General notes

Conventions used throughout this exam

- $\Sigma_{\text{bin}} = \{\texttt{0}, \texttt{1}\}$.
- $\Sigma_{\text{abc}} = \{\texttt{a}, \texttt{b}, \texttt{c}\}$.
- Σ_{ASCII} is the ASCII alphabet.
- $\Sigma_{\text{cap}} \subset \Sigma_{\text{ASCII}}$ is the set of all capitals.
- $\Sigma_{\text{dig}} \subset \Sigma_{\text{ASCII}}$ is the set of all decimal digits.
- "Formally describe a language" means that a formal language is to be described in a set-theoretic way, e.g.

$$L = \{\, w \in \Sigma_{\text{bin}}^* \mid |w| < 5\}$$

- *Binary numbers* are arbitrary words of Σ_{bin}^* containing at least one symbol, however, they must not begin with `0` unless the word is `0` itself.
- If not noted otherwise, regular expressions are based on the operator set $\{+, \cdot, ^*, (,)\}$, and the concatenation operator $\cdot$ can be omitted.

Algebraic operations on regular expressions

1. $r_1 + r_2 \doteq r_2 + r_1$ (commutative law)
2. $(r_1 + r_2) + r_3 \doteq r_1 + (r_2 + r_3)$ (associative law)
3. $(r_1 r_2) r_3 \doteq r_1 (r_2 r_3)$ (associative law)
4. $\emptyset r \doteq \emptyset$
5. $\varepsilon r \doteq r$
6. $\emptyset + r \doteq r$
7. $(r_1 + r_2) r_3 \doteq r_1 r_3 + r_2 r_3$ (distributive law)
8. $r_1 (r_2 + r_3) \doteq r_1 r_2 + r_1 r_3$ (distributive law)
9. $r + r \doteq r$
10. $(r^*)^* \doteq r^*$
11. $\emptyset^* \doteq \varepsilon$
12. $\varepsilon^* \doteq \varepsilon$
13. $r^* \doteq \varepsilon + r^* r$

14. $r^* \doteq (\varepsilon + r)^*$

15. $\{r \doteq rs + t \quad \text{with} \quad \varepsilon \notin L(s)\} \longrightarrow r \doteq ts^*$ (proof by Arto Salomaa)

16. $r^* r \doteq r r^*$

17. $r^* \doteq \varepsilon + r^*$

1 Formal languages (6 pts)

Formally describe the following languages:

a) all words $w \in \Sigma^*_{\text{ASCII}}$ containing at least three instances of the string `AC/DC`,

b) all binary numbers divisible by `11` (binary),

c) an IP address (ignore the fact that the four decimal numbers contained in an IP address must not exceed 255).

2 Regular expressions (9 pts)

a) Give a regular expression for all the words $w \in \Sigma^*_{\text{abc}}$ containing exactly two instances of the character `a` and at least one instance of `b`.

b) Give a regular expression for all the words $w \in \Sigma^*_{\text{abc}}$ containing the substring `ca` exactly once.

c) List all the words of the language $L = L_1 \cap L_2$ with

$$\begin{aligned} L_1 &= \{w \in \Sigma^*_{\text{bin}} | |w| \leq 3\}, \\ L_2 &= L(\texttt{00}^{**} + (\texttt{1}^{**}\texttt{0}) + \texttt{0}\varepsilon\texttt{0} + \texttt{00}^*(\texttt{00}(\texttt{01}))). \end{aligned} \tag{1}$$

3 Algebraic operations (8 pts)

For this section, use the algebraic operations listed in the General Notes section above.

a) Simplify as much as possible:

$$r = (\varepsilon^* + (\varepsilon + \texttt{0}^*\texttt{0} + (\varepsilon\texttt{0}^*)^*\texttt{0} + \emptyset)^*(\varepsilon + (\texttt{0}^*)^*) + \emptyset\texttt{1} + (\varepsilon^*)^*)^* + \varepsilon\texttt{1} + \varepsilon + \texttt{1} + \emptyset\texttt{0}$$

b) Prove that

$$(s + t)^* \doteq (s + t)^* s + (s + t)^*.$$

In subtask b), indicate which rules you are using by referring to their ID (1-17) at every step you take.

4 JFlex (6 pts)

Give regular expressions using the JFlex operator set matching

a) all input lines containing an IP address [see Task 2c) for further instructions],

a) all input lines containing the word `two` at least twice,

b) all input lines containing the word `two` at most twice.

5 Transforming NFAs into DFAs (8 pts)

Convert the following NFA into a minimized DFA:

$$A = \langle Q, \Sigma, \delta, q_0, F\rangle$$

with

1. $Q = \{0, 1, 2, 3\}$
2. $\Sigma = \Sigma_{\mathrm{abc}}$
3. $\delta = \{\langle 0, \mathtt{a}, 2\rangle\, \langle 1, \mathtt{c}, 1\rangle, \langle 3, \mathtt{a}, 2\rangle, \langle 1, \mathtt{a}, 1\rangle, \langle 0, \varepsilon, 1\rangle,$
   $\langle 0, \mathtt{b}, 1\rangle, \langle 1, \mathtt{c}, 0\rangle, \langle 0, \mathtt{c}, 3\rangle, \langle 2, \varepsilon, 1\rangle, \langle 3, \mathtt{b}, 1\rangle\}$
4. $q_0 = 1$
5. $F = \{3\}$

6 Transforming regular expressions into FSMs (4 pts)

Determine a minimized DFA representing the same language as the regular expression

$$r = (\mathtt{1}^*\mathtt{01})^*.$$

7 Formal grammars (8 pts)

a) Define a formal grammar representing the language L specified in Task 8.

b) Consider the grammar

$$G = \langle V_N, V_T, P, S\rangle$$

with

1. $V_N = \{A, B\}$,
2. $V_T = \{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$,

3. $P:$

$$\begin{aligned} A &\rightarrow \mathtt{0} & 1 \\ A &\rightarrow ABA & 2 \\ BA &\rightarrow \mathtt{1} & 3 \\ AB &\rightarrow \mathtt{2} & 4 \end{aligned}$$

4. $S = A$.

i. What is G's highest type? Justify your answer.

ii. Formally describe the language $L(G)$.

iii. If G is not itself regular, define a regular grammar G' with

$$L(G') = L(G).$$

8 The pumping lemma (4 pts)

Prove whether or not the following language is regular:

$$L = \{\mathtt{a}^k\mathtt{b}^l | k, l \in \mathbb{I} \wedge l \leq k\}.$$