# Bachelor in Applied Computer Science<br>Collection of Exam Exercises for<br>**Formal Languages**

A.Y. 2008/2009

Diego Calvanese
Faculty of Computer Science
Free University of Bozen-Bolzano
`calvanese@inf.unibz.it`

May 6, 2009

This document contains a collection of exercises on Formal Languages taken from exams of previous years. The exercises cover the whole program of the Formal Languages course of the Bachelor in Applied Computer Science. All exercises, except possibly for some in Section 7, can be solved in a straightforward way by applying the standard techniques and algorithms that are taught in the course, and that are covered in the textbook *Introduction to Automata Theory, Languages, and Computation* (3rd edition), by J.E. Hopcroft, R. Motwani, and J.D. Ullman, Addison Wesley, 2007.

The exercises in Section 7 are slightly more advanced, and some of them may require an intuition or a deeper understanding of the subject matter. Such exercises are intended for interested students, who want to deepen their knowledge of the subject. They are not representative of exercises that students might be confronted with at the Formal Languages exam.

A rough estimate of the expected time to solve each exercise is 3 minutes per assigned point, e.g., an exercise of 6 points should be solved in approximately 18 minutes. This corresponds to solving in 90 minutes exercises that total 30 points, which make up a typical exam. The Formal Languages exam is a closed book exam, i.e., the only resources allowed are blank paper, pens, and one's brain. Hence, for maximum effectiveness, it is suggested to solve at least some substantial part of the exercises under the same conditions. A complete solution to an exercise should always contain a sufficiently clear explanation of the reasoning that has brought to the solution. Such an explanation might either be given explicitly, in particular when requested, or implicitly, by detailing the steps of an algorithm that has brought to the solution. In any case, the clarity of the explanations will affect the evaluation given to an exercise, and hence the overall final grade for the exam.

# 1 Properties of Languages

**Problem 1.1** [2 points each] Decide which of the following statements holds for *all* languages $L_1$ and $L_2$, and which does not hold. To show that a statement does not hold for *all* languages, you should give two example languages $L_1$ and $L_2$ for which the statement is false. When a statement holds, you should give a brief explanation of why this is the case.

(1) $L_1^* \cup L_2^* = (L_1 \cup L_2)^*$

(2) $L_1^* \cap L_2^* = (L_1 \cap L_2)^*$

(3) $(L_1^* \cap L_2^*)^* = (L_1 \cap L_2)^*$

(4) $(L_1^* \cdot L_2^*)^* = (L_1 \cup L_2)^*$

(5) $(L_1 \cdot L_2)^* = (L_1 \cup L_2)^*$

(6) $((L_1 \cup \{\varepsilon\}) \cdot (L_2 \cup \{\varepsilon\}))^* = (L_1 \cup L_2)^*$

(7) $(L_1^* \cdot L_2^*)^+ = (L_1^+ \cdot L_2^+)^*$

**Problem 1.2** [2 points each] Decide which of the following statements is TRUE and which is FALSE. You must give a brief explanation of your answer to receive full credit.

(1) If $L_1$ is regular and $L_2$ is non-regular, then $L_1 \cdot L_2$ is non-regular.

(2) If $L_1$ is regular and $L_2$ is non-regular, then $L_1 \cap L_2$ is regular.

(3) If $L_1$ is regular and $L_2$ is non-regular, then $L_1 \cup L_2$ is non-regular.

(4) If $L_1$ is non-regular and $L_2$ is non-regular, then $L_1 \cup L_2$ can be regular.

(5) If $L_1$ is non-regular and $L_2$ is non-regular, then $L_1 \cup L_2$ is non-regular.

(6) If $L_1$ is regular and $L_2$ is non-regular, then $L_1 \cdot L_2$ can be regular.

(7) If $L_1$ is regular and $L_1 \cup L_2$ is regular, then $L_2$ is regular.

(8) If $L_1$ is regular and $L_2 \subseteq L_1$, then $L_2$ is regular.

(9) If $L_1$ is non-regular and $L_1 \subseteq L_2$, then $L_2$ is non-regular.

(10) If $L_1$ is regular and $L_2$ is regular, then $L_1 \setminus L_2$ is regular.

(11) If $L \setminus \{\varepsilon\}$ is regular, then $L$ is regular.

(12) If $L^*$ is regular, then $L$ is regular.

**Problem 1.3** [2 points each] Decide which of the following statements is TRUE and which is FALSE. You must give a brief explanation of your answer to receive full credit.

