CIS770 Homework 5

Andre Gregoire

March 24, 2016

Problem1

1.1.

G = (V, Σ, R, S) = ($\{S, X, Y\}$, $\{a, b\}$, R, S)

where R contains the transitions: S $\rightarrow$ { ϵ | aXb | bYa | SS }

X $\rightarrow$ { S | ϵ }

Y $\rightarrow$ { S | ϵ }

1.2.

Any string w generaged by G, w $\in$ L

Base case: 1 step derivation, only one derivation contains one step and that is S $\stackrel{*}{\Rightarrow}$ ϵ, thus w = ϵ so w $\in$ L

Ind. Hyp: Assume every derivation S $\stackrel{*}{\Rightarrow}$ w, with n $\geq$ 1 steps, w $\in$ L

Ind. Step: Let S $\stackrel{*}{\Rightarrow}$ w be a derivation with n + 1 steps, because n+1 > 1, the first derivation must be, aSb, bSa, SS with S being non terminal characters.

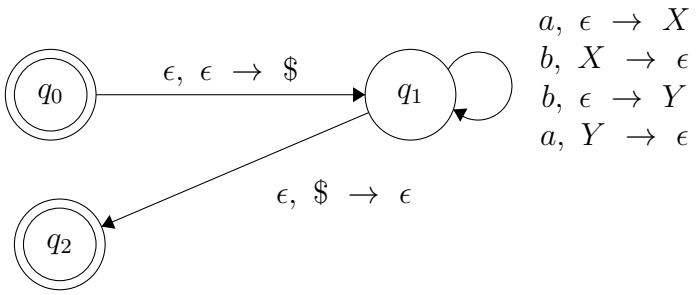
Case(aXb): w must be in the form aw_1b where w_1 is a string derived from non terminal S in the remaining n steps and $w_1 \in$ L by the inductive hypothesis. Thus w $\in$ L because by adding an a and b from the original transition to w_1 keeps the number of a's and b's being equivalent.

Case(bYa): similar to the above case, w must be in the form bw_1a where w_1 is a string derived from non terminal S in the remaining n steps and $w_1 \in$ L by the inductive hypothesis. Thus w $\in$ L because by adding an a and b from the original transition to w_1 keeps the number of a's and b's being equivalent.

Case(SS): w must be in the form $w_1 w_2$ where w_1 is a string derived from the first non terminal S and w_2 from the second non terminal S and $w_1 \in$ L, $w_2 \in$ L by the inductive hypothesis,

Therefore every string generated by G contains an equal amount of a's and b's and exist in L.

1.3.



To build the PDA I just stepped through the CFG. It is in an accepting state when the string is empty thus the initial state is an accepting state because there is an equal number of a's and b's and the stack is empty. After we push our initial symbol, in this case '$' we start reading in characters from the string and pop/push to the stack accordingly. If we read an 'a' and there is a non-terminal 'Y' on the top of the stack we pop it and push nothing. This is denote a pair of a and b's in the string so each pair makes up a single 'a' and a single 'b' this is to keep track of the number of a's and b's we've read in. if there is not a non-terminal Y on the top of the stack we push a non-terminal X to the stack. Similar to 'a' if we read a 'b' and there is a non-terminal X on the top of the stack we push nothing because we found a pair, and if there is not a non-terminal X we push a non-terminal Y to hopefully find a matching 'a' to pop it from the stack. If the string in question does exist in our language the final character in the stack should be our initial push '$' and we can pop it and go back to the accepting state.

P = (Q, Σ, Γ, δ, q_0, F)

