CS536 Assignment 2

Due: Feb. 13th, 2024

(Ethics: Any behavior on any homework or exam that could be considered copying or cheating will result in an immediate zero on the assignment for all parties involved and will be reported to academichonesty@iit.edu. See the IIT Code of Academic Honesty, https://web.iit.edu/student-affairs/handbook/fine-print/code-academic-honesty)

- ❖ Do not use or consider a nondeterministic statement in any question of this assignment.
- ❖ Quantified variables have domain $\mathbb{Z}$ unless specified.
- ❖ Different variables in the same question are not syntactically equal unless specified.

1. In computational theory, we have a lemma called Pumping Lemma and it says: "If A is a regular language, then there is a positive integer p where if s is any string in A of length at least p, then s can be divided into three pieces, $s = xyz$, satisfying the following three conditions: 1) for each $i \geq 0, xy^iz \in A$ 2) $|y| > 0$ and 3) $|xy| \leq p$".

   This is a long lemma, and we can use notations and predicate functions to analyze it. Let's use R to represent the set of all regular languages, use $size(x)$ to represent the length of string x, and let $D(s,p) \equiv$ "s can be divided into three pieces, $s = xyz$, satisfying the following conditions: 1) for each $i \geq 0, xy^iz \in A$ 2) $|y| > 0$ and 3) $|xy| \leq p$". Then the above lemma becomes:

$$A \in R \Rightarrow \exists p > 0. \forall s \in A. size(s) \geq p \rightarrow D(s,p)$$

   Answer the following questions.
   a. Given a language A, can we use the above lemma to prove that $A \in R$? Why?
   b. Given a language A, what can be used as a witness / or witnesses to prove that $A \notin R$? (The witness should be: for each positive integer p, find an $s \in A$ with some property that can be described using notations and functions defined above.)

2. Let $e \equiv \textbf{if } x \geq 0 \textbf{ then } b[0] \textbf{ else } a[1][3] \textbf{ fi}$. Answer the following questions.
   a. If $a \equiv b$ (in other words, a and b are the same array), then is e a legal expression? Why?
   b. Let $\sigma = \big(x = -1,\ b = (2),\ a = (\alpha,\ \beta)\big)$, where $\alpha = (2,4)$ and $\beta = (0,3)$. Is σ proper for e? Does it satisfy e? Why?

3. Let u and v some be variables and α and β be some semantic values (u and v might be the same variable, α and β might be the same value). When is $\sigma[u \mapsto \alpha][v \mapsto \beta] = \sigma[v \mapsto \beta][u \mapsto \alpha]$ and when is $\sigma[u \mapsto \alpha][v \mapsto \beta] \equiv \sigma[v \mapsto \beta][u \mapsto \alpha]$? Discuss the four different cases depending on whether $u \equiv v$ and whether $\alpha = \beta$.

		$\sigma[u \mapsto \alpha][v \mapsto \beta] = \sigma[v \mapsto \beta][u \mapsto \alpha]$?	$\sigma[u \mapsto \alpha][v \mapsto \beta] \equiv \sigma[v \mapsto \beta][u \mapsto \alpha]$?
$u \equiv v$	$\alpha = \beta$		
$u \equiv v$	$\alpha \neq \beta$		
$u \not\equiv v$	$\alpha = \beta$		
$u \not\equiv v$	$\alpha \neq \beta$		

4. Consider state $\sigma = \{x = 2, y = 4\}$. Answer the following questions and show your work.
   a. What is $\sigma[x \mapsto \sigma(y)][y \mapsto \sigma(x)]$?
   b. Let $\tau = \sigma[x \mapsto 3]$, and $\gamma = \tau[y \mapsto \tau(x) * 4]$. What is γ?

5. Answer the following questions and briefly justify your answers.
   a. Does $\{x = 1, b = (5, 3, 6)\}$ satisfy $\forall x. \forall 0 \le k < 3. x < b[k]$?
   b. Does $\{b = (2, 5, 4, 8)\}$ satisfy $\exists m. 0 \le m < 4 \wedge b[m] < 2$?

