Types, Expressions, States, Quantified Predicates

CS 536: Science of Programming, Fall 2019

Due Mon Sep 16, 11:59 pm

A. Formatting and Submitting Your Work

- You don't have to use a word processor to write out your answers: Feel free to convert logical symbols into ASCII text: For ∧, ∨, →, ¬, ∀, ∃,write `and`, `or`, `->`, `!`, `all`, and `exist`. For ⇒, ⇔, ≡, and ≢, write `=>`, `<=>`, `==`, and `!==`. Quantified variables range over ℤ unless otherwise specified.

B. Problems [50 points total]

Lecture 3: Types, Expressions, and Arrays

1. (6 = 3 * 2 points) For each of the following, is the expression legal or illegal according to the syntax we're using. If illegal, why? If legal, what is the type of the resulting expression?
   a. `( x < y ? x : F )` // assume < works on integers, not booleans
   b. `b[0] + b[1][1]` // assume `b` is 2-dimensional
   c. `match(b1, b2, n)` // match asks if the first n elements of `b1` match the first n elements of `b2`
      // (Assume `b1` and `b2` are one-dimensional.)

2. (6 = 3 * 2 points) For each of the following are well-formed states? For the ones that aren't, why?
   a. { `x` = (2), `y` = 4 }
   b. { `u` = (3, 4), `v` = 0, `w` = `u[1]` }
   c. { `r` = *one*, `s` = *four*, `t` = `r` + `s` }

3. (4 = 2 * 2 points) Let σ = {`x` = 2, `b` = β } where β = (*five*, *two plus two*, 6).
   a. Rewrite σ giving the value of `b` as a set of ordered pairs.
   b. Rewrite σ giving the value of `b` as separate bindings for b[0], b[1], etc.

4. (6 = 3 * 2 points) Let φ ≡ `x` = `y`*`z` ∧ `y` = 3*`z` ∧ `z` = `b[0]` + `b[2]` ∧ 3 < `b[1]` < `b[2]` < 6. Complete the definition of σ = {`x` = ____, `y` = ____, `z` = 5, `b` = _____ } so that σ ⊨ φ.

5. (6 = 3 * 2 points) Take the expression `0 * b[b[j]]`. For each state below, is it well-formed and proper for the expression? And if so, does the expression terminate correctly (and with what result)? If not, why?
   a. {`j` = 0, `b` = (3, 2, 5, 4), `c` = (3), `d` = 8}
   b. ∅
   c. {`j` = 0, `b` = 0}

Lecture 4: State Updates, Satisfaction of Quantified Predicates

6. (4 = 2 * 2 points) Let $\sigma = \{\mathtt{x} = 2, \mathtt{y} = 4, \mathtt{b} = (-1, 0, 4, 2)\}$.
   a. Is there a difference between $\sigma[\mathtt{z} \mapsto 1]$ and $\sigma \cup \{(\mathtt{z}, 1)\}$? Justify your answer (very briefly).
   b. Repeat, on $\sigma[\mathtt{x} \mapsto 5]$ and $\sigma \cup \{(\mathtt{x}, 5)\}$?

7. (6 = 2 * 3 points) Recall how satisfaction of quantified predicates and state updates are defined.
   a. Does $\{\mathtt{x} = 4,\ \mathtt{y} = 6,\ \mathtt{b} = (4,2,8)\} \vDash (\exists \mathtt{x}.\exists \mathtt{j}.\ \mathtt{b}[\mathtt{j}] < \mathtt{x} < \mathtt{y})$? If not, why?
   b. Does $\{\mathtt{x} = 0,\ \mathtt{y} = 7,\ \mathtt{b} = (4,2,8)\} \vDash (\forall \mathtt{x}.\forall \mathtt{k}.0 < \mathtt{k} < 3 \rightarrow \mathtt{x} < \mathtt{b}[\mathtt{k}])$? If not, why?

8. (6 = 2 * 3 points) In English, explain briefly when each of the following holds.
   a. $\nvDash (\forall\, \mathtt{x} \in U\ .\ (\exists\, \mathtt{y} \in V\ .\ P(\mathtt{x}, \mathtt{y})))$
   b. $\nvDash \forall\, \mathtt{y}\ .((\exists\, \mathtt{x} \in U.\ P(\mathtt{x}, \mathtt{y})) \rightarrow (\exists\, \mathtt{y} \in U.\ Q(\mathtt{x}, \mathtt{y})))$

9. (6 points) Write a definition for a predicate function $\mathtt{P}(\mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{s}, \mathtt{t}) \equiv \ldots$ such that every element in $\mathtt{b}[\mathtt{c}]$, $\mathtt{b}[\mathtt{c}+1]$, …, $\mathtt{b}[\mathtt{d}-1]$ is less than some element in $\mathtt{b}[\mathtt{s}]$, $\mathtt{b}[\mathtt{s}+1]$, …, $\mathtt{b}[\mathtt{t}-1]$. Also make sure that all four of $\mathtt{c}$, $\mathtt{d}$, $\mathtt{s}$, $\mathtt{t}$ are legal indexes for $\mathtt{b}$, which is of length $\mathtt{n}$. Feel free to write helper predicate functions if it makes your life easier.

