COMS22202: 2015/16

Language Engineering

Dr Oliver Ray
(csxor@Bristol.ac.uk)

Department of Computer Science
University of Bristol

Thursday 18th February, 2016

Boolean Expressions: p14

Syntax: $b ::= \mathbf{true} \mid \mathbf{false} \mid a_1 = a_2 \mid a_1 \le a_2 \mid \neg b \mid b_1 \wedge b_2$

Semantics: $\mathcal{B} : \mathbf{Bexp} \to (\mathbf{State} \to \mathbf{T})$ where $\mathbf{T} = \{\mathbf{tt}, \mathbf{ff}\}$

$$\mathcal{B}[[\,\mathbf{true}\,]]\,s = \mathbf{tt}$$

$$\mathcal{B}[[\,\mathbf{false}\,]]\,s = \mathbf{ff}$$

$$\mathcal{B}[[\,a_1 = a_2\,]]\,s = \begin{cases} \mathbf{tt} & \text{if } \mathcal{A}[[a_1]]s = \mathcal{A}[[a_2]]s \\ \mathbf{ff} & \text{if } \mathcal{A}[[a_1]]s \neq \mathcal{A}[[a_2]]s \end{cases}$$

$$\mathcal{B}[[\,a_1 \le a_2\,]]\,s = \begin{cases} \mathbf{tt} & \text{if } \mathcal{A}[[a_1]]s \le \mathcal{A}[[a_2]]s \\ \mathbf{ff} & \text{if } \mathcal{A}[[a_1]]s > \mathcal{A}[[a_2]]s \end{cases}$$

$$\mathcal{B}[[\,\neg b\,]]\,s = \begin{cases} \mathbf{tt} & \text{if } \mathcal{B}[[b]]s = \mathbf{ff} \\ \mathbf{ff} & \text{if } \mathcal{B}[[b]]s = \mathbf{tt} \end{cases}$$

$$\mathcal{B}[[\,b_1 \wedge b_2\,]]\,s = \begin{cases} \mathbf{tt} & \text{if } \mathcal{B}[[b_1]]s = \mathbf{tt} = \mathcal{B}[[b_2]]s \\ \mathbf{ff} & \text{otherwise} \end{cases}$$

(Free) Variables: p15-16

$$FV : \mathrm{Aexp} \to \wp(\mathrm{Var})$$

$$
\begin{aligned}
FV(\, n \,) &= \emptyset \\
FV(\, x \,) &= \{\, x \,\} \\
FV(\, a_1 + a_2 \,) &= FV(\, a_1 \,) \cup FV(\, a_2 \,) \\
FV(\, a_1 * a_2 \,) &= FV(\, a_1 \,) \cup FV(\, a_2 \,) \\
FV(\, a_1 - a_2 \,) &= FV(\, a_1 \,) \cup FV(\, a_2 \,)
\end{aligned}
$$

$$FV : \mathrm{Bexp} \to \wp(\mathrm{Var})$$

$$
\begin{aligned}
FV(\, \mathrm{true} \,) &= \emptyset \\
FV(\, \mathrm{false} \,) &= \emptyset \\
FV(\, a_1 = a_2 \,) &= FV(\, a_1 \,) \cup FV(\, a_2 \,) \\
FV(\, a_1 \leq a_2 \,) &= FV(\, a_1 \,) \cup FV(\, a_2 \,) \\
FV(\, \neg b \,) &= FV(\, b \,) \\
FV(\, b_1 \wedge b_2 \,) &= FV(\, b_1 \,) \cup FV(\, b_2 \,)
\end{aligned}
$$

