Middle East Technical University Department of Computer Engineering

CENG 280

Formal Languages and Abstract Machines

Spring 2022-2023

Homework 2

Name Surname: Ozan Cinci
Student ID: 2448223

Answer for Q1

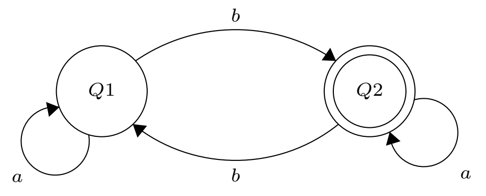
1)

First, there will be two states. The first one includes final states and the second one includes non-final states.

Q1 = q0, q1, q3, q4

Q2 = q2, q5

This is the construction step. Then we check for 1 length strings. For input string "a", all the states in Q1 ends up in Q1, and all the states in Q2 ends up in Q2. For input string "b", all the states in Q1 ends up in Q2, and all the states in Q2 ends up in Q1. So, there will be no split in Q1 and Q2. Since the last two step result is the same there is no need to check for longer strings. Even if we do check, there will be no split in any new states. As a result, there will be only two states.



2)

number of equivalence classes = number of states in minimal DFA

So there will be two equivalence classes. The first ones are already accepted one by the language since they contain at least one "b", and the second one is to be accepted in case of it adds "b" to the end. The string "e" is also not accepted since it does not contain any "b".

[b] = L

[e] = Σ* - L = a* $\cup$ Lb

3)

MyHill-Nerode Theorem states that a language is regular if and only if there is a finite number of equivalence classes under the indistinguishability relation. Let's consider a case. Assume we have set of elements that following this patterns and has infinity many elements S = (abc, a^2bc, a^3bc, a^4bc, a^5bc ...).

As you can see, none of items in S follows the rules of language. What if we add d^2 to the end. Then only one item which is a^4bc will be accepted and the others will not. So there are a split. What if we add d^4, again only one item will be accepted and others not so split. As you can see, there are infinitely many items and if we follow the acceptence patterns, there will be infinitely many equivalence classes. Since number of equivalence classes are not finite, this language cannot be regular.

Answer for Q2

1)

$$G = \{V, \Sigma, R, S\}$$
$$V = \{S, a, b\}$$
$$\Sigma = \{a, b\}$$
$$R = \{S- > aSb|bSa|b\}$$

2)

$$G = \{V, \Sigma, R, S\}$$
$$V = \{S, S1, S2, 0, 1, 2\}$$
$$\Sigma = \{0, 1, 2\}$$
$$R = \{S- > S1S2|e,$$
$$S1- > 0S_1 1|e,$$
$$S2- > 1S_2 2|e\}$$

3)

$$G = \{V, \Sigma, R, S\}$$

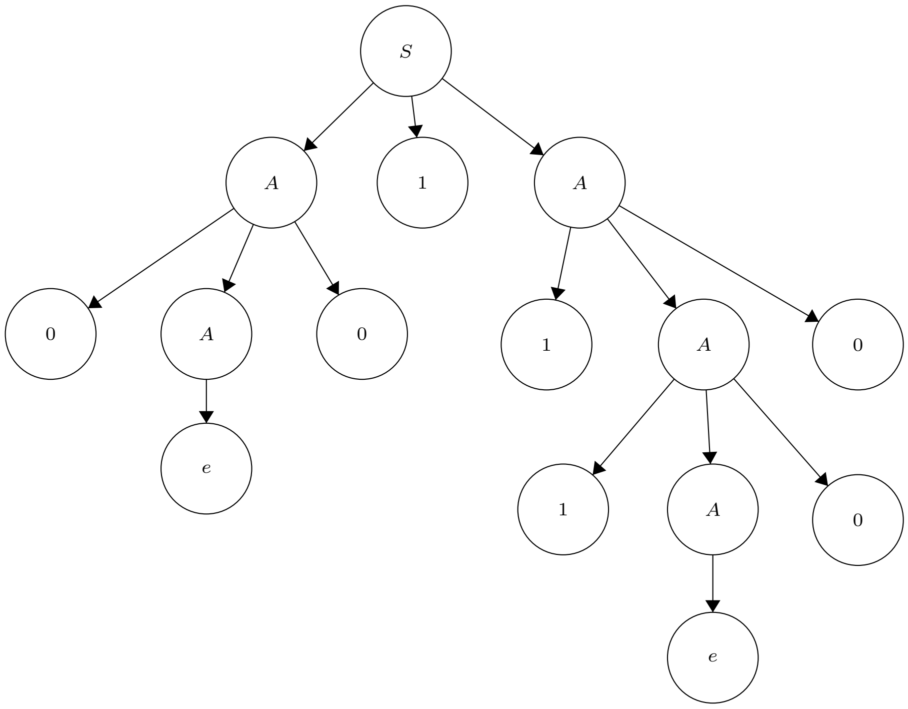
$$V = \{S, A, 0, 1\}$$

$$\Sigma = \{0, 1\}$$

$$R = \{S- > A0A|A1A,$$

$$A- > 0A1|0A0|1A1|1A0|e\}$$

$Parsetreefor: 0011100$



Answer for Q3

1)

It generates strings that the first and last letters are same. Σ=0,1 and length of strings are arbitrary. Except the equality of the first and last letter rule, there is no such rule for inside elements. They can be any combination over Σ*.

$$R.e : 0(0 + 1)^*0 + 1(0 + 1)^*1$$

2)

It generates arbitrary length strings that contains at least two 1s. They can be consecutive or not, it does not matter. Every strings over Σ* is a string that can be generated by this context free language as long as string contains at least two 1s in any order in the string.

$$R.e : (0 + 1)^*1(0 + 1)^*1(0 + 1)^*$$