(1) There exists a language $L$ such that $L$ is not regular but $L^*$ is regular.

(2) There exists a language $L$ such that $L^*$ is not regular but $(L^*)^*$ is regular.

(3) There exists a language $L$ such that $L = L \cdot L$.

(4) There exists a language $L$ with $\varepsilon \notin L$ such that $L = L^*$.

(5) For all languages $L$, we have that $L^* = (L \cup \varepsilon)^*$.

(6) For all regular languages $L_1$, $L_2$, and $L_3$, if $L_1 \subseteq L_2$ and $L_2^* \subseteq L_3^*$, then $L_1 \subseteq L_3$.

(7) For all languages $L_1$ and $L_2$, if $L_1 \subseteq L_2$ and $L_2^* \subseteq L_1^*$, then $L_1 = L_2$.

(8) For all languages $L_1$ and $L_2$, if $L_1 \subseteq L_2$, then $L_1^* \subseteq L_2^*$.

(9) For all languages $L_1$ and $L_2$, if $L_1 \subsetneq L_2$, then $L_1^* \subsetneq L_2^*$. [N.B. $A \subsetneq B$ means $A \subseteq B$ and $A \neq B$]

(10) For all languages $L_1$ and $L_2$, if $L_1^* \subseteq L_2^*$, then $L_1 \subseteq L_2$.

(11) For all languages $L_1$ and $L_2$, if $L_1^* = L_2^*$, then $L_1 = L_2$.

(12) For all languages $L_1$ and $L_2$, if $L_1 \cap L_2 = \emptyset$, then either $L_1 = \emptyset$ or $L_2 = \emptyset$.

(13) For all languages $L_1$ and $L_2$, if $L_1 \cap L_2 = \emptyset$ and $L_1 \cup L_2 = \Sigma^*$, then $L_1 = \overline{L_2}$.

(14) For all languages $L_1$ and $L_2$, if $L_1 = \overline{L_2}$, then $L_1 \cap L_2 = \emptyset$ and $L_1 \cup L_2 = \Sigma^*$.

(15) If $L$ is constituted by a *finite* set of strings, then $L$ is a regular language.

(16) There exist nonempty languages $L_1$ and $L_2$, with $L_1 \neq \{\varepsilon\}$, $L_2 \neq \{\varepsilon\}$, and $L_1 \neq L_2$, such that $L_1 \cdot L_2 = L_2 \cdot L_1$.

(17) A regular expression denotes an infinite language if and only if it contains the $*$ operator.

**Problem 1.4** [2 points each] Decide which of the following statements is TRUE and which is FALSE. You must give a brief explanation of your answer to receive full credit.

(1) If $L$ is not of type 2 (i.e., not context-free), then it is not of type 3 (i.e., not regular).

(2) If $L_1$ is context free and $L_1 \subseteq L_2$, then $L_2$ is non-regular.

(3) If $L_1$ is regular and $L_2$ is context-free, then $L_1 \cap L_2$ is regular.

(4) If $L$ is context-free, then $L \setminus \{\varepsilon\}$ is context-free.

**Problem 1.5** [2 points each] For each of the following languages, construct a regular expression that generates it:

(1) the set of binary strings that have 101 or 010 (or both) as substring;

(2) the set of binary strings that have both 00 and 11 as substrings;

(3) the set of binary strings that have 00 but not 11 as substrings;

(4) the set of strings over the alphabet $\{a, b, c\}$ that contain the substring $aa$ starting at an odd position and the substring $bb$ starting at an even position;

(5) the set of strings over the alphabet $\{x, y, z\}$ that begin with $z$ and end with a sequence of two or more $y$'s;

(6) the set of strings over the alphabet $\{0, 1, 2\}$ that contain an even number of 2's;

(7) the set of strings over the alphabet $\{x, y, z\}$ that contain an odd number of $y$'s;

(8) the set of strings over the alphabet $\{x, y, z\}$ in which each $y$ is immediately followed by $x$;

(9) the set of strings over the alphabet $\{a, b, c\}$ in which each $b$ is immediately preceded by $a$;

**Problem 1.6** [2 points each] For each of the following languages, construct a **deterministic** finite automaton (DFA) that accepts it:

(1) the set of strings over the alphabet $\{a, b, c\}$ that begin with a sequence of one or more $a$'s, and end with a $b$;

(2) the set of strings over the alphabet $\{0, 1, 2\}$ in which each 1 is immediately followed by 2;

(3) the set of strings over the alphabet $\{0, 1, 2\}$ in which each 2 is immediately preceded by 1;

(4) the set of strings over the alphabet $\{x, y, z\}$ in which each $x$ is immediately preceded or immediately followed (or both) by $y$;

**Problem 1.7** [2 points] Simplify as much as possible the regular expression

$$E = (((a + b)^* \cdot ((b \cdot \emptyset) + \varepsilon))^* + (b + a)^*) + \varepsilon$$

Motivate each simplification step you have applied by an algebraic law for regular expressions.

