Middle East Technical University Department of Computer Engineering

CENG 280

Formal Languages and Abstract Machines

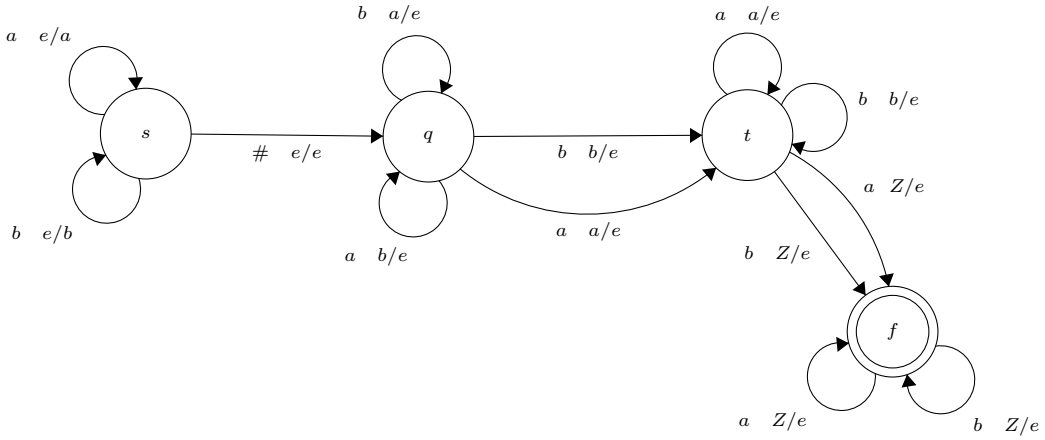
Spring 2022-2023

Homework 4

Name Surname:Kerem Karabacak
Student ID: 2644417

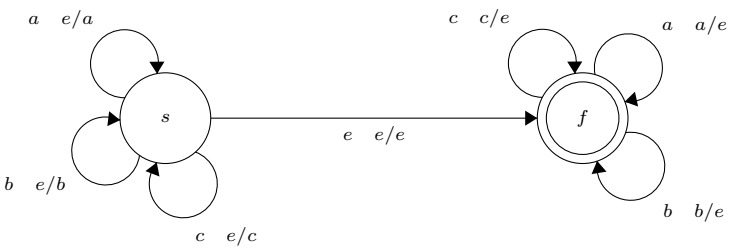
Answer for Q1

1. M = (K, $\sum$, Γ, $\triangle$, s, F) where;
K = {s,q,f} $\sum$ = {a,b,#} Γ = {a,b,Z} s = {s} F = {f}
$\triangle$:
((s,e,e),(s,Z)), ((s,a,e),(s,a)), ((s,b,e),(s,b)), ((s,#,e),(q,e)), ((q,a,a),(f,e)), ((q,b,b),(f,e)), ((f,a,Z),(f,e)), ((f,b,Z),(f,e))



Explanation: We are startnig with pushing Z, which is a empty stack checker for us. Then pushing all symbols until #. After # we are changing state and checking for equalities.

2. M = (K, $\sum$, Γ, $\triangle$, s, F) where;
K = {s,f} $\sum$ = {a,b,c} Γ = {a,b,c} s = {s} F = {f}
$\triangle$:
((s,c,e),(s,c)), ((s,a,e),(s,a)), ((s,b,e),(f,e)), ((f,c,c), (f,e)), ((f,a,a),(f,e)), ((f,b,b),(f,e))



Answer for Q2

CFL's are closed under Kleene star means that if we take Kleene star of any CFL, we will get CFL as a resutlt. We know that Kleen star of a language must contain empty string, but if a CFL does not contain S $\rightarrow$ e as a rule, then we won't be able to obtain empty string from this language. Consider following example:
G = {w = $a^m b^n$ and m = n+ 1 }
V = $\{S\} \cup \sum$ S = S $\sum$ = {a,b}
R = $\{S \rightarrow aSb|a\}$
If we add S $\rightarrow$ SS to the Rules set, we still won't be able to obtain empty string. Therefore CFL won't be equal to L^*.

Answer for Q3

1. I will use X as only stack symbol ,and Z for the emptyness check.
a)We can push X as "a" comes as input. We can pop X's as "b" comes as input. Also there should be a transition to final state if stack has Z only. If input is valid it can reach the final state. Since we only used X and Z as stack symbols, this language is S-CFL.

b)This language is not S-CFL. In this language we should check emptyness more than one. For example aabbbaa. In this case if we push X for all a's first, and pop them when the input is b, stack will be empty for once. Again we need to push X for b, and stack will be empty for the second time. Since we have one emptyness check. We can't control this language by S-PDA. Therefore this language is not a S-CFL.

c)Again we can start pushing X when input is a, and pop X's when input is b in the same state. After reaching Z, make a transition to another state, and start pushing X when input b, pop X's when input is c. Again we should add a transition if stack has only Z to the final state. There is no other transition in final state, so if more c's available in input, since there is no valid transition for that input will rejected. It's same for second state, if there is more b's there will be no valid transitions. Since we used only X and Z, this language is S-CFL.

2. L = $\{w = a^m b^n, m = n + 2\}$
G = (V,$\sum$, R,S): V = $\{S\} \cup \sum$ $\sum$ = {a,b} S = S
R = $\{S \rightarrow aSb|aa\}$
If we want to look from S-CFL perspective, we should push all a's, then pop them as b comes. After b's finished, we should change two states(to final state) and pop 2 a's. If stack is empty input is accepted. If it's not input is rejected. Since we only use a for pop and push operations, this language is S-CFL.

3. Since we are dealing number of symbols in S-CFL's, a counter is what we need. It's basically holds the number of occurence for each symbol.

4. FA's problem is lack of memory. With counter we can fix it partially. This counter can be incremented or decremented by one with each symbol from the input. If we are in final state and all counter's are 0, then input is accepted. Otherwise rejected. Why did I say partially? Because with a counter we can't keep track of patterns. We can't figure out where was the last occurence of any symbol. This is why it's less powerful compare to an infinite stack.

5. If S-CFL is closed under complementation, then complement of this must be a S-CFL. Let's consider third language from first part. Let's say complement of this is L'. L' contains also following language i.e. following language is subset of L':

$$L_1 = \{w | w = a^n b^n c^n\}$$

We know that this language is not only S-CFL, also not CFL i.e. there is no PDA for this language. Therefore S-CFL's are not closed under complement.