Middle East Technical University — Department of Computer Engineering

CENG 280

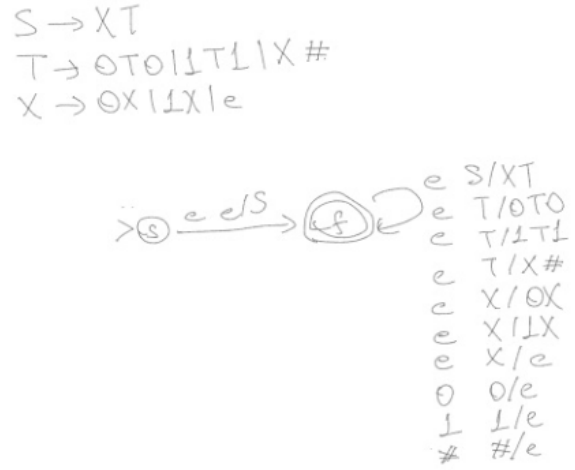
Formal Languages and Abstract Machines

Spring 2022-2023

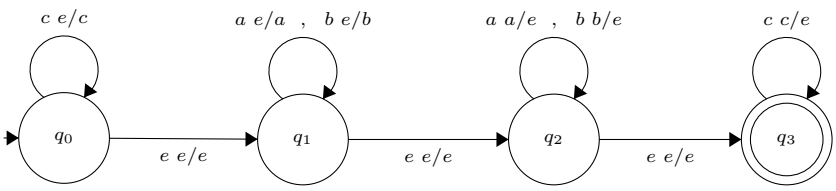
Homework 4
Sample Solutions

Question 1 (40 pts)

1. $\{w\#x \mid x^R \textit{ is a substring of } w \textit{ for } w, x \in \{a,b\}^*\}$



2. $\{c^n w w^R c^n | w \in \{a,b\}^* \wedge n \geq 0\}$
$M_2 = (K, \Sigma, \Gamma, \Delta, s, F)$ where $K = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a,b,c\}$, $\Gamma = \{a,b,c\}$, $s = q_0$, $F = \{q_3\}$ and
$\Delta = \{$
$((q_0, c, e), (q_0, c))$,
$((q_0, e, e), (q_1, e))$,
$((q_1, a, e), (q_1, a)), ((q_1, b, e), (q_1, b))$,
$((q_1, e, e), (q_2, e))$,
$((q_2, a, a), (q_2, e)), ((q_2, b, b), (q_2, e))$,
$((q_2, e, e), (q_3, e))$,
$((q_3, c, c), (q_3, e))$
$\}$



Question 2 (30 pts)

It is known that the class of Context-free Languages is closed under Kleene Star. Give a counterexample (with clear definition of a grammar) to show that for an arbitrary CFL $L = L(G)$ where $G = (V, \Sigma, R, S)$ adding rule $S \to SS$ does not generate L^*.

A counter example can be shown for the following grammar G. $G = (\{S, a\}, \{a\}, R, S)$ where
$R = \{S \text{ -> } \texttt{a}\}$

Now lets add new rule to form a new grammar with rules R'.
$R' = \{S \text{ -> } \texttt{a} \mid \texttt{SS} \}$
This grammar will not generate e which is n $L(G)^*$. Thus this new grammar is incapable of generating star closure of $L(G)$. Therefore adding the rule does not prove the closure under Kleene Star.

Question 3 (30 pts)

1.
   - $L_1 = \{a^n b^n | n \geq 0\}$ is a S-CFL.
     Say your stack symbol is X. Start at the initial state, push an X to the stack for each a that is read. Once a's are finished, go to another state in which an X is popped from the stack for each b that is read.
   - $L_2 = \{w | w \in \{a, b\}^*$ *and the number of* $a's$ *in* w *is not equal to the number of* $b's$ *in* $w\}$ is an S-CFL if the automaton has a (additional) bottom of the stack symbol, is not an S-CFL else.
   - $L_3 = \{a^n b^{m+n} c^m | m, n \in \mathbb{N}\}$ is an S-CFL if the automaton has a (additional) bottom of the stack symbol, is not an S-CFL else.

2. Many many many different correct answers are possible.

3. A counter, value of which cannot be read (but can be compared to zero, if the stack-based equivalent automaton has an additional bottom of the stack symbol).(Cannot store a negative value by definition.)

4. Since S-PDAs have a single stack symbol, the stack can only be used to count number of occurrences of something (some symbol etc.). This can be simulated with a single counter: Instead of pushing the stack symbol, increment the value at the counter; instead of popping a symbol from the stack, decrement the value of the counter.