**Problem 1.8** [2 points each]

(a) Show that $L^* = L \cdot L^*$ if and only if $\varepsilon \in L$.

(b) Give a necessary and sufficient condition for a language $L$ to satisfy the equation $L^+ = L^*$.

**Problem 1.9** [2 points] Explain what is wrong in the following argument: "Let $L$ be a language that is not regular. Since regular languages are closed under the $*$ operator, we have that also $L^*$ is not regular."

**Problem 1.10** [4 points] Consider the language $L = \{x0^n y1^n z \mid n \geq 0, x \in L_1, y \in L_2, z \in L_3\}$, where $L_1$, $L_2$, $L_3$ are nonempty languages over the alphabet $\{0, 1\}$.

(a) Find nonempty regular languages $L_1$, $L_2$, $L_3$ such that $L$ is regular.

(b) Find nonempty regular languages $L_1$, $L_2$, $L_3$ such that $L$ is not regular.

**Problem 1.11** [8 points] Describe in detail an algorithm to solve the following problem: Given a regular expression $E$ with associated language $\mathcal{L}(E)$ over the alphabet $\Sigma$, construct a regular expression $\overline{E}$ such that $\mathcal{L}(\overline{E}) = \Sigma^* \setminus \mathcal{L}(E)$. (Notice that set difference is not an operator that can be used in a regular expressions.)

Illustrate the algorithm on the example of the regular expression $0^* \cdot 1^*$.

**Problem 1.12** [8 points] Describe an algorithm to solve the following problem: Given two regular expressions $E_1$ and $E_2$, respectively with associated languages $\mathcal{L}(E_1)$ and $\mathcal{L}(E_2)$ over the alphabet $\Sigma$, construct a regular expression $E$ such that $\mathcal{L}(E) = \mathcal{L}(E_1) \cap \mathcal{L}(E_2)$. (Notice that set intersection is not an operator that can be used in regular expressions.) In describing the algorithm, you can make use of algorithms that have been discussed in class, without the need of detailing the various steps of these algorithms.

Illustrate the algorithm on the example of the regular expressions $E_1 = 0^*$ and $E_2 = 0$. Notice that $E_1$ and $E_2$ are sufficiently simple to allow you to calculate on them the results of the algorithms discussed in class, without the need of detailing the various steps of these algorithms.

**Problem 1.13** [8 points] Describe an algorithm to solve the following problem: Given two regular expressions $E_1$ and $E_2$, respectively with associated languages $\mathcal{L}(E_1)$ and $\mathcal{L}(E_2)$ over the alphabet $\Sigma$, construct a regular expression $E$ such that $\mathcal{L}(E) = \mathcal{L}(E_1) \setminus \mathcal{L}(E_2)$. (Notice that neither set difference, nor set intersection, nor set complement are operators that can be used in regular expressions.) In describing the algorithm, you can make use of algorithms that have been discussed in class, without the need of detailing the various steps of these algorithms.

Illustrate the algorithm on the example of the regular expressions $E_1 = 1^*$ and $E_2 = 1$. Notice that $E_1$ and $E_2$ are sufficiently simple to allow you to calculate on them the results of the algorithms discussed in class, without the need of detailing the various steps of these algorithms.

# 2 Regular Expressions and Finite State Automata

**Problem 2.1** [6 points each] For each of the following regular expressions $E$ do the following: Construct an $\varepsilon$-NFA $A_\varepsilon$ such that $\mathcal{L}(A_\varepsilon) = \mathcal{L}(E)$. Try to simplify intermediate results whenever possible. Then, by eliminating $\varepsilon$-transitions from $A_\varepsilon$, construct an NFA $A$ such that $\mathcal{L}(A) = \mathcal{L}(A_\varepsilon)$. Illustrate the steps of the algorithm you have followed to construct $A_\varepsilon$ and $A$.

