Miscellaneous Problems on Semantics and Computability

* 1. For each of the following states $\sigma_1, \sigma_2, \sigma_3$:

1. $\sigma_1 = [x \mapsto 0, y \mapsto 0]$
2. $\sigma_2 = [x \mapsto -1, y \mapsto 10]$
3. $\sigma_3 = [x \mapsto -1, z \mapsto 10]$

calculate the following:

(a) the denotation of the arithmetic expression $x + (y * 3)$;

(b) the denotation of the Boolean expression $x == y \;\&\&\; x - y \leq 3$;

(c) And, for each $i \in \{1, 2, 3\}$, a state σ such that $\mathsf{while}\ x \leq y\ \&\&\ !(x = y)\ \mathsf{do}\ x \leftarrow x+z,\ \sigma_i \Downarrow \sigma$ when one exists.

* 2. Find two distinct states $\sigma, \sigma' \in \mathsf{State}$ such that $\mathsf{while}\ 1 \leq x\ \mathsf{do}\ y \leftarrow y+x,\ x \leftarrow x-1,\ \sigma \Downarrow [y \mapsto 10]$ and likewise for $\mathsf{while}\ 1 \leq x\ \mathsf{do}\ y \leftarrow y+x,\ x \leftarrow x-1,\ \sigma' \Downarrow [y \mapsto 10]$.

** 3. Suppose $e \in \mathcal{B}$ is a Boolean expression and $S_1, S_2, S_3 \in \mathcal{S}$ are statements. Prove that the statement $\mathsf{if}\ e\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2$ and the statement $\mathsf{if}\ e\ \mathsf{then}\ \{\mathsf{if}\ e\ \mathsf{then}\ S_1\ \mathsf{else}\ S_3\}\ \mathsf{else}\ \{\mathsf{if}\ !e\ \mathsf{then}\ S_2\ \mathsf{else}\ S_3\}$ are semantically equivalently.

** 4. Let $\mathbb{P}$ be the three element set $\{+, -, \pm\}$. We define the function $\mathsf{sign}_\sigma : \mathcal{A} \to \mathbb{P}$ for a given state

$\sigma \in \mathsf{State}$ by recursion as follows:

$$
\begin{aligned}
\mathsf{sign}_\sigma(x) &= \begin{cases} + & \text{if } \sigma(x) \geq 0 \\ - & \text{otherwise} \end{cases} \\
\mathsf{sign}_\sigma(n) &= \begin{cases} + & \text{if } n \geq 0 \\ - & \text{otherwise} \end{cases} \\
\mathsf{sign}_\sigma(e_1 + e_2) &= \begin{cases} + & \text{if } \mathsf{sign}_\sigma(e_1) = + \text{ and } \mathsf{sign}_\sigma(e_2) = + \\ - & \text{if } \mathsf{sign}_\sigma(e_1) = - \text{ and } \mathsf{sign}_\sigma(e_2) = - \\ \pm & \text{otherwise} \end{cases} \\
\mathsf{sign}_\sigma(e_1 - e_2) &= \begin{cases} + & \text{if } \mathsf{sign}_\sigma(e_1) = + \text{ and } \mathsf{sign}_\sigma(e_2) = - \\ - & \text{if } \mathsf{sign}_\sigma(e_1) = - \text{ and } \mathsf{sign}_\sigma(e_2) = + \\ \pm & \text{otherwise} \end{cases} \\
\mathsf{sign}_\sigma(e_1 * e_2) &= \begin{cases} \pm & \text{if } \mathsf{sign}_\sigma(e_1) = \pm \text{ or } \mathsf{sign}_\sigma(e_2) = \pm \\ + & \text{if } \mathsf{sign}_\sigma(e_1) = \mathsf{sign}_\sigma(e_2) \\ + & \text{if } \mathsf{sign}_\sigma(e_1) \neq \text{ and } \mathsf{sign}_\sigma(e_2) \end{cases}
\end{aligned}
$$

Prove by structural induction over arithmetic expressions, for any arithmetic $e \in \mathcal{A}$ and state $\sigma \in \mathsf{State}$, that:

- If $\mathsf{sign}_\sigma(e) = +$, then $[\![e]\!]_{\mathcal{A}}(\sigma) \geq 0$;
- And, if $\mathsf{sign}_\sigma(e) = -$, then $[\![e]\!]_{\mathcal{A}}(\sigma) < 0$

You should treat this as a single induction proof rather than proving each cases separately.

