# Problem Set 6
## CS 373: Theory of Computation

Assigned: February 28, 2013  Due on: March 7, 2013

**Instructions:** This homework has 3 problems that can be solved in groups of size at most 3. Please follow the homework guidelines given on the class website; submitions not following these guidelines will not be graded.

**Recommended Reading:** Lecture 12, and 13.

**Problem 1**. [Category: Design+Proof] Design a context-free grammar for the language $L = \{a^i b^j \mid 2i \leq j \leq 3i,\ i, j \in \mathbb{N}\}$. Provide a formal proof that your construction is correct. *Hint:* Build a grammar for the case when $j = 2i$ and $j = 3i$, and think of a way to fuse the two together. **[10 points]**

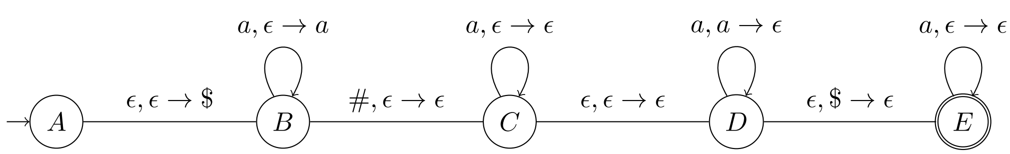
**Problem 2**. [Category: Comprehension+Design] Let $G = (V, \Sigma, R, \langle\text{STMT}\rangle)$ be the following grammar

$$\begin{array}{rcl} \langle\text{STMT}\rangle & \longrightarrow & \langle\text{ASSIGN}\rangle \mid \langle\text{IF-THEN}\rangle \mid \langle\text{IF-THEN-ELSE}\rangle \\ \langle\text{IF-THEN}\rangle & \longrightarrow & \texttt{if condition then } \langle\text{STMT}\rangle \\ \langle\text{IF-THEN-ELSE}\rangle & \longrightarrow & \texttt{if condition then } \langle\text{STMT}\rangle \texttt{ else } \langle\text{STMT}\rangle \\ \langle\text{assign}\rangle & \longrightarrow & \texttt{a := 1} \end{array}$$

where $\Sigma = \{\texttt{if}, \texttt{then}, \texttt{else}, \texttt{condition}, \texttt{a := 1}\}$ and $V = \{\langle\text{STMT}\rangle, \langle\text{IF-THEN}\rangle, \langle\text{IF-THEN-ELSE}\rangle, \langle\text{ASSIGN}\rangle\}$. $G$ is a natural looking grammar for a fragment of a programming language, but $G$ is ambiguous.

1. Show that $G$ is ambiguous. **[5 points]**
2. Give a new unambiguous grammar for the same language. You need not prove that your grammar is correct but explain your construction. You may want to look at examples in Lecture 12. **[5 points]**

**Problem 3**. [Category: Comprehension] Consider the PDA $P$ over the input alphabet $\{0, 1, \#\}$ shown in the figure below; $a$, in the transitions below, is either 0 or 1.




1. Write the formal description of the PDA $P$ listing the states, stack alphabet, transition function, initial state and final states. **[5 points]**
2. For each of the following strings either show that they are accepted by $P$ by describing an accepting computation, or show that they are not accepted by showing the *entire* computation tree on the input: 01#10, 01#01, 01#111000. **[3 points]**
3. Describe the language recognized by the PDA $P$. Give an informal justification for your answer, by explaining how the PDA works. **[2 points]**