(1) $E = 1 \cdot (0^* + 0 \cdot 1)^*$

(2) $E = (b + a)^* + (a \cdot b)^*$

(3) $E = ((1 \cdot 0)^* \cdot 0)^* + (1 \cdot 1)$

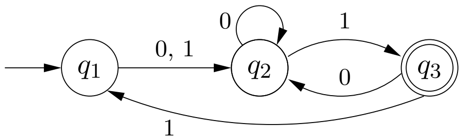
(4) $E = ((a \cdot b) + (b + c)^*)^*$

(5) $E = ((0 \cdot 1) + (1 \cdot 0))^* \cdot 1^*$

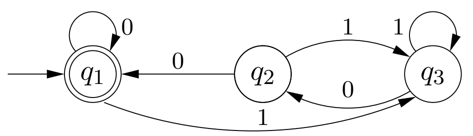
(6) $E = 0^* \cdot ((1 \cdot 0) + (0 \cdot 1))^*$

**Problem 2.2** [6 points each] For each of the following DFAs or NFAs $A$, construct a regular expression $E$ such that $\mathcal{L}(E) = \mathcal{L}(A)$. Illustrate the steps of the algorithm you have followed to construct $E$. For each automaton $A$, give 3 strings (of length at least 4) that are in $\mathcal{L}(A)$ and 3 strings (of length at least 4) that are not in $\mathcal{L}(A)$.

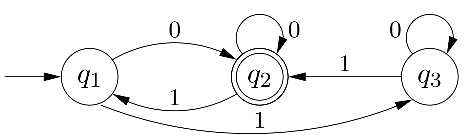
(1)



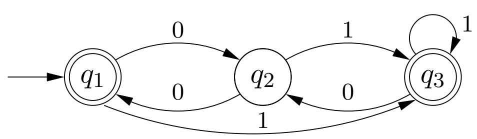

(2)



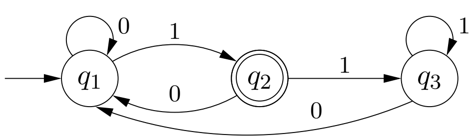

(3)



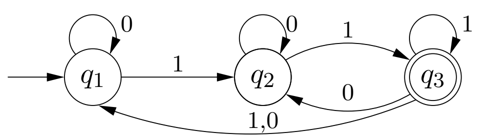

(4)



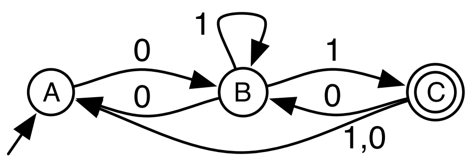

(5)




(6)



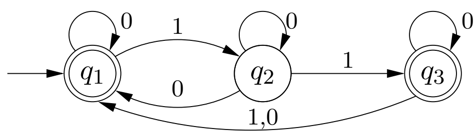

(7)



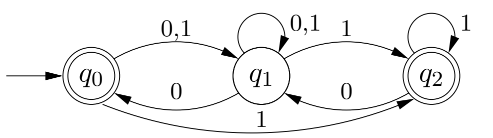
# 3 Transformations between Finite State Automata

**Problem 3.1** [6 points each] For each of the following NFAs $N$, construct a DFA $A$ such that $\mathcal{L}(A) = \mathcal{L}(N)$. Illustrate the steps of the algorithm you have followed to construct $A$.

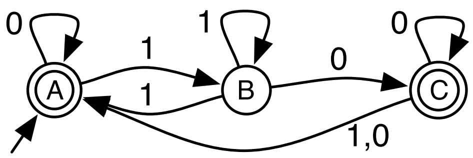
(1)



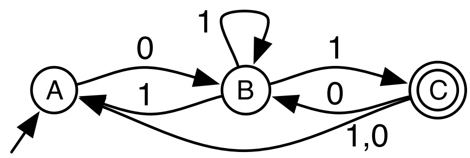

(2)



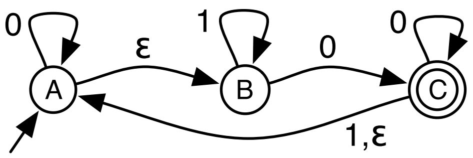

(3)




(4)




**Problem 3.2** [6 points] Consider the following $\varepsilon$-NFA $N_1$ over $\{0, 1\}$:



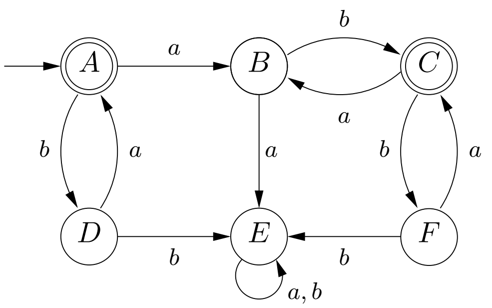

(a) Construct an NFA $N_2$ such that $\mathcal{L}(N_2) = \mathcal{L}(N_1)$. The algorithm you have followed to construct $N_2$ should become evident in your construction.