*** 5. Let P be the following While program:

$$
\begin{aligned}
&\mathsf{while}\ x + 1 \leq y\ \mathsf{do} \\
&\quad y \leftarrow y - 1;
\end{aligned}
$$

Prove by induction, over a combination of x and y, that P terminates from any initial state $\sigma \in \mathsf{State}$ such that $\sigma(x) \leq \sigma(y)$. That is, prove, for all $\sigma \in \mathsf{State}$ such that $\sigma(x) \leq \sigma(y)$, that there exists some $\sigma' \in \mathsf{State}$ such that $P, \sigma \Downarrow \sigma'$. You may *not* assume that $\sigma(x)$ or $\sigma(y)$ are greater than or equal to 0.

The following question uses the notation $\sigma \sim_x \sigma'$ which indicates, for some variable $x \in \mathsf{Var}$ and for two states $\sigma, \sigma' \in \mathsf{State}$, that $\sigma(y) = \sigma'(y)$ for all *other* variable $y \in \mathsf{Var} \setminus \{x\}$. For example, $[x \mapsto 2,\ y \mapsto 3] \sim_x [x \mapsto -100,\ y \mapsto 3]$ but $[x \mapsto 2,\ y \mapsto 3] \not\sim_x [x \mapsto 2,\ y \mapsto 4]$.

*** 6. Let us suppose we introduce a new language construct for Boolean expressions so that the extended grammar is given as follows:

$$B \rightarrow \mathsf{true} \mid \mathsf{false} \mid B\ \&\&\ B \mid B \mid\mid B \mid !B \mid \mathsf{forall}\ x.\,B$$

The denotational semantics for the new construct is given by the following equation:

$$[\![\mathsf{forall}\ x.\,e]\!]_{\mathcal{B}}(\sigma) = \begin{cases} \top & \forall \sigma' \in \mathsf{State}.\, \sigma \sim_x \sigma' \Rightarrow [\![e]\!]_{\mathcal{B}}(\sigma') \\ \bot & \text{otherwise} \end{cases}$$

with all other constructs retaining their original semantics.

(a) Evaluate the denotation of the expressions $\mathsf{forall}\ x.x = 2$ and $\mathsf{forall}\ x.x \leq y \parallel y \leq x$ in the state $[x \mapsto 2, y \mapsto 2]$.

(b) Prove that if $e \in \mathcal{B}$ and $e' \in \mathcal{B}$ are semantically equivalent Boolean expressions, then $\mathsf{forall}\ x.e$ and $\mathsf{forall}\ x.e'$ are semantically equivalent.

(c) Suppose that $e \in \mathcal{B}$ is a Boolean expression such that $[\![e]\!]_{\mathcal{B}}(\sigma) = [\![e]\!]_{\mathcal{B}}(\sigma')$ for any two states $\sigma \sim_x \sigma'$. Prove that $\mathsf{forall}\ x.e$ is semantically equivalent to e.

* 7. Show that the function $f : \mathbb{N} \rightharpoonup \mathbb{N}$ defined by

$$f(x) \begin{cases} \simeq x+1 & \text{if } x^2 - 1 \text{ is at least 2022} \\ \uparrow & \text{otherwise} \end{cases}$$

is computable.

* 8. State whether each of the following statements is true or false.

(a) Every injection has an inverse.

(b) The set $\mathbb{N}$ is decidable.

(c) Some WHILE programs compute total functions.

(d) If a function has an inverse, it must be a surjection.

(e) The set of all WHILE programs is countable.

* 9. Write a program that demonstrates that the function

$$f : \mathbb{N} \rightharpoonup \mathbb{N}$$
$$f(n) \begin{cases} \simeq 2^n & \text{if } n \text{ is divisible by 3} \\ \uparrow & \text{otherwise} \end{cases}$$

is computable.

** 10. Construct a bijection $\mathbb{Z} \times \mathbb{Z} \xrightarrow{\cong} \mathbb{N}$. Prove that it is a bijection by constructing its inverse, and show that it is indeed an inverse.

** 11. Show that the predicate

$$U = \{\ulcorner S_1 \urcorner \mid \text{for all } k \leq 100 \text{ it is true that } [\![S_1]\!]_{\mathtt{x}}(k) + 1 = [\![S_1]\!]_{\mathtt{x}}(k+1)\}$$

is semi-decidable.

** 12. Show that if the predicates $A \subseteq \mathbb{N}$ and $B \subseteq \mathbb{N}$ are decidable then their *symmetric difference*, i.e. the set

$$A \oplus B = \{x \in \mathbb{N} \mid (x \in A \land x \notin B) \lor (x \notin A \land x \in B)\} = (A \cup B) - (A \cap B)$$

is also decidable.

*** 13. Show that the predicate

$$V = \{\langle \ulcorner S_1 \urcorner, \ulcorner S_2 \urcorner \rangle \mid \text{there exists } k \in \mathbb{N} \text{ such that } [\![S_1]\!]_{\mathtt{x}}(k) + 1 = [\![S_2]\!]_{\mathtt{x}}(k+1)\}$$

is undecidable. (The use of $=$ above means that both sides of the equation must be defined, and equal.)