Week 6 CS-312 Homework

Cory Ness   Jack Engledow   James Sgrazzutti

November 11, 2020

1 Problem 8.2

1.1 Question

Give a regular grammer for L of the form

$$S \rightarrow SaS \mid b$$

1.2 Answer

$$\begin{aligned} S &\rightarrow b \mid bT \\ T &\rightarrow aG \\ G &\rightarrow b \mid bT \end{aligned}$$

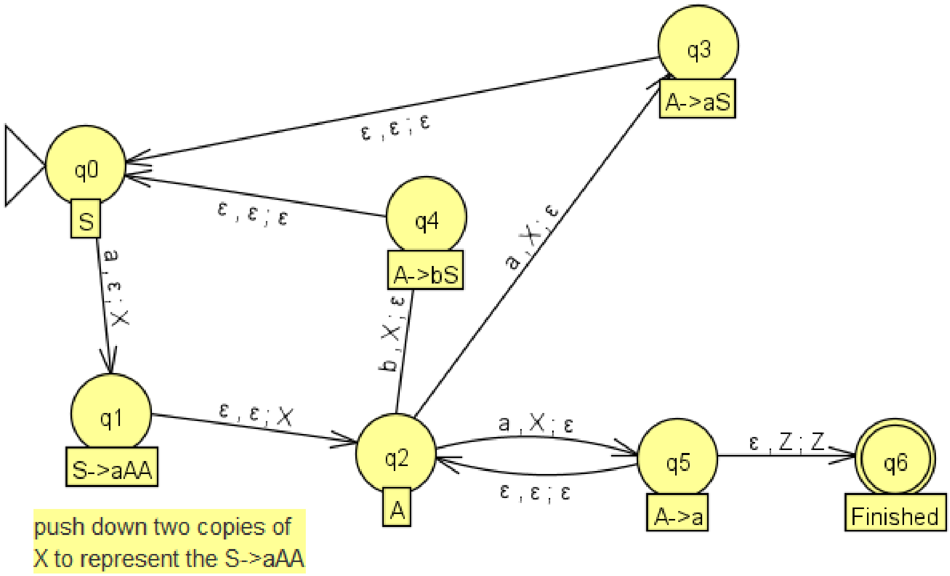
2 Problem 8.8

2.1 Question

Give a PDA equivalent to the following grammar:

$$\begin{aligned} S &\rightarrow aAA \\ A &\rightarrow aS \mid bS \mid a \end{aligned}$$

2.2 Answer



3 Problem 8.11

3.1 Question

A 2-PDA is like a PDA except that it has two stacks. Show the following can be recognized by a 2-PDA.

3.2 $\{0^n 1^n 2^n : n \geq 0\}$

q0, initial state. Transitions to itself on 0 while pushing X to the first stack. Transitions to q1 on nothing.
q1. Transitions to itself on 1 while popping X from the first stack, pushes X to the second stack. Transitions to q2 on nothing.
q2. Transitions to itself on 2 while popping X from the second stack. Transitions to q3 on nothing while popping Z from both stacks and pushing Z onto both stacks.
q3, final state.

3.3 $\{0^n 1^n 2^n 3^n : n \geq 0\}$

q0, initial state. Transitions to itself on 0 while pushing X to the first stack. Transitions to q1 on nothing.
q1. Transitions to itself on 1 while popping X from the first stack, pushes X to the second stack. Transitions to q2 on nothing while popping and pushing Z on the first stack.
q2. Transitions to itself on 2 while popping X from the second stack and pushing X to the first stack. Transitions to q3 on nothing while popping and pushing Z on the second stack.
q3. Transitions to itself on 3 while popping X from the first stack. Transitions to q4 on

nothing while popping Z from both stacks and pushing Z onto both stacks.
q4, final state.

3.4 $\{x\#x : x \in \{0,1\}^*\}$

q0, initial state. Transitions to itself on 0, pushing 0 to the first stack. Transitions to itself on 1, pushes 1 to the first stack. Transitions to q1 on #.
q1. Transitions to itself on nothing, popping 0 from the first stack, and pushing 0 to the second stack. Transitions to itself on nothing, popping 1 from the first stack, and pushing 1 to the second stack. Transitions to q2 on nothing, popping and pushing Z from the first stack.
q2. Transitions to itself on 1, popping 1 from the second stack. Transitions to itself on 0, popping 0 from the second stack. Transitions to q3 on nothing, popping and pushing Z from both stacks.
q3, final state.

4 Problem 9.2

4.1 Question

Convert the following grammar with start variable S into Chomsky Normal Form.

$$\begin{array}{lcl} S & \rightarrow & ASa \mid aB \\ A & \rightarrow & B \mid S \\ B & \rightarrow & b \mid \epsilon \end{array}$$

4.2 Answer

$$\begin{array}{lcl} S & \rightarrow & AC \mid DB \\ A & \rightarrow & B \mid S \mid b \mid \epsilon \\ C & \rightarrow & SD \\ D & \rightarrow & a \end{array}$$

5 Problem 9.4

5.1 Question

Show that the language $\{a^n b^{2n} a^n\}$ is not context-free.

5.2 Answer

Choose string $z = 0^k 1^{2k} 2^k$. Split the string into $z = uvwxy$. Because vwx combined has a length of at most k, the string vx cannot contain both 0's and 2's, and contains at least 1 symbol because vx must be nonempty. Therefore, $z^0 = uwy$ must contain fewer than 1^{2k} 1's or an uneven number of 0's and 1's.

6 Problem 9.12

6.1 Question

Show by example that context-free languages are *not* closed under intersection. Hint: Start with the context-free language $\{0^n1^n2^m\}$.

6.2 Answer

Start with L_1 and L_2 where $L_1 = \{0^n1^n2^m\}$ and $L_2 = \{0^m1^n2^n\}$. Both L_1 and L_2 are context-free languages. $L_1 \cap L_2 = \{0^n1^n2^n\}$ which is not a context-free language.

7 Problem 9.17

7.1 Question

A CFG is called linear if the right-hand side of every production contains at most one variable. Thus, a regular grammar is always linear. But a linear grammar need not generate a regular language; for example, we saw that palindromes are generated by a linear grammar. Show that the set of languages generated by linear grammars is closed under union.

7.2 Answer

Start with languages L_1 and L_2 where both L_1 and L_2 are linear. Produce a new language from the union of the two L where $L- > L_1|L_2$. Clearly L is also linear, as both its productions contain 1 variable, where the variable is a linear CFG.