(b) Show **all** sequences of transitions of $N_1$ and of $N_2$ that lead to acceptance of 0010.

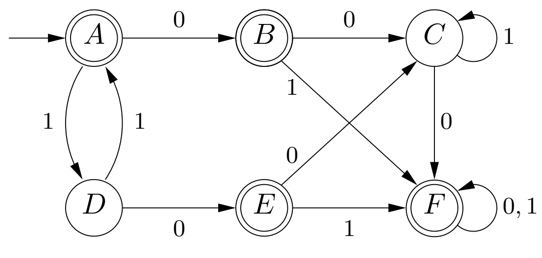
# 4 Minimization of Deterministic Finite State Automata

**Problem 4.1** [6 points each] Consider the following DFAs $A$. Construct for each of them a DFA $A_m$ with minimal number of states such that $\mathcal{L}(A_m) = \mathcal{L}(A)$. Illustrate the steps of the algorithm you have followed to construct $A_m$.

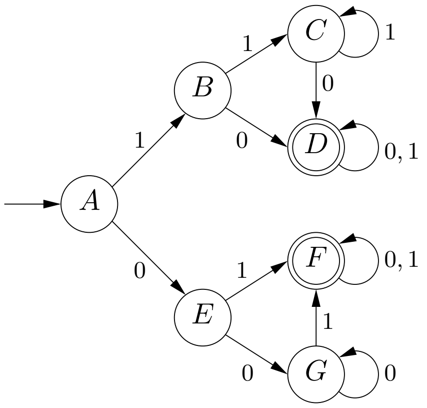
(1)



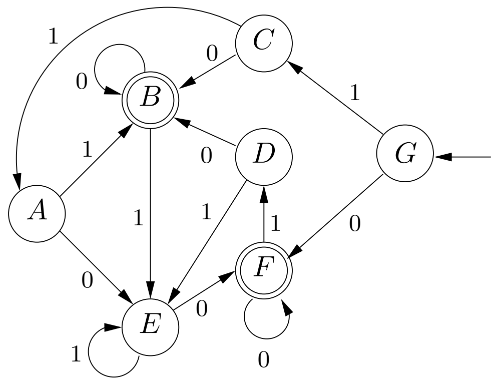

(2)



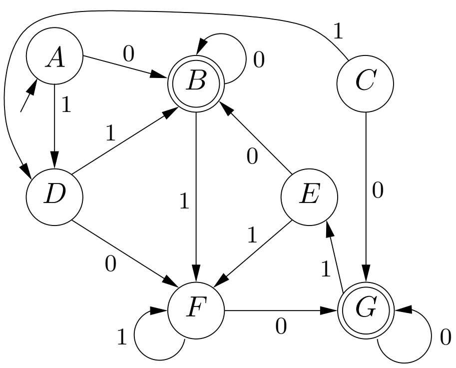
(3)



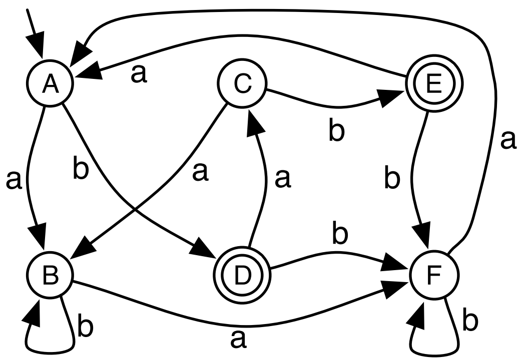

(4)



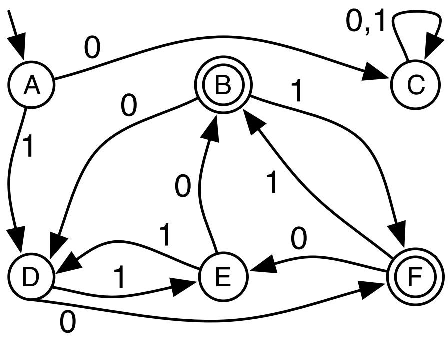

(5)




(6)




(7)




**Problem 4.2** [2 points each] For each of the automata in Problem 4.1, give 3 strings (of length at least 4) that are in $\mathcal{L}(A)$ and 3 strings (of length at least 4) that are not in $\mathcal{L}(A)$. Provide a description of $\mathcal{L}(A)$ in plain English.

