CS101.3 Homework 1: IMP++

Due: Friday, Jan 23, 3PM

Collaboration: You are allowed and encouraged to work together and collaborate with others. However, your submission must be your own; you must write up your homework without referring to material developed with other groups.

You may use the WWW for reference material. You may use the material you found to develop your understanding, but your submission must be your own.

Summary: you may use any and all resources at your disposal, but your submission must be your own work.

IMP++ is an extension of the IMP language presented in class and textbook with the `++` operators. Namely, the `Aexp` definition is augmented as follows:

$$a \ ::= \ n \mid X \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid X\texttt{++} \mid \texttt{++}X$$

The intended meaning of the `++` operators is the standard one — X`++` increments the value in location X and evaluates to the old value, and `++`X increments the value in location X and evaluate to the new (incremented) value.

1. Write the large step operational semantics for arithmetical expressions in IMP++. Note that since evaluation of arithmetic expressions will now potentially affect the state, the general form of the large step evaluation judgement will be $\langle \sigma, a \rangle \rightarrow \langle \sigma', n \rangle$

Is your semantics deterministic? E.g., do $\langle \sigma, a \rangle \rightarrow \langle \sigma_1, n_1 \rangle$ and $\langle \sigma, a \rangle \rightarrow \langle \sigma_2, n_2 \rangle$ imply $\sigma_1 = \sigma_2$ and $n_1 = n_2$? You do not need to justify your answer.

2. The definition of $a_1 \sim a_2$ needs to be updated as well. Write down the new definition and use it to prove $((X\texttt{++}) + 1) \sim (\texttt{++}X)$.

3. Write the small step operational semantics for arithmetical expressions in IMP++. Write all the relevant rules (not just the ones that involve the `++` operator), even if some of them are the same as in IMP. There are several incompatible small step semantics for IMP++ (pay attention to evaluation order), you must pick the one that agrees with the large step semantics you wrote in problem 1.

4. Formulate and prove a theorem stating the equivalence of your small-step and large-step semantics (similar to the one proven in second lecture). When appropriate, you may use the phrases "the proof of this case is the same as for IMP" and "the proofs of the remaining cases are the same as for IMP".