Solution: Types, Expressions, States, Quantified Predicates

CS 536: Science of Programming, Spring 2023

Due Thu Feb 2, 11:59 pm

Class 3: Types, Expressions, and Arrays

1. $b[0]+b[2][3]$ is an illegal expression ($b[0]$ is an array)

2. $\{u=\underline{(4)}, w=u[1], \; r=\underline{one}, \; s=\underline{four}, \; t=r+s\}$ is ill-formed ($u[1]$ is not semantic)

3. (Array notations) $\sigma=\{x=2, b=\beta\}$ where $\beta=(7,12,3)$ is equivalent to:
   a. $\sigma=\{x=2, b=\{(0,7),(1,12),(2,3)\}$
   b. $\sigma=\{x=2, b[0]=7, \ldots, b[1]=12, b[2]=3\}$

4. (State for expression?) For expression $c*b[b[k]]$,
   a. $\{k=0, \; b=\underline{(3,6,1,4)}, c=\underline{2}\}$ State is well-formed, proper; expression evaluates to $2*b[b[0]]=2*b[3]=2*4=8$.
   b. $\{k=\underline{0}, \; b=\underline{3}\}$ State is well-formed but improper (b isn't an array, c is missing)

Class 4: State Updates, Satisfaction of Quantified Predicates

5. State updates with $\sigma=\{x=\underline{2}, y=\underline{4}\}$
   a. $\sigma[x\mapsto\underline{1}]=\{x=\underline{1}, y=\underline{4}\}$ is not equal to $\sigma\cup\{(x,\underline{1})\}=\{x=\underline{2}, x=\underline{1}, y=\underline{4}\}$.
   b. $\sigma[v\mapsto\underline{2}]=\sigma\cup\{(v,\underline{2})\}=\{x=\underline{1}, y=\underline{4}, v=\underline{2}\}$.

6. Complicated updates with $\sigma=\{x=\underline{2}, y=\underline{4}\}$
   a. $\sigma(y)=4$ and $\sigma(x)=2$, so $\sigma[x\mapsto\sigma(y)][y\mapsto\sigma(x)]=\sigma[x\mapsto 4][y\mapsto 2]=\{x=4, y=2\}$.
   b. Since $x\not\equiv y$, we don't need to know what $\tau(y)$ is to calculate
      $\tau(x) \; = \sigma[x\mapsto\underline{3}] \; [y\mapsto ???](x)=3$
      So $\tau=\sigma[x\mapsto\underline{3}][y\mapsto\underline{\tau(x)*4}]=\tau=\sigma[x\mapsto\underline{3}][y\mapsto\underline{3*4}]=\{x=\underline{3}, y=\underline{12}\}$

7. (Justification of nested quantified predicates)
   a. State $\{x=\underline{0}, b=\underline{(5,3,6)}\}$ does not satisfy $(\forall x.\forall k.0<k<3\wedge x<b[k])$ because there are values for k outside the range $0<k<3$ ($k=0$ or $k=3$ are examples). (By the way, if we replaced the $\wedge$ with $\rightarrow$, we'd get the equivalent of $\forall x.\forall 0<k<3.x<b[k]$ and there would be satisfaction.)

b. State $\{b = (2, 5, 4, 8)\}$ does satisfy $(\exists m.\, 0 \le m < 4 \to b[m] < 2)$ because there exist m that are outside the range $0 \le m < 4$ (such as -1 and 4), which makes the implication true. (By the way, if we replaced the $\to$ with $\wedge$, we'd get the equivalent of $\exists\, 0 \le m < 4.\, b[m] < 2$, and there would be satisfaction.)

8. (Invalidity)

$(\exists x \in V.\, (\exists y \in U.\, P(x, y)) \wedge (\forall z \in W.\, Q(x, z)))$ is invalid if

(for some state σ, every value $\alpha \in V$,

(for every $\beta \in U$, $\{x = \alpha, y = \beta\} \models \neg P(x, y))$

or $($ for some $\gamma \in W$, $\{x = \alpha, y = \beta, z = \gamma\} \not\models Q(x, y, z))\ )$.

9. (Predicate function $Unique(b, x, m)$ on arrays)

Basically, we want to check that

$b[x] \ne b[x+1],\ b[x+2],\ \ldots,\ b[x+m-1]$ and

$b[x+1] \ne b[x+2],\ b[x+3],\ \ldots,\ b[x+m-1]$ and

$b[x+i] \ne b[x+i+1],\ b[x+i+2],\ \ldots,\ b[x+m-1]$ and

… and

$b[x+m-2] \ne b[x+m-1]$

The definition below has index i range between 0 and $m-2$ and index j range between $i+1$ and $m-1$. A similar definition has an index, call it u, range between x and $x+m-2$, and another index range between $u+1$ and $x+m-1$.

$Unique(b, x, m) \equiv$

$0 \le x < length(b) \wedge 0 \le m < length(b) - x$

$\wedge\ \forall i.\, 0 \le i < m-1 \to \forall j.\, i+1 \le j < x+m \to b[i] \ne b[j]$