# 5 Showing Languages to be Non-regular

**Problem 5.1** [6 points each] In each of the following cases, show that the language $L$ is not regular by exploiting the pumping lemma for regular languages:

(a) $L = \{ a^{k^2} \mid k \geqslant 0 \}$

(b) $L = \{ a^n b^m \mid n \geqslant m \}$

(c) $L = \{ a^i b^j \mid i, j \geqslant 0, i \neq j \}$

[*Hint*: Exploit in your argument closure properties of regular languages and the known facts that the language $L_1 = \{a^i b^j \mid i, j \geqslant 0\}$ is regular and the language $L_2 = \{a^n b^n \mid n \geqslant 0\}$ is not regular.]

(d) $L = \{a^m b^n c^k \mid m, n, k \geq 0, m \neq n \text{ or } m \neq k \text{ or } n \neq k\}$

[*Hint*: Consider first the language $\overline{L} \cap a^* b^* c^*$ and show that it is not regular using the pumping lemma. Then exploit the closure properties of regular languages.]

(e) $L = \{\ xy \mid x, y \in \{0,1\}^* \text{ and } \#_0(x) \geq \#_0(y)\ \}$,
where $\#_0(w)$ denotes the number of 0's in a string $w$.

[*Hint*: Consider e.g., the string $0^n 1^{2n} 0^n$, for a suitable value of $n$.]

(f) $L = \{\ a^n b^m \mid n \leq m \leq 2n\ \}$

# 6 Context Free Languages and Grammars

**Problem 6.1** [4 points each] In each of the following cases, show that the language $L$ is context free by exhibiting a context free grammar that generates it. Be precise in the specification of the grammar, by providing explicitly all its elements.

(1) $L = \{uawb \mid u, w \in \{a, b\}^*, \text{ with } |u| = |w|\}$

(2) $L = \{a^m b^n c^p d^q \mid m + n = p + q\}$

(3) $L = \{a^m b^n \mid m, n \geq 0, m \neq n\}$

**Problem 6.2** [6 points] Consider the language $L$ over $\Sigma = \{0, 1, \#\}$ defined as follows:

$$L = \{x^R \# y \mid x, y \in \{0,1\}^*,\ x \text{ is a substring of } y\}$$

where $x^R$ denotes the reverse string of $x$.

(a) Show that $L$ is context free by exhibiting a context free grammar $G$ that generates $L$. Be precise in the specification of the grammar, by providing explicitly all its elements.

(b) Show the leftmost derivation according to $G$ for the string 110#001110 and for the string 10#1010011. Draw the corresponding parse trees.

(c) Is the grammar you have provided ambiguous? Argue convincingly.

**Problem 6.3** [6 points]

(a) Describe the algorithm to eliminate the $\varepsilon$-productions from a context free grammar.

(b) Describe the algorithm to eliminate non-generating symbols from a context free grammar.

Apply first algorithm (a) and then algorithm (b) to the grammar $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & C \mid bAAa \mid CaA \\ A & \longrightarrow & Aa \mid CB \mid \varepsilon \end{array} \qquad \begin{array}{lcl} B & \longrightarrow & CB \mid BA \mid Ba \\ C & \longrightarrow & Ca \mid CB \mid \varepsilon \end{array}$$

**Problem 6.4** [6 points]

(a) Describe the algorithm to eliminate the $\varepsilon$-productions from a context free grammar.

(b) Describe the algorithm to eliminate the non-reachable symbols from a context free grammar.

Apply first algorithm (a) and then algorithm (b) to the grammar $G = (\{S, A, B, C, D\}, \{a, b, c\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & B \mid BaBb \mid DbB \\ A & \longrightarrow & DC \mid Aac \mid Bbc \\ B & \longrightarrow & Bb \mid DC \mid \varepsilon \end{array} \qquad \begin{array}{lcl} C & \longrightarrow & DC \mid CB \mid Ba \\ D & \longrightarrow & Db \mid DC \mid \varepsilon \end{array}$$

**Problem 6.5** [6 points]

(a) Describe the algorithm to eliminate the unit-productions from a context free grammar.

(b) Describe the algorithm to eliminate the non-generating symbols from a context free grammar.

Apply first algorithm (a) and then algorithm (b) to the grammar $G = (\{S, A, B, C, D\}, \{a, b, c\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & A \mid AaBb \mid Da \\ A & \longrightarrow & Aa \mid C \mid D \\ B & \longrightarrow & BD \mid Cc \mid C \end{array} \qquad \begin{array}{lcl} C & \longrightarrow & AB \mid aB \mid Ca \\ D & \longrightarrow & c \mid aA \end{array}$$

**Problem 6.6** [6 points each] Describe the steps, in the correct order, that are necessary to convert a context free grammar into Chomsky Normal Form. Apply these steps to each of the following grammars $G$.

(1) $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & B \mid BaBb \mid AbB \\ A & \longrightarrow & Ab \mid AC \end{array} \qquad \begin{array}{lcl} B & \longrightarrow & Bb \mid AC \mid \varepsilon \\ C & \longrightarrow & AC \mid CB \mid Ba \end{array}$$

(2) $G = (\{S, A, B, C, D\}, \{a, b\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & aAa \mid bBb \mid \varepsilon \\ A & \longrightarrow & C \mid a \\ B & \longrightarrow & C \mid b \end{array} \qquad \begin{array}{lcl} C & \longrightarrow & CDE \mid \varepsilon \\ D & \longrightarrow & A \mid B \mid ab \end{array}$$

(3) $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & aAb \mid bBa \mid AA \\ A & \longrightarrow & S \mid B \end{array} \qquad \begin{array}{lcl} B & \longrightarrow & C \\ C & \longrightarrow & S \mid \varepsilon \end{array}$$

(4) $G = (\{S, A\}, \{0\}, P, S)$, where $P$ consists of the following productions:

$$S \longrightarrow ASA \mid A \mid \varepsilon \qquad A \longrightarrow 00 \mid \varepsilon$$

(5) $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & A \mid ABa \mid AbA \\ A & \longrightarrow & Aa \mid \varepsilon \end{array} \qquad \begin{array}{lcl} B & \longrightarrow & Bb \mid BC \\ C & \longrightarrow & CB \mid CA \mid bB \end{array}$$

(6) $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & ABCa \mid aAbb \mid \varepsilon \\ A & \longrightarrow & \varepsilon \end{array} \qquad \begin{array}{lcl} B & \longrightarrow & bB \mid b \mid AC \\ C & \longrightarrow & aCa \mid \varepsilon \end{array}$$

(7) $G = (\{S, A, B, C, D, E\}, \{a, b\}, P, S)$, where $P$ consists of the following productions:

$$\begin{array}{lcl} S & \longrightarrow & AB \mid BBB \mid C \\ A & \longrightarrow & Ab \mid DA \mid EaE \\ B & \longrightarrow & aa \mid bB \mid C \end{array} \qquad \begin{array}{lcl} C & \longrightarrow & \varepsilon \\ D & \longrightarrow & aCa \mid \varepsilon \\ E & \longrightarrow & Eb \mid AaA \end{array}$$

# 7 Advanced Exercises

The following exercises are optional, and go beyond what has been covered in the Formal Languages course. They are intended for interested students, who want to deepen their knowledge of the subject.

**Problem 7.1** [6 points] The *quotient* $L_1/L_2$ of two languages $L_1$ and $L_2$ is defined as

$$L_1/L_2 \quad = \quad \{\ x \mid \text{ there is } y \in L_2 \text{ such that } xy \in L_1\ \}.$$

For example, if

$$\begin{array}{rcl} L_1 & = & \{\ w \in \{0,1\}^* \mid w \text{ has an even number of 0's }\}, \\ L_2 & = & \{\ 0\ \}, \\ L_3 & = & \{\ 0, 00\ \}, \end{array}$$

then

$$\begin{array}{rcl} L_1/L_2 & = & \{\ w \in \{0,1\}^* \mid w \text{ has an odd number of 0's }\}, \\ L_1/L_3 & = & \{\ 0, 1\ \}^*. \end{array}$$

Show that, for an *arbitrary* language $L_2$, if $L_1$ is regular, then $L_1/L_2$ is also regular.

[*Hint*: Start from a DFA $A$ for $L_1$, and show how to modify the set of final states of $A$ to obtain a DFA for $L_1/L_2$.]

**Problem 7.2** [6 points] Let $L_1$ be the set of strings over $\{a, b\}$ that do *not* have $aab$ as a substring. Let further $L_2$ be the language over $\{a, b\}$ inductively defined as follows:

1. $\varepsilon$ is in $L_2$;
2. for every $w$ in $L_2$, also $wa$, $bw$, and $abw$ are in $L_2$;
3. nothing else is in $L_2$.

