CS3133 Foundations of CS - A Term 2009 Prof. Carolina Ruiz
Exam 1 Solutions By Prof. Ruiz and Li Feng Dept. of Computer Science
Sept. 25, 2009 WPI

PROBLEM 1: Regular Expressions (15 points)

Give a regular expression that represents the set of strings over $\{a, b\}$ that contain the substring aa AND the substring bb.

> **Examples:** The strings $aba\underline{bb}abab\underline{aaaa}$ and $\underline{aaaa}ba\underline{bbb}ab\underline{aabb}a$ satisfy the condition above, but $\underline{aaa}bab\underline{aa}$, $ba\underline{bbb}aba$, and $ababababab$ do not.

Solution:

Since aa can occur before or after bb, a string in our set can be of either of the two forms:

$$string_1\ aa\ string_2\ bb\ string_3 \text{ or } string_1\ bb\ string_2\ aa\ string_3$$

where $string_1$, $string_2$, and $string_3$ are strings of zero or more a's or b's, that is, $(a \cup b)^*$. Hence, a regular expression that represents this set is:

$$(a \cup b)^* aa (a \cup b)^* bb (a \cup b)^* \bigcup (a \cup b)^* bb (a \cup b)^* aa (a \cup b)^*$$

PROBLEM 2: Normal Forms for CFGs (35 points + 5 bonus points)

Consider the following grammar G:

$$
\begin{aligned}
S &\to aABC \mid a \\
A &\to aA \mid a \\
B &\to bcB \mid bc \\
C &\to cC \mid c
\end{aligned}
$$

1. Note that the start symbol S is non-recursive and that G does not contain λ–rules.

2. **(5 points)** Argue decisively that G does not contain any chain rules.

Solution:
This grammar does not contain any rules of the form $X \to Y$ where X and Y are variables.

---

3. **(10 points)** Argue decisively that G does not contain any useless symbols (neither non–terminal–generating variables (5 points) nor unreachable symbols (5 points)).

Solution:

- Each variable in the grammar can generate a terminal string: For instance, S and A generate a, B generates bc, and C generates c.
- Each variable is reachable from the start symbol S.

---

4. **(20 points)** Now, convert G into its equivalent Chomsky normal form. Show your work.

Solution:
This problem is solved in your textbook. See Example 4.5.1 page 123.

The idea is to convert each rule in the grammar to one of the following forms:

- $X \to YZ$, where Y and Z are variables different from the start symbol S
- $X \to a$, where $a \in \Sigma$
- $S \to \lambda$

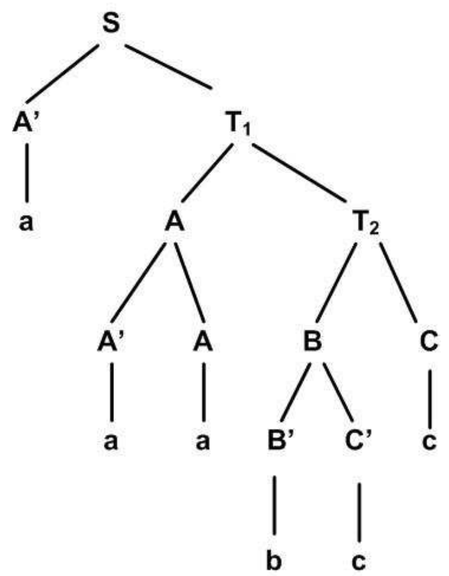
The given grammar in Chomsky normal form is:

$$
\begin{aligned}
&S \rightarrow A'T_1 \mid a \\
&T_1 \rightarrow AT_2 \\
&T_2 \rightarrow BC \\
&A \rightarrow A'A \mid a \\
&B \rightarrow B'T_3 \mid B'C' \\
&T_3 \rightarrow C'B \\
&C \rightarrow C'C \mid c \\
&A' \rightarrow a \\
&B' \rightarrow b \\
&C' \rightarrow c
\end{aligned}
$$

---

5. **(5 points)** Construct the parse tree of the string *aaabcc* from the new grammar in Chomsky normal form. Show your work.