Q = $\{q_0, q_1, q_2\}$

Γ = V $\cup$ Σ $\cup$ {\$} where \$ $\notin$ V $\cup$ Σ

$q_0 = q_0$

F = $\{q_0, q_2\}$

δ transitions

$\delta(q_0, \epsilon, \epsilon) = \{(q_1, \$)\}$

$\delta(q_1, b, X) = \{(q_1, \epsilon)\}$

$\delta(q_1, a, Y) = \{(q_1, \epsilon)\}$

$\delta(q_1, a, \epsilon) = \{(q_1, X)\}$

$\delta(q_1, b, \epsilon) = \{(q_1, Y)\}$

$\delta(q_1, \epsilon, \$) = \{(q_2, \epsilon)\}$

Problem2

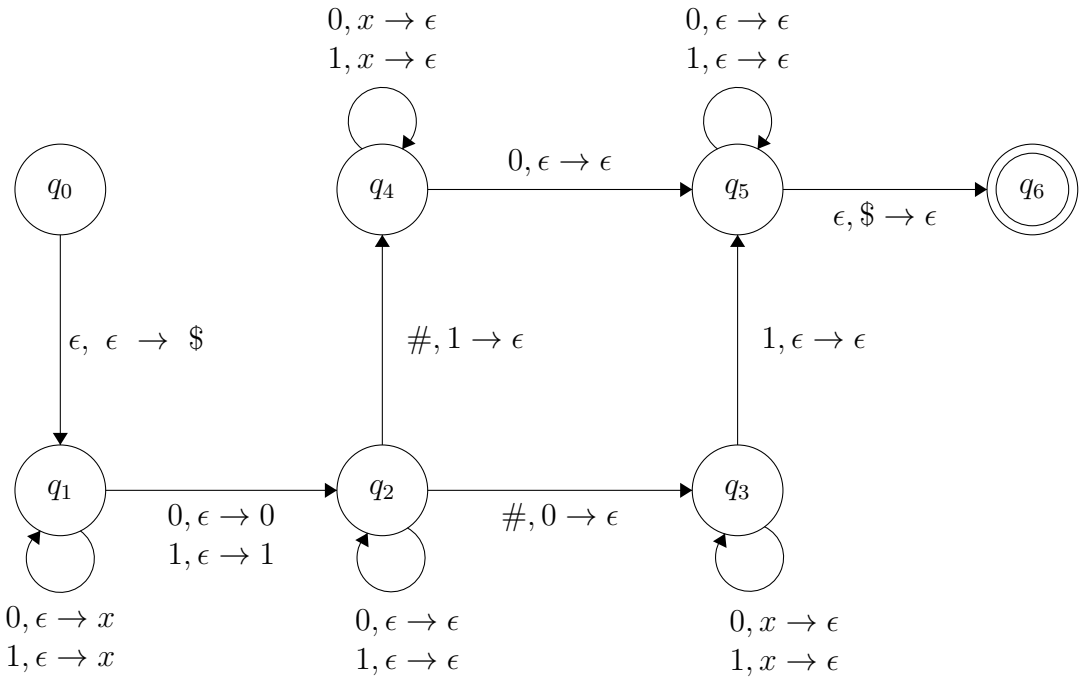
2.1. G = (V, Σ, R, S) = ($\{S, X, Y, R, T\}$, $\{a, b, c\}$, R, S)
where R contains the transitions:

S $\rightarrow$ X | Y
X $\rightarrow$ Xc | R
Y $\rightarrow$ aY | T
R $\rightarrow$ aRb | ϵ
T $\rightarrow$ bTc | ϵ

Yes this grammar is ambiguous because a single string can be generated from multiple transition paths, such as 'abc'. You can obtain 'abc' by starting in either of the starting transitions (S $\rightarrow$ X | Y). i.e.

S $\Rightarrow$ U $\Rightarrow$ aU $\Rightarrow$ aX $\Rightarrow$ abXc $\Rightarrow$ abϵc = abc
S $\Rightarrow$ V $\Rightarrow$ Vc $\Rightarrow$ Yc $\Rightarrow$ aYbc $\Rightarrow$ aϵbc = abc

Extra Credit



We begin by pushing the start symbol ($) and we need to make sure the strings x and y are not equal. So there is some character in a random position i such that $x_i \neq y_i$. We will push filler symbols(x) onto the stack until this random position is chosen and place this value on the stack. This value determines which path we choose when branch out after reading in the middle character(#). In both paths the filler values are popped and if we reach the end of the placeholders and the next value is different then x_i value used to determine our branched path it moves to the next state to finish reading the string and move onto the accepting state, however if they are the same the current thread is stuck and is thus not accepted.

Formal Definition, I don't think this was necessary but if something about how i denoted this was wrong please let me know.

P = (Q, Σ, Γ, δ, q_0, F) where:

Q = $\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

Σ = {0,1}

Γ = V $\cup$ Σ $\cup$ {\$, x} where \$, x $\notin$ V $\cup$ Σ

$q_0 = q_0$

F = $\{q_0, q_6\}$

Transitions(δ):

$$\begin{aligned}
\delta(q_0, \epsilon, \epsilon) &= \{(q_1, \$)\} & \delta(q_3, 0, x) &= \{(q_3, \epsilon)\} \\
\delta(q_1, 0, \epsilon) &= \{(q_1, x)\} & \delta(q_3, 1, x) &= \{(q_3, \epsilon)\} \\
\delta(q_1, 1, \epsilon) &= \{(q_1, x)\} & \delta(q_3, 1, \epsilon) &= \{(q_5, \epsilon)\} \\
\delta(q_1, 0, \epsilon) &= \{(q_2, 0)\} & \delta(q_4, 0, x) &= \{(q_3, \epsilon)\} \\
\delta(q_1, 1, \epsilon) &= \{(q_2, 1)\} & \delta(q_4, 1, x) &= \{(q_3, \epsilon)\} \\
\delta(q_2, 0, \epsilon) &= \{(q_2, \epsilon)\} & \delta(q_4, 0, \epsilon) &= \{(q_5, \epsilon)\} \\
\delta(q_2, 1, \epsilon) &= \{(q_2, \epsilon)\} & \delta(q_5, 0, \epsilon) &= \{(q_5, \epsilon)\} \\
\delta(q_2, \#, 1) &= \{(q_4, \epsilon)\} & \delta(q_5, 1, \epsilon) &= \{(q_5, \epsilon)\} \\
\delta(q_2, \#, 0) &= \{(q_3, \epsilon)\} & \delta(q_5, \epsilon, \$) &= \{(q_6, \epsilon)\}
\end{aligned}$$

Note: Is there such thing as a pda with more than one stack? My original idea for this problem ended up requiring a second stack but I was unsure if it was possible.