(a) Prove that $L_2 \subseteq L_1$, making all steps of the proof explicit. [*Hint:* use structural induction on the rules used to define $L_2$.]

(b) Prove that $L_1 \subseteq L_2$, making all steps of the proof explicit. [*Hint:* use induction on the length of a string in $L_1$.]

**Problem 7.3** [6 points] A *nonrestarting DFA* is a DFA $(Q, \Sigma, \delta, q_0, F)$ such that the initial state $q_0$ has no incoming transition, i.e., $\delta(q, a) \neq q_0$, for all $q \in Q$ and all $a \in \Sigma$. Prove that for every regular language $L$ there is an effective way to construct a nonrestarting DFA $D_{nr}$ such that $\mathcal{L}(D_{nr}) = L$.

[*Hint*: Describe first how to construct, from an arbitrary DFA $D$, a suitable nonrestarting DFA $D_{nr}$, and then show, by induction, that the DFA $D_{nr}$ that you have constructed is such that (1) each string accepted by $D$ is also accepted by $D_{nr}$, and (2) each string accepted by $D_{nr}$ is also accepted by $D$.]

**Problem 7.4** [4 points] Consider the grammar $G = (\{S, T\}, \{0, 1\}, P, S)$, where $P$ consists of the following productions

$$\begin{array}{rcl} S & \longrightarrow & 0S \mid 1T \mid 0 \\ T & \longrightarrow & 1T \mid 1 \end{array}$$

Show that no string in the language $\mathcal{L}(G)$ contains the substring 10.

**Problem 7.5** [6 points] Consider the grammar $G = (\{S, A, B\}, \{0, 1\}, P, S)$, where $P$ consists of the following productions

$$\begin{array}{rcl} S & \longrightarrow & A \mid B \\ A & \longrightarrow & 0A \mid AA1 \mid 0 \\ B & \longrightarrow & B1 \mid 0BB \mid 1 \end{array}$$

Prove that in every word of the language $\mathcal{L}(G)$ the number of 0's and the number of 1's are different.

**Problem 7.6** [6 points] Consider a context-free grammar $G = (V_N, V_T, P, S)$ in which every production in $P$ is of the form $A \longrightarrow xB$ or $A \longrightarrow x$, with $A, B \in V_N$ and $x \in V_T^*$. Show that the language generated by $G$ is regular. Does this still hold if we allow in $G$ also productions of the form $A \longrightarrow Bx$? Argue convincingly.

**Problem 7.7** [4 points] A context-free grammar $G = (V_N, V_T, P, S)$ is said to be *linear* if every production in $P$ is of the form $A \longrightarrow xB$ or $A \longrightarrow Bx$ or $A \longrightarrow x$, where $A, B \in V_N$ and $x \in V_T^*$. Show that the language generated by a linear grammar is not necessarily regular.

**Problem 7.8** [6 points each] Let $L$ be the language generated by the grammar $G = (\{S\}, \{a, b\}, P, S)$, where $P$ consists of the following productions

$$S \longrightarrow \varepsilon \mid Sa \mid bS \mid abS$$

(a) Prove that *no* string generated by $G$ has $aab$ as a substring. Make all steps of the proof explicit.

[*Hint:* Use an induction on the length of the derivation according to $G$ of a sentential form $w_1 S w_2$, establishing properties that hold for $w_1$ and $w_2$.]

(b) Prove that each string that does not have $aab$ as a substring is generated by $G$, making all steps of the proof explicit.

[*Hint:* use induction on the length of the string $w$, and distinguish different cases according to the first two symbols or the last symbol of $w$.]

**Problem 7.9** [6 points] Consider the context-free grammar $G = (V_N, V_T, P, S)$ with $V_N = \{S, X, Y\}$, $V_T = \{a, b\}$, and $P$ constituted by the following productions:

$$\begin{array}{lllllllll} S & \rightarrow & XY & \quad X & \rightarrow & aX & \quad Y & \rightarrow & bY \\ & & & \quad X & \rightarrow & \varepsilon & \quad Y & \rightarrow & \varepsilon \end{array}$$

(a) Prove that $\mathcal{L}(G)$ is a regular language.

(b) Prove that there is no leftmost derivation of the sentential form $X$ (even though $S \stackrel{*}{\Rightarrow} X$).