Solution:



We include here the leftmost derivation of *aaabcc* also (this is not required though):

$$
\begin{aligned}
S &\Rightarrow A'T_1 \\
&\Rightarrow aT_1 \\
&\Rightarrow aAT_2 \\
&\Rightarrow aA'AT_2 \\
&\Rightarrow aaAT_2 \\
&\Rightarrow aaaT_2 \\
&\Rightarrow aaaBC \\
&\Rightarrow aaaB'C'C \\
&\Rightarrow aaabC'C \\
&\Rightarrow aaabcC \\
&\Rightarrow aaabcc
\end{aligned}
$$

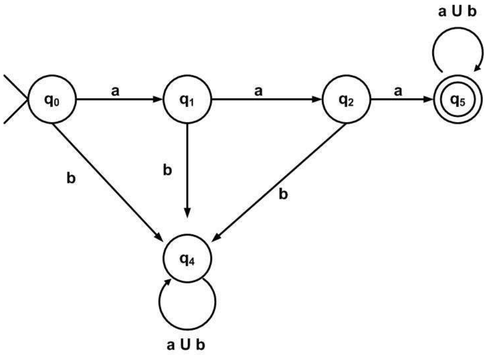
PROBLEM 3: Deterministic Finite Automata (25 points)

Give the state diagram of a deterministic finite automaton (DFA) that accepts the set of strings over $\{a, b\}$ that do NOT begin with the substring aaa.

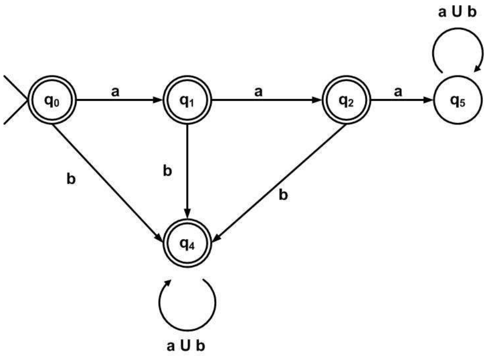
> **Examples:** The strings *aababaaaaaba*, *abaaababbab*, and *baababbab* belong to this language, but <u>*aaa*</u>, <u>*aaa*</u>*aba*, and <u>*aaa*</u>*bab* do not.

Solution:

Let's start by constructing a DFA that accepts the complement of the desired laguage. That is, a DFA that accepts the strings that start with aaa:



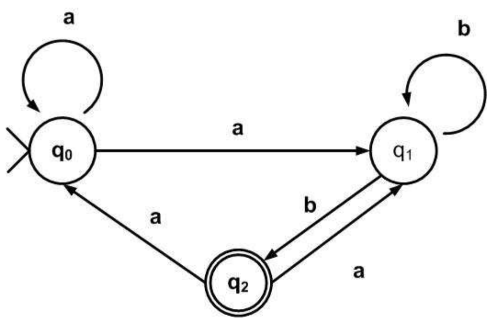
Now, to obtain the DFA that accepts strings only if they don't start with aaa, we take the complement of the set of accepting states from the DFA above:



(Needless to say, you can construct a DFA for this language directly without constructing a DFA that accepts the complement of this language first.)

PROBLEM 4: Nondeterministic Finite Automata (25 points + 5 bonus points)

Let M be the nondeterministic finite automaton over $\Sigma = \{a, b\}$:



1. **(10 points)** Construct the transition table of M.

Solution:

This problem was solved in Homework 2 CS3133 A term 2009. We reproduce the answer here.

δ	a	b
q_0	$\{q_0, q_1\}$	$\emptyset$
q_1	$\emptyset$	$\{q_1, q_2\}$
q_2	$\{q_0, q_1\}$	$\emptyset$

---

2. Note that the NFA M does NOT contain any λ–transitions. Hence, for each state q in M and each input symbol d in Σ, λ–closure$(q) = \{q\}$, and the input transition function is the same as the δ function: $t(q, d) = \delta(q, d)$.

3. **(15 points)** Follow the algorithm discussed in class and in the textbook to construct the set of states and transitions of a DFA that is equivalent to M. **Show your work.**

Solution:

State	**Symbol**	**Next State**
λ–closure$(\{q_0\})$	a	$\delta(q_0, a) = \{q_0, q_1\}$
$= \{q_0\}$	b	$\delta(q_0, b) = \emptyset$
$\{q_0, q_1\}$	a	$\delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
	b	$\delta(q_0, b) \cup \delta(q_1, b) = \emptyset \cup \{q_1, q_2\} = \{q_1, q_2\}$
$\{q_1, q_2\}$	a	$\delta(q_1, a) \cup \delta(q_2, a) = \emptyset \cup \{q_0, q_1\} = \{q_1, q_2\}$
	b	$\delta(q_1, b) \cup \delta(q_2, b) = \{q_1, q_2\} \cup \emptyset = \{q_1, q_2\}$

---

4. **(5 points)** Draw a state diagram to depict the states and transitions of the DFA constructed above. **Remember to explicitly mark the accepting states of the DFA.**

Solution: