Satisfaction & Validity; Syntax of Programs

CS 536 Lecture 5, Wed Jan 25, 2012

5.1 Satisfaction & Validity, continued

A. Why

• Satisfaction and validity are what it means for predicates to be "true".

B. Outcomes

At the end of today, you should

• Be able to check predicates for satisfaction in a state and validity (in all states).

C. Questions (repeated from Activity 4)

1. Say *u* and *v* are variables (possibly the same) and α and β are values (possibly equal). When is $\sigma[u \mapsto \alpha][v \mapsto \beta] = \sigma[v \mapsto \beta][u \mapsto \alpha]$?
2. Let $\sigma = \{x = 3, y = 4, b = (7, 9, 12, 16)\}$.
   a. Does $\sigma \models \exists\, k \,.\, b[k] < b[k+1]$? If so, what witness did you use? Are there other possible witnesses?

Yes, e.g. k = 0 works: $\sigma[k \mapsto 0] \models b[k] < b[k+1]$. Can also use witnesses 1 and 2

   b. What satisfactions would we have to check in order to see if $\sigma \models \forall\, j \,.\, 0 \le j \le 2 \rightarrow b[j] < b[j+1]$?

3. Does $\sigma \models \exists\, z \,.\, b[z] < b[z+1]$? Does it matter that z is undefined in σ?
4. Does $\sigma \models \exists\, x \,.\, b[x] < b[x+1]$? Does it matter that σ(x) is defined or what value it has?
5. Is $\models \forall\, x \,.\, \exists\, y \,.\, y > x^2$? Does it matter [for purposes of satisfaction] that the witness for y would have to depend on the value of x?

$\models \forall\, x \,.\, \exists\, y \,.\, y > x^2$

if for every state σ, $\sigma \models \forall\, x \,.\, \exists\, y \,.\, y > x^2$

if for every σ and integer α, $\sigma[x \mapsto \alpha] \models \exists\, y \,.\, y > x^2$

if for every σ and α there is a β such that $\sigma[x \mapsto \alpha][y \mapsto \beta] \models y > x^2$

Note $\beta = \alpha^2 + 1$ works: $\sigma[x \mapsto \alpha][y \mapsto \alpha^2+1] \models y > x^2$

6. Is $\models \exists\, x \,.\, \forall\, y \,.\, 0 \le y \wedge y \le 20 \rightarrow x \ge y^2$?

Yes, we can use any value ≥ 400 for x

Activity 5.2: Our Simple Programming Language

A. Why

- Our simple programming language is a model for the kind of constructs seen in actual languages.

B. Outcomes

At the end of today, you should

- Be able to write and recognize simple programs in our programming language

C. Questions

1. In our simple language, `if x < 0 then x := 0 fi` is equivalent to what other statement?
2. Translate the following C/C++/Java program fragments into our simple programming language:
   a. `++x; if (x < y) { x = y = y+1; }`
   b. `y = z * ++x; z = z+x;`
   c. `y = z * x++; z = z+x;`
   d. `x = z = 0; while (x++ < n) z = z+x;`
   e. `z = 1; for (x = n ; x >= 1 ; --x) z = z * x;`

D. Solutions

1. `if x < 0 then x := 0 else skip fi`
2. a. `x:=x+1; if x < y then y:=y+1; x:=y fi`
   b. `x:=x+1; y:=z*x; z:=z+x`
   c. `y:=z*x; x:=x+1; z:=z+x`
   d. `z:=0; x:=z; while x < n do x:=x+1; z:=z+x od; x:=x+1`
   e. `z:=1; x:=n; while x >= 1 do z:=z*x; x:=x-1 od`

Activity 5.3 Program Semantics

A. Why

- The meaning of a program is that it transforms states.

B. Outcomes

At the end of this activity, you should

- Be able to calculate the meaning of a program in a state.

C. Questions

1. What is $M(\texttt{x := x+1}, \{x = 5\})$?
2. What is $M(\texttt{y := 2*x}, \{x = 6\})$?
3. What is $M(\texttt{x := x+1}, \sigma)$?
4. What is $M(\texttt{x := x+1; y := 2*x}, \{x = 5\})$?

D. Solutions

1. $M(\texttt{x := x+1}, \{x = 5\}) = \{x = 6\}$
2. $M(\texttt{y := 2*x}, \{x = 6\}) = \{x = 6, y = 12\}$
3. $M(\texttt{x := x+1}, \sigma) = \sigma[x \mapsto \sigma(x)+1]$
4. $M(\texttt{x := x+1; y := 2*x}, \{x = 5\})$
   $= M(\texttt{y := 2*x}, M(\texttt{x := x+1}, \{x = 5\}))$
   $= M(\texttt{y := 2*x}, \{x = 6\})$
   $= \{x = 6, y = 12\}$

