BLG311E – FORMAL LANGUAGES AND AUTOMATA

2017 SPRING

RECITEMENT 6 (MIDTERM 2 SOLUTIONS)

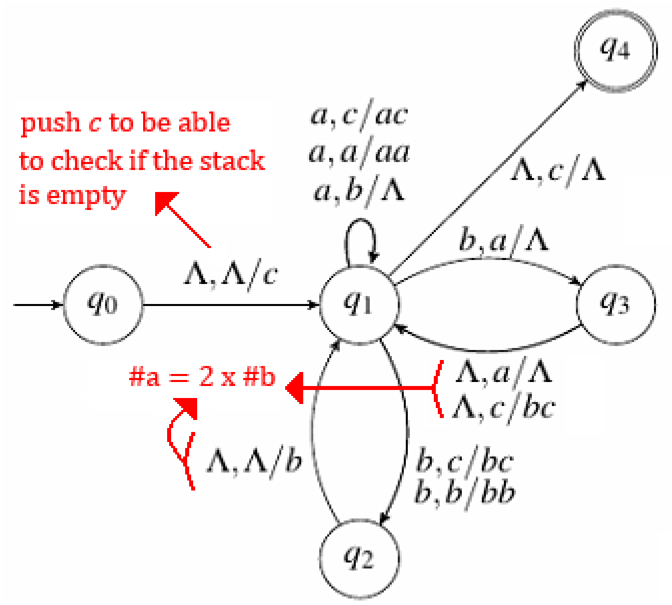
Q1) (30 p.)

(a) (15 p.) Build a PDA for the language L which contains words $\{w \in \{a, b\}^* \mid \#a = 2 \times \#b\}$ (The number of a's is twice the number of b's.). Draw the finite automaton that shows the state-transition relation of the PDA.

(b) (15 p.) Build the grammar of the language in (a).

Solution:

(a)



(b) $<S> ::= <S><S> \mid aa<S>b \mid ab<S>a \mid ba<S>a \mid$
$b<S>aa \mid a<S>ab \mid a<S>ba \mid \Lambda$ (Type-2)

Q2) (30 p.) Prove that the following expression cannot be represented by a Type-3 grammar.

$$a^n b^+ a^n$$

Solution:

We can use pumping lemma to prove that the given expression is not regular therefore cannot be represented by a Type-3 grammar.

$$L = a^n b^+ a^n$$

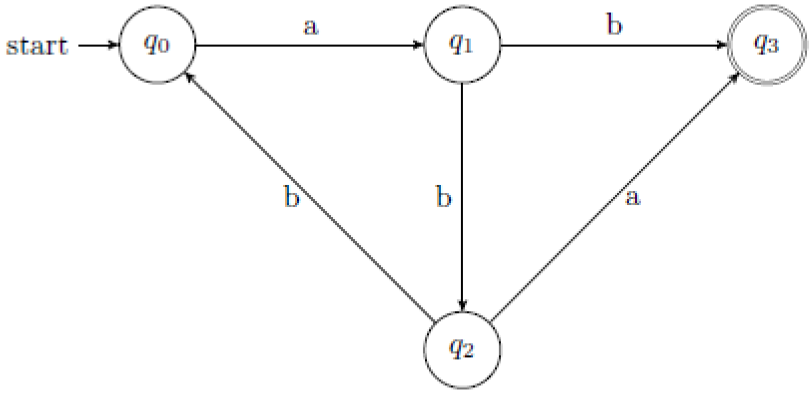
Assumptions:

- Suppose that there exists a finite automaton M having k states and accepting L.
- Choose $x = a^k b^m a^k,\ m > 0$ so $x \in L$ and $|x| \geq k$.

By pumping lemma:

- $x = uvw$, $|v| > 0$ and $|uv| \leq k$.
- For all possible splits that satisfy these rules: $v = a^l$ where $1 \leq l \leq k$.
- Lemma states that for a regular language all $uv^i w, i \geq 0$ must belong to the language.
- Consider any string $uv^i w$ where $i \neq 1$.
- $(i = 0)\ uw = a^{k-l} b^m a^k$ → The string does not belong to L since $k - l \neq k$.
- $(i > 1)\ uv^i w = a^{k+(i-1)l} b^m a^k$→ The string does not belong to L since $k + (i-1)l \neq k$.
- This is a contradiction so L is not regular.

Q3) (40 p.) Consider the following automaton.



(a) (5 p.) Heuristically state the simplest regular expression that this NFA recognizes.

(b) (15 p.) Systematically build an equivalent DFA to this NFA and draw its state-transition diagram.

(c) (5 p.) Build the simplest grammar recognized by the DFA in item (b).

(d) (15 p.) Systematically build the regular expression recognized by the DFA in item (b).

Solution:

(a) $(abb)^*(ab \vee aba)$

(b) $s_0 = q_0$

$\delta'(s_0, a) = \delta(q_0, a) = q_1 \to s_1$

$\delta'(s_0, b) = \delta(q_0, b) = \emptyset$

$\delta'(s_1, a) = \delta(q_1, a) = \emptyset$

$\delta'(s_1, b) = \delta(q_1, b) = \{q_2, \boldsymbol{q_3}\} \to \boldsymbol{s_2}$

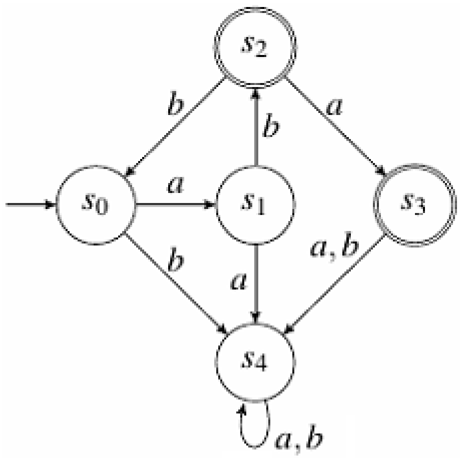
$\delta'(s_2, a) = \delta(\{q_2, q_3\}, a) = \boldsymbol{q_3} \to \boldsymbol{s_3}$

$\delta'(s_2, b) = \delta(\{q_2, q_3\}, b) = q_0 \to s_0$

$\delta'(s_3, a) = \delta(q_3, a) = \emptyset$

$\delta'(s_3, b) = \delta(q_3, b) = \emptyset$

$\delta'(\emptyset, a) = \delta'(\emptyset, b) = \emptyset \to s_4$



(c) $< s_0 > ::= a < s_1 >$

$< s_1 > ::= b < s_2 > | \, b$ (s_4 and s_3 are not included in grammar as s_4 is a death state

$< s_2 > ::= b < s_0 > | \, a$ and transitions from s_3 are only to this dead state s_4.)

(d) __**Theorem:**__ $x = xa \text{ v } b \wedge \Lambda \notin A \Rightarrow x = ba^*$ **Similarly:** $x = ax \text{ v } b \Rightarrow x = a^*b$

$L = s_0 = as_1$

$s_1 = bs_2 \vee b$

$s_2 = bs_0 \vee a$

Place s_0 in the expression of s_2: $s_2 = bs_0 \vee a = bas_1 \vee a$

Place s_2 in the expression of s_1: $s_1 = bs_2 \vee b = b(bas_1 \vee a) \vee b = bbas_1 \vee ba \vee b$

Using the theorem above: $s_1 = (bba)^*(ba \vee b)$

Place s_1 in the expression of s_0: $s_0 = as_1 = a(bba)^*(ba \vee b) = L$