CSE 322

Intro to Formal Models in CS

Homework #6

Due: Friday 16 Nov 07

W. L. Ruzzo 9 Nov 07

1. Give an informal description of a pushdown automaton recognizing the language defined by the following CFG:

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

   You do not need to follow either of the CFG-to-PDA constructions, nor give a formal correctness proof, but do give an informal argument/explanation for why/how it works.

2. Let G be the following CFG:

$$\begin{aligned} S &\rightarrow AA \mid B \\ A &\rightarrow 0A \mid A0 \mid 1 \\ B &\rightarrow 00B0 \mid 1 \end{aligned}$$

   (a) Describe $L(G)$ in English.

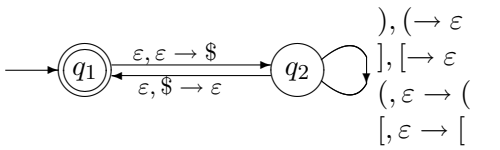
   (b) Give an informal proof that your description is correct. In particular, include definitions of the strings of terminals generated by A and B.

   (c) As in problem 1, give a PDA M recognizing the same language and informally argue correctness.

   (d) Show that M is nondeterministic by giving a (short!) input on which it has two computations. Show the two computations.

   (e) By analogy to the definition of an ambiguous CFG, I define a PDA to be *ambiguous* if there is some input on which it has two different accepting computations. Is M ambiguous? Why or why not? (Either show two accepting computations on some short input, or argue informally why this is impossible.)

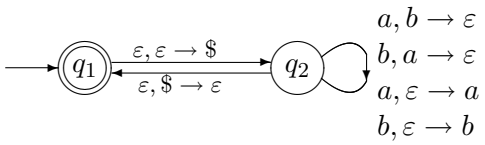
3. Consider the following PDA M:



   (a) Describe in English the language L recognized by M. I want a *nonprocedural* description; don't say "do this then if that do something else..." (Hint: it has a very simple description.)

(b) Prove informally that M recognizes this language. Don't forget to argue that it rejects all strings *not* in L, on all possible computations. You don't need to do a detailed induction proof or the like, but I want an argument that is thorough and convincing.

4. Consider the following PDA M:



(a) Show that M is nondeterministic by giving a short input on which it has two computations. Show the two computations.

(b) Describe in English the language L recognized by M. I want a *nonprocedural* description; don't say "do this then if that do something else..." (Hint: it has a very simple description.)

(c) Prove informally that M recognizes this language. Don't forget to argue that it rejects all strings *not* in L, on all possible computations (and recall it's nondeterministic). You don't need to do a detailed induction proof or the like, but I want an argument that is thorough and convincing.

(d) Show that M is "ambiguous" by giving a short input on which it has two accepting computations. Show the two computations.