# Examination
# Formal Languages and Automata Theory
# TDDD14/TDDD85

(Formella Språk och Automatateori)

2024–05–30

1. You may answer in Swedish or English.
2. Allowed help materials
   - A sheet of handwritten notes - 2 sided A5 or 1 sided A4.
     The contents is up to you. Return the notes together with the exam. The notes should be signed in the same way as the exam sheets and returned together with the exam.
   - English dictionary
3. The maximum number of points is 28. The grades are as follows:

| Grade | TDDD14 | TDDD85 |
|---|---|---|
| 3 | 14–19 | 12–17 |
| 4 | 20–23 | 18–22 |
| 5 | 24–28 | 23–28 |

Make sure that you justify your answers! Unexplained answers will be granted 0 points. (For instance, if you are writing a grammar for a given language then you should also explain that the grammar indeed generates the language. If you apply some known method then you should explain each step. And so on.)

GOOD LUCK !

1. *(4p)* Which of the following claims are true?

   (a) The subset construction for converting an NFA to a DFA always produces a minimal DFA.

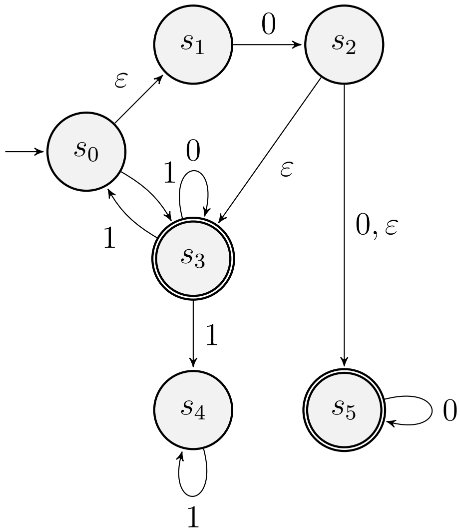
   (b) If $R$ is a regular expression which contains at least one occurrence of the star operator then $L(R)$ is infinite (recall that $L(R)$ is the set of strings generated/matched by the regular expression $R$).

   (c) LR(0) grammars have exactly the same expressive power as deterministic pushdown automata (DPDA).

   (d) There exists languages which are undecidable but still Turing-recognizable.

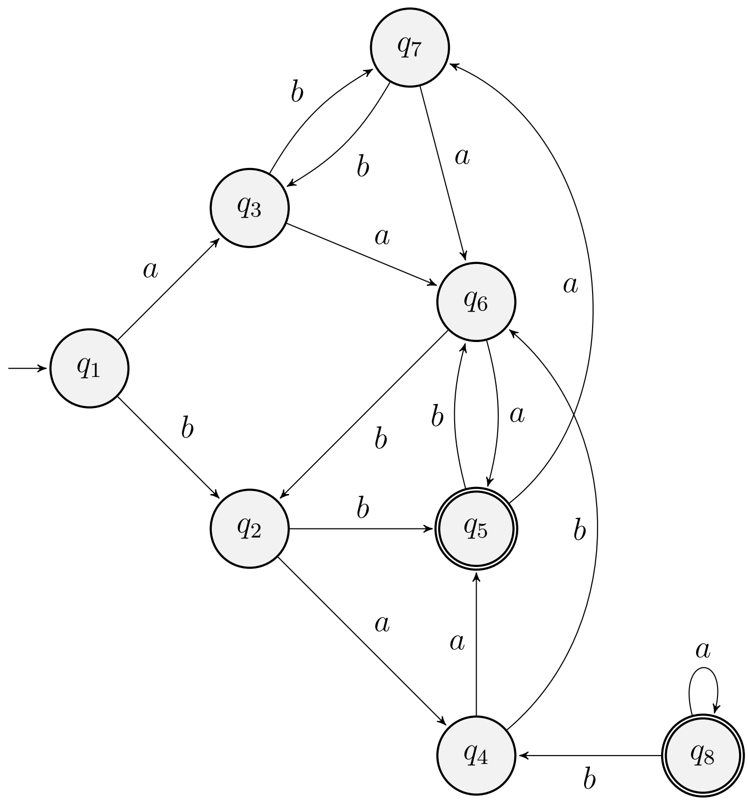
   **Answer each claim by first stating whether the claim is true or false, and then motivate your claim. A substantial motivation is necessary in order to get points.**

2. *(4p)* Consider the following NFA.




   Using the subset construction method, construct an equivalent DFA. **Explain each step of the construction. A table/figure without any explanation will be given 0 points.**

3. *(4p)* Consider the following DFA.




Construct the minimal DFA equivalent to this DFA by using the marking algorithm (or a similar algorithm).

**Include all relevant calculations. In particular, if you in an iteration detect that two states are not equivalent, then you have to include the corresponding calculation.**

4. *(6p)* Consider the language $L$ generated by the grammar

$$S \to aAc \qquad\qquad A \to aAc \mid aaAc \mid b$$

($S$ is the start symbol and $a, b, c$ are terminal symbols.)

(a) Prove that $L$ is not regular by using the pumping lemma for regular languages.

(b) Prove that $L$ is not regular by using the Myhill-Nerode theorem.

**Hint: begin by describing the strings in $L$. What is the relationship between the number of $a$'s and $c$'s?**

5. *(2p)* In the Lisp family of programming languages, the syntax is based on nested parenthesized expressions with prefix notation. For example,

`(f (g a))` applies the function $f$ where the operand is the result of the function $g$ applied to $a$. We can specify a context-free grammar for a simplified Lisp-like language. The alphabet of this language is $\Sigma = \{f, a, b, (, ), \text{␣}\}$, where ␣ represents whitespace.

$$E \rightarrow \text{'('} \quad I \quad \text{'␣'} \quad E \quad \text{')'} \quad | \quad I \qquad (1)$$

$$I \rightarrow f \quad | \quad a \quad | \quad b \qquad (2)$$

Construct a (N)PDA from the CFG above. Describe the set of states $Q$ and the transition relation $\delta$ as well as the acceptance condition.

*Hint: The process from the course literature first converts the grammar to Greibach Normal Form. A good first step may be to introduce new nonterminal productions for some of the terminals appearing on the right-hand-side of production 1 above.*

6. *(4p)* For a Boolean string $w \in \{0, 1\}^*$, let $\overline{w}$ denote the complement. For example, $\overline{010} = 101$. Consider the language $L = \{\, w\overline{w} \mid w \in \{0, 1\}^*\}$. Is $L$ context-free? Either prove your claim by giving a context-free grammar for the language or prove that it is not context-free by using the pumping lemma for context-free languages.

7. *(4p)* Let $A$ and $B$ be two context-free languages. Under what conditions is it true that $A \leq_m B$ (i.e., that $A$ is mapping reducible to $B$)?