CSC 501 - Assignment #3

version 7.0

Due Friday 10/21/16 in Sakai

Problems

1. Let $L(G) = \{(a), ((a)), (((a))), ((((a)))), \ldots\}$,
   (a) Give a grammar G that generates the language $L(G)$.
   (b) Give an inductive definition of the set $L(G)$.
   (c) Give an inductive proof that *all* terms in set $L(G)$ have matched parentheses.

2. Let $c \equiv x_0 := x_1; x_2 := x_1$ and $c' \equiv x_2 := x_1; x_0 := x_1$, show that $c \sim c'$.

3. Let Σ be the set of all states (as defined in class) with elements $\sigma : \mathbf{Loc} \rightarrow \mathbb{I}$. Now, we redefine the initial state $\sigma_0 \in \Sigma$ as

$$\sigma_0(x) = \perp$$

   for all $x \in \mathbf{Loc}$. Here we say that the value of a variable is *undefined* in the initial state.

   (a) If we interpret a variable lookup in the initial state as a *non-terminating* computation:
      i. What effect does this have on our inductive proof that all arithmetic expressions terminate?
      ii. What is the difference between arithmetic expressions that do terminate and arithmetic expressions that do not terminate?
   (b) Which semantics is a better model for the way programming languages such as Java and C work today, $\sigma_0(x) = \perp$ or $\sigma_0(x) = 0$ for all $x \in \mathbf{Loc}$? Why?

All questions are based on the operational semantics rules covered in class.