}\ t_3) = H_{if}(t_1, t_2, t_3, F(t_1), F(t_2), F(t_3))$;
5. $F(\textit{let}\ x = t_1\ \textit{in}\ t_2) = H_{let}(x, t_1, t_2, F(t_1), F(t_2))$.