Mini-Quiz Mar. 7

Your name: ____________________

Circle the name of your TA/LA:

Chiyoun Jay Jeffrey Jessica Tina

- This quiz is **closed book**. Total time is 25 minutes.
- There are two (2) problems totaling 15 points.
- Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem. Please keep your entire answer to a problem on that problem's page.
- GOOD LUCK!

DO NOT WRITE BELOW THIS LINE

Problem	Points	Grade	Grader
1	8		
2	7		
Total	15		

Problem 1 (8 points). BAexp's are defined in the Appendix.

(a) (3 points) The value of flatten(e) for $e \in$ BAexp is the sequence of integers in e obtained by "erasing" everything but the integers that appear within tagged variables and tagged `int`'s. For example,

$$\begin{aligned} e &::= \langle \texttt{sum}, \langle \texttt{var}, 3\rangle, \langle \texttt{sum}, \langle \texttt{var}, 2\rangle, \langle \texttt{int}, 2\rangle\rangle\rangle \\ f &::= \langle \texttt{prod}, \langle \texttt{var}, 4\rangle, \langle \texttt{var}, 5\rangle\rangle \\ g &::= \langle \texttt{prod}, e, \langle \texttt{sum}, \langle \texttt{var}, 7\rangle, f\rangle\rangle \\ \text{flatten}(g) &= \langle 3, 2, 2, 7, 4, 5\rangle . \end{aligned}$$

Give a recursive definition of flatten. (You may use the operation of *concatenation* (append) of two sequences.)

(b) (5 points) Prove by structural induction on the definition of BAexp that for all $e \in \text{BAexp}$,

$$2 \cdot \text{length}(\text{flatten}(e)) = |e| + 1$$

Problem 2 (7 points). The following state machine describes an algorithm to multiply any two nonnegative integers, without multiplying or dividing by any number other than 3.

Its states are triples of nonnegative integers (r, s, a). The initial state is $(x, y, 0)$. The transitions are given by the rule that for $s > 0$:

$$(r, s, a) \to \begin{cases} (3r, s/3, a) & \text{if } 3 \mid s \\ (3r, (s-1)/3, a+r) & \text{if } 3 \mid (s-1) \\ (3r, (s-2)/3, a+2r) & \text{otherwise.} \end{cases}$$

(a) (1 point) Circle the predicate that is invariant for this program:

- $P((5, 10, 0)) ::= \quad [(15, 3, 5)]$
- $P((r, s, a)) ::= \quad [rs + a^2 = xy]$
- $P((r, s, a)) ::= \quad [rs + a = xy]$
- $P((r, s, a)) ::= \quad [sx + rxy = ar]$

(b) (4 points) Use the invariant principle to prove that the algorithm is partially correct —that is, if $s = 0$, then $a = xy$.

(c) (2 points) When the state machine reaches a state with $s = 0$, it outputs a and terminates. Prove that the algorithm terminates.

Appendix

The set, BAexp, of *Basic Arithmetic Expressions* is defined recursively as a tagged data type as follows:

- **Base cases:** If $n \in \mathbb{Z}$, then
  1. $\langle \texttt{int}, n \rangle \in$ BAexp, and
  2. $\langle \texttt{var}, n \rangle \in$ BAexp.
- **Constructor cases:** if $e, e' \in$ BAexp, then
  1. $\langle \texttt{sum}, e, e' \rangle \in$ BAexp, and
  2. $\langle \texttt{prod}, e, e' \rangle \in$ BAexp.

The size, $|e|$, of $e \in$ BAexp is defined recursively on this definition by:

- **Base cases:**
  1. $|\langle \texttt{int}, n \rangle| ::= 1$
  2. $|\langle \texttt{var}, n \rangle| ::= 1$
- **Constructor cases:**
  1. $|\langle \texttt{sum}, e_1, e_2 \rangle| ::= |e_1| + |e_2| + 1$
  2. $|\langle \texttt{prod}, e_1, e_2 \rangle| ::= |e_1| + |e_2| + 1$

Some string definitions:

Definition 2.1. The *length*, $|s|$, of a string, s, is defined recursively by the rules:

- $|\lambda| ::= 0$
- $|sa| ::= 1 + |s|$.

Definition 2.2. The *concatenation*, st, of strings s and t over an alphabet, A, is defined recursively on t by the rules:

- $s\lambda ::= s$.
- $s(ta) ::= (st)a$ for $a \in A$.

For example, consider two strings, $s = \langle 1, 3, 6, 2 \rangle$, and $t = \langle 2, 7, 3 \rangle$.

The concatenation of s and t is: $st = \langle 1, 3, 6, 2, 2, 7, 3 \rangle$. Likewise, the concatenation of t and s is $ts = \langle 2, 7, 3, 1, 3, 6, 2 \rangle$.

The length of s is: $length(s) = 4$. The length of t is: $length(t) = 3$.