ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Prof. Dr. Andreas Podelski
Matthias Heizmann
Christian Schilling

4. Exercise Sheet for the Tutorial

Computer Science Theory

Announcement: Starting with this exercise sheet, we change the course interval from one-week to two-week. Accordingly, exercise sheets are designed for two weeks of work.

Exercise 1: Limits of the Pumping Lemma

Consider the following language over the alphabet $\Sigma = \{a, b, c\}$:

$$L = \{a^i b^j c^k \mid i = 0 \text{ or } k < j, \text{ for } i, j, k \in \mathbb{N}\}$$

Apply the pumping lemma. Does it work? What does this mean?

Exercise 2: $\equiv_{\mathcal{A}}$ Equivalence

Let $\mathcal{A} = (\Sigma, Q, \rightarrow, q_0, F)$ be a nondeterministic finite automaton. For words $u, v \in \Sigma^*$ we define the relation $\equiv_{\mathcal{A}}$ as

$$u \equiv_{\mathcal{A}} v \quad \text{iff} \quad \text{there is } q \in Q \text{ such that } q_0 \xrightarrow{u} q \text{ and } q_0 \xrightarrow{v} q.$$

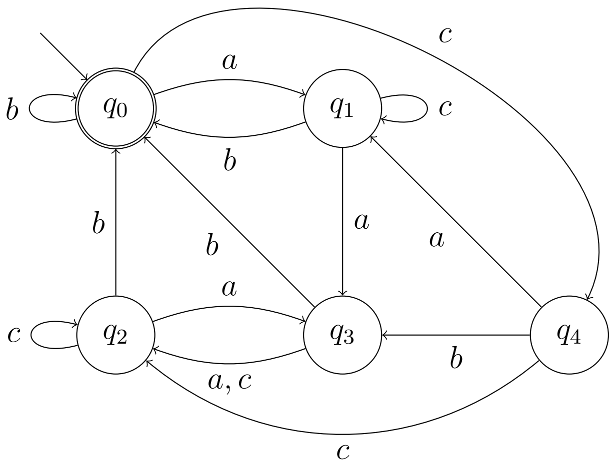
Show that $\equiv_{\mathcal{A}}$ is *not* an equivalence relation for *nondeterministic* finite automata.

Hint: You should give a counterexample: an NFA $\mathcal{A}$ and two words $u, v \in \Sigma^*$ such that at least one of the properties of an *equivalence relation* is not satisfied for $\equiv_{\mathcal{A}}$.

Exercise 3: Minimal Automaton
Consider the following algorithm:

> ## MINIMIZATION ALGORITHM
>
> **Input:** DFA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$.
>
> **Output:** Minimal DFA recognizing $L(\mathcal{A})$.
>
> 1. Eliminate all unreachable states.
> 2. Maintain a table such that there is a cell for every set of states $\{q, q'\}$ with $q \neq q'$ (ignoring the order).
> 3. Mark every set of states $\{q, q'\}$ with $q \in F$ and $q' \notin F$.
> 4. For every unmarked set of states $\{q, q'\}$ and every symbol $a \in \Sigma$, consider the set of states $\{\delta(q, a), \delta(q', a)\}$. If $\{\delta(q, a), \delta(q', a)\}$ is marked, then also mark $\{q, q'\}$.
> 5. Repeat step 4 until there are no more changes in the table.
> 6. Merge all states for which the corresponding set is not marked.

Apply this algorithm to the following DFA.



Provide the minimal automaton and the final marking table.

Exercise 4: Context-free Grammars
Consider the context-free grammar $G = (N, T, P, S)$ with $N = \{S\}$, $T = \{a, b\}$ and

$$P = \{S \to \varepsilon,\quad S \to aSbS,\quad S \to bSaS\}.$$

(a) Provide a derivation for the word *abbbaa*.

(b) Which language L is generated by G? Provide a simple formulation of L and describe why $L(G) \subseteq L$ holds (we ignore $L \subseteq L(G)$).

(c) Is G unambiguous? Justify your answer.

Exercise 5: Logical Formulae as Context-free Grammar
Consider the finite set $X = \{x_1, \dots, x_n\}$ of variables. In propositional logic, a *formula* is defined inductively as follows:

- Every variable $x \in X$ is a formula.
- If ϕ is a formula, then $\neg\phi$ is a formula.
- If ϕ, ψ are formulae, then $(\phi \wedge \psi)$ and $(\phi \vee \psi)$ are formulae.

(a) Provide a context-free grammar that generates the language of all formulae with variables from X. Use the following terminal symbols:

$$T = X \ \cup \ \{\neg,), (, \wedge, \vee\}$$

(b) Exemplarily give two words $w_1, w_2 \in T^*$ generated by your grammar (which should be formulae) and two words $w_3, w_4 \in T^*$ not generated by your grammar (which should not be formulae).