Assignment 2: Operational Semantics

Complete all the questions. Turn in your answers through the submission page (click on the link on the Assignments page) by 5:00pm on Friday September 23rd.

1. Using the syntax for the imperative language as given in class, modify it to add a new control structure:

   \[ C ::= \ldots | \text{repeat } C \text{ until } B \text{ end} \]

Give the new command the appropriate operational semantic rule in the style given in class, and then show a derivation for the following program:

\[
\begin{align*}
s &:= 0; \\
t &:= 0; \\
\text{repeat} & \quad \text{if } s \text{ mod 3 equal } 1 \text{ then} \\
& \quad \quad t := t \text{ plus } 1; \\
& \quad \text{else} \\
& \quad \quad \text{nop} \\
& \quad \text{end}; \\
& s := s \text{ plus } 1 \\
\text{until } & \quad s \text{ greater } 9 \\
\text{end;}
\end{align*}
\]

The full derivation is very large, so you will need to find a way (like we did for the while loop example in class) to summarize it. However, don’t miss out important steps!

2. Give a reasoned argument why any ‘repeat’ program can be rewritten as a ‘while’ program. i.e. you want to show that:

\[
\forall \sigma, \sigma'. \supseteq \sigma, [\text{repeat } C \text{ until } B \text{ end}, \sigma] \rightarrow \sigma' \iff [C; \text{while not } B \text{ do } C \text{ end}, \sigma] \rightarrow \sigma'
\]

In order to this you will need to examine the forms of derivations involving the loops. If your thinking is general enough, then you will be able to show that the implication is true. Note that the reverse implication is also true, but you don’t need to show it.