(Semantic) Equivalence: p14

Two arithmetic expressions a_1 and a_2 are equivalent (written $a_1 \equiv a_2$)

iff $\mathcal{A}[[\, a_1 \,]]\, s = \mathcal{A}[[\, a_2 \,]]\, s$ for all states s

e.g. $(x+y) \equiv (y+x)$

Two Boolean expressions b_1 and b_2 are equivalent (written $b_1 \equiv b_2$)

iff $\mathcal{B}[[\, b_1 \,]]\, s = \mathcal{B}[[\, b_2 \,]]\, s$ for all states s

e.g. $(a \leq b \wedge \text{true}) \equiv (a \leq b)$

(Arth.) Variable Substitutions: p16

$\text{Aexp} \,[\, \text{Var} \mapsto \text{Aexp} \,] \rightarrow \text{Aexp}$

$$n\,[y \mapsto a] = n$$

$$x\,[y \mapsto a] = \begin{cases} a & \text{if } x = y \\ x & \text{if } x \neq y \end{cases}$$

$$(a_1 + a_2)\,[y \mapsto a] = (\, a_1\,[y \mapsto a]\,) + (\, a_2\,[y \mapsto a]\,)$$

$$(a_1 * a_2)\,[y \mapsto a] = (\, a_1\,[y \mapsto a]\,) * (\, a_2\,[y \mapsto a]\,)$$

$$(a_1 - a_2)\,[y \mapsto a] = (\, a_1\,[y \mapsto a]\,) - (\, a_2\,[y \mapsto a]\,)$$

e.g. $(1+(x*2)-y)\,[x \mapsto (x+y+1)] = (1+((x+y+1)*2)-y)$

State Updates: p16

$\text{State}\ [\ \text{Var} \mapsto \mathbb{Z}\ ] \to \text{State}$

$$s\,[y \mapsto v]\,x \quad = \quad \begin{cases} v & \text{if } x = y \\ s\,x & \text{if } x \neq y \end{cases}$$

e.g. $s_{x=1,\ y=2,\ z=3}\ [y \mapsto 0] \ = \ s'_{x=1,\ y=0,\ z=3}$

(Least) Fixpoints: cf. p 95

- The notion of the least fixpoint of a function is heavily studied in mathematics (domains, complete partial orders, lambda calculus, etc.) and is closely related to the denotational semantics of loops

- For any unary operator $f : X \to X$ on some domain X with a partial order $\leq$:
  - a fixpoint of f is any element $x \in X$ such that $f(x)=x$

    fix(f) denotes the set of all such fixpoints

    i.e. $fix(f) = \{x \in X \mid f(x)=x\}$

  - A least fixpoint of f is a fixpoint of f that is least with respect to the $\leq$ order

    lfp(f) denotes the least fixpoint (which is unique if it exists)

    i.e. $lfp(f) = x$ iff $x \in fix(f)$ and $x \leq y$ for all $y \in fix(f)$

- Recall that a partial order is any relation that is reflexive, transitive and antisymmetric (cf. p.95)

Fixpoint Examples

Q) Find the fixpoints and least fixpoints of the following functions on reals R:

- square = $\lambda x . x*x$
- half = $\lambda x . x/2$
- inc = $\lambda x . x+1$
- id = $\lambda x . x$

A) The fixpoints and least fixpoints are as follows:

- fix(square) = {0,1} lfp(square) = 0
- fix(half) = {0} lfp(half) = 0
- fix(inc) = {} lfp(inc) = undefined
- fix(id) = R lfp(id) = undefined

Conditional Functions: cf. p 87

- The notion of a conditional function is closely related to the denotational semantics of conditionals (and loops!)

- We want to use one of two functions c or d to map some inputs x to some outputs y; and we have a Boolean test b for that will determine which function to apply in each case x:

- cond : $(X \to T) \times (X \to Y) \times (X \to Y) \;\; \to \;\; (X \to Y)$

$$\mathrm{cond}(b, c, d)\, x = \begin{cases} c(x) & \text{if } b(x) = tt \\ d(x) & \text{otherwise} \end{cases}$$

Function Composition: p 214

- Let f ang g be any two functions such that $f : Y \to Z$ and $g : X \to Y$

- Then the composition of f and g, denoted $f \circ g$, is the function $(f \circ g) : X \to Z$

  such that $(f \circ g)\ x = f\ (\ g(x)\ )$

- For example $(id \circ (inc \circ (half \circ square))) = \lambda x\ .\ (\ 1 + x^2/2\ )$