6. Translate the programs below into our programming language.
   a. $m = 0; x = 0; y = 1;$ **while** $(m++< n)\{y = ++x;\ y \mathrel{*=} x;\}\, m = m * m;$
   b. $m = n;\ p = 1; y = 1;$ **while** $(--m < n)\,\{\, p = p * (y++); \}$

7. Let $S \equiv$ **if** $x > 0$ **then** $x := x + 1$ **else** $y := -2 * x$ **fi** and let $W \equiv$ **while** $x > y$ **do** S **od**.
   a. Evaluate $\langle W, \sigma \rangle$ where $\sigma \vDash y < x \le 0$ to completion. Do not use $\to^*$ or $\to^n$ in your solution for this question.
   b. Evaluate $\langle W, \sigma \rangle$ where $\sigma \vDash x > 0 \wedge y \le 0$ to completion. You can use $\to^*$ and/or $\to^n$ in your solution for this question.

8. Let $W \equiv$ **while** $x > 0$ **do** S **od**, where $S \equiv$ **if** $x < y$ **then** $x := y/x$ **else** $x := x - 1; y := b[y]$ **fi**. Answer the following questions.
   a. Calculate $M(S, \sigma)$ where $\sigma(x) = -2$ and $\sigma(y) = -1$.
   b. Calculate $M(W, \sigma)$ where $\sigma = \{x = 1,\ y = 2,\ b = (4, 2, 0)\}$.
   c. Calculate $M(W, \sigma)$ where $\sigma = \{x = 2,\ y = 2,\ b = (0, 1, 2)\}$.
   d. Calculate $M(W, \sigma)$ where $\sigma = \{x = 8,\ y = 2,\ b = (4, 2, 0)\}$.
   e. Is there a state σ that makes $M(W, \sigma) = \{\perp_e\}$ because of the "division by zero" error?

9. Let $S \equiv x := sqrt(x)\ /\ b[y]$ and let $\sigma = \{b = (3, 0, -2, 4), x = \alpha, y = \beta\}$. Find all possible integer values for α and β such that $M(S, \sigma) = \{\perp_e\}$.

10. In lecture 5, without the consideration of $\perp$, we have the following conclusions:

$$\sigma \nvDash \exists x \in S. p \Leftrightarrow \sigma \vDash \forall x \in S. \neg p$$
$$\sigma \nvDash \forall x \in S. p \Leftrightarrow \sigma \vDash \exists x \in S. \neg p$$

We used "$\sigma \nvDash p \Leftrightarrow \sigma \vDash \neg p$" during the proof to get the above conclusions, so they are not totally correct anymore with the consideration of $\perp$. Let's create some conclusions without replacing $\sigma \nvDash p$ by $\sigma \vDash \neg p$. Remind that, the following definitions about satisfaction of quantified predicates are still correct (I rephrased them here):

- $\sigma \vDash \exists x \in S. p$ means there exists $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha] \vDash p$.
- $\sigma \vDash \forall x \in S. p$ means for every value $\alpha \in S$, we have $\sigma[x \mapsto \alpha] \vDash p$.

Fill in the blanks as described so that each of the following sentences is correct.

- Fill *type 1* with word "some", "every" or "this".
- Fill *type 2* with "$\vDash$" or "$\nvDash$".

a. $\sigma \vDash \exists x \in S. p$ means for *type 1* state σ and for *type 1* $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha]$ *type 2* p.
b. $\sigma \vDash \forall x \in S. p$ means for *type 1* state σ and for *type 1* $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha]$ *type 2* p.
c. $\sigma \nvDash \exists x \in S. p$ means for *type 1* state σ and for *type 1* $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha]$ *type 2* p.
d. $\sigma \nvDash \forall x \in S. p$ means for *type 1* state σ and for *type 1* $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha]$ *type 2* p.
e. $\vDash \exists x \in S. p$ means for *type 1* state σ, we have σ *type 2* $\exists x \in S. p$.
f. $\vDash \forall x \in S. p$ means for *type 1* state σ, we have σ *type 2* $\forall x \in S. p$.
g. $\nvDash \exists x \in S. p$ means for *type 1* state σ, we have σ *type 2* $\exists x \in S. p$.
h. $\nvDash \forall x \in S. p$ means for *type 1* state σ, we have σ *type 2* $\forall x \in S. p$.