ITU Faculty of Computer and Informatics
BLG311E - Formal Languages and Automata
Final
May 29, 2016

Duration: 100 minutes

Name:	**Student Id:**	**Signature:**

1. (35p) Consider the regular languages represented by the following regular expressions

$$L_1 = a^*baa^*$$

$$L_2 = aba^*$$

a) Heuristically find the simplest regular expression that can recognize all the words that belong to L_1 but not L_2. $(L_1 - L_2)$

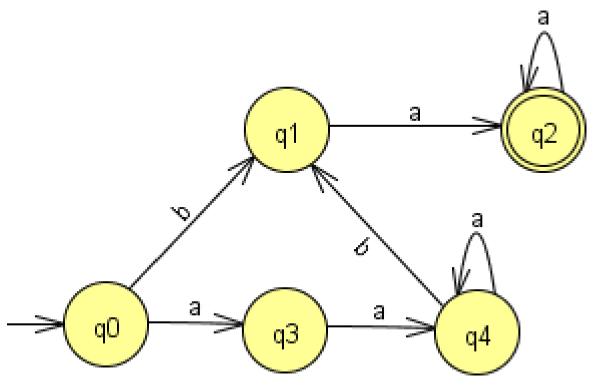
b) Derive the DFA of the regular expression you have found in step (a) and draw its state transition diagram (You may omit the dead state in your diagram)

c) Systematically derive the regular expression of the DFA you found in step (b).

Solution:

a) $L_1 - L_2 = a^*baa^* - aba^* = (\Lambda \vee aaa^*)baa^*$ as $a \subset a^*$ and $aa^* \subset a^*$

b)



c) $q_0 = \Lambda$

$q_1 = q_0 b \vee q_4 b$

$q_2 = q_1 a \vee q_2 a$

$q_3 = q_0 a$

$q_4 = q_3 a \vee q_4 a$

Theorem: $x = xa \vee b \wedge \Lambda \notin A \Rightarrow x = ba^*$

$q_0 = \Lambda \rightarrow q_1 = b \vee q_4 b$ and $q_3 = a$

$q_3 = a \rightarrow q_4 = aa \vee q_4 a$ and using the theorem $q_4 = aaa^*$

$q_4 = aaa^* \rightarrow q_1 = b \vee aaa^*b$

$q_1 = b \vee aaa^*b \rightarrow q_2 = (b \vee aaa^*b)a \vee q_2 a$

Using the theorem $q_2 = (b \vee aaa^*b)aa^* = (\Lambda \vee aaa^*)baa^*$

Name:	Student Id:	Signature:

2. (35p) **a)** Let L be a context-free language and R be a regular language. What can you say about the type of the language $L \cap R$. Explain why.

b) Sort the below given automata in increasing computational capability (not efficiency). If they have equal power state it with an equal sign. (Example format: 3 < 5 < 6 = 6 < 10 = 10 = 10 < 12)

Automata: DFA, NFA, DPDA (deterministic PDA), NPDA (non-deterministic PDA), DTM (deterministic Turing Machine), NTM (non-deterministic Turing Machine)

c) Let $L = \{w \mid w \in \{0,1\}^* \text{ and w has equal number of } 0'\text{s and } 1'\text{s}\}$. Give the Context-Free grammar that generates this language. Show the parse tree for the string 01001011 with your grammar.

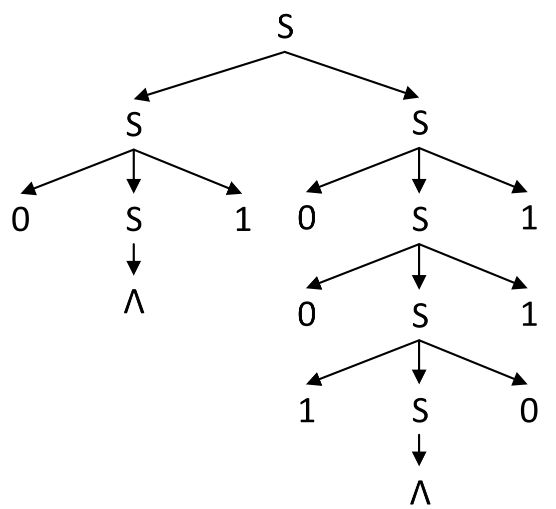
Solution:

a) A PDA is needed for recognizing L and a FA is needed for recognizing R. Thus recognizing $L \cap R$ requires a PDA and a FA working in parallel to produce the intersection of their accepted states. This is also a PDA using the stack in some of its states which makes $L \cap R$ a context-free language.

b) DFA = NFA < DPDA < NPDA < DTM = NTM (non-determinism makes a difference for only PDAs as all NPDAs do not have an equivalent DPDA)

c) $S \rightarrow 0S1 \mid 1S0 \mid SS \mid \Lambda$

The parse tree for the string 01001011:



Name:	Student Id:	Signature:

3. (35p) Design a Turing Machine that can count 0's and 1's given a binary string as an input. If the input string ends with a 0, it counts the 0's in the string and writes as many 1's after the blank symbol (#) following the input. If the last symbol of the input string is 1, then your machine will count the number of 1's in the string and again will write as many 1's after the blank following the input. You can see how the machine works from the following examples. As you can see from the examples, initially the tape head is on the first blank after the input. You can assume that input will never be an empty string. (Underlined symbol represents the position of the tape head).

001110110# =>* 001110110#1111#

1101# =>* 1101#111#

Solution: