Theoretical Computer Science 1
Exercise Sheet 4

René Maseli
Thomas Haas
TU Braunschweig
Winter Semester 2023/24

Release: 2023-12-08 Due: 2023-12-21 23:59

Hand in your solutions to the Vips directory of the StudIP course until Thursday, December 21th 2023 23:59. You should provide your solutions either directly as .pdf file or as a readable scan/photo of your handwritten notes. Submit your results as a group of four and state **all** members of your group with **student id, name and course**.

Definition: Finite-state Transducer

A finite-state transducer over a finite input alphabet Σ and a finite output alphabet Γ is formally a quadruple $T = \langle Q, q_0, \rightarrow, Q_F \rangle$ consisting of

1. a finite set of states Q,
2. an initial state $q_0 \in Q$,
3. a transition relation $\rightarrow \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \times Q$,
4. and a set of accepting states $Q_F \subseteq Q$.

In the following we fix notation and important definitions:

1. $\langle p, a, x, q \rangle \in \rightarrow$ is denoted by $p \xrightarrow{a/x} q$. When reading an a in state p, the transducer transitions to state q and outputs x. Intuitively, $a = \varepsilon$ denotes a spontaneous transition, while $x = \varepsilon$ denotes a transition without output.
2. A computation $q_0 \xrightarrow{w_1/o_1} q_1 \xrightarrow{w_2/o_2} \cdots \xrightarrow{w_{n-1}/o_{n-1}} q_{n-1} \xrightarrow{w_n/o_n} q_n$ can also be denoted by $q_0 \xrightarrow{w/o} q_n$, where $w \in \Sigma^*$ and $o \in \Gamma^*$ are the respective concatenations without ε.
3. We define for any language $L \subseteq \Sigma^*$ the translation under T as
   $T(L) = \{\, o \in \Gamma^* \mid \exists w \in L, q_f \in Q_F : q_0 \xrightarrow{w/o} q_f \,\}$.

Homework Exercise 1: Finite-state transducers [5 points]

A transducer can be thought of as an NFA with spontaneous transitions, which not only accepts input words but also outputs new words. It translates input words from Σ^* to output words in Γ^*. Transducers are used in linguistics and the processing of natural languages.

a) [3 points] Construct a transducer T that for any given word $w \in \{a, b, c\}^*$ works a follows: it prepends a b to every occurence of a and removes every second occurrence of c. Give a regular expression for $T((ac)^*)$. A proof of correctness is not needed.

b) [2 points] We call a transducer deterministic if in any state and for any input, the transducer has exactly one possible, and hence unique, transition; this transition may be spontaneous. For example, a state with an a-labeled transition may not have another a-labeled transition nor another spontaneous transition, because in either case there would be two possible transitions on a.

Show that it is **not** possible to determinize transducers in general. That means, there are transducers T which do not have any equivalent deterministic transducer T^{det} such that $T(L) = T^{det}(L)$ for all languages $L \subseteq \Sigma^*$.

Homework Exercise 2: Closedness under transducers [12 points]

Prove that any class of languages is closed under translations of transducers, if and only if it is closed under intersection with regular languages, images and co-images of homomorphisms.

a) [3 points] Let $h : \Sigma^* \to \Gamma^*$ be an arbitrary homomorphism between words. Construct a transducer T_h such that $T_h(L) = h(L)$ holds for all languages $L \subseteq \Sigma^*$. Prove the correctness of your construction.

b) [3 points] Now prove that there is also a transducer $T_{h^{-1}}$ such that $T_{h^{-1}}(L) = h^{-1}(L)$ holds for all $L \subseteq \Gamma^*$. Prove the correctness of your construction.

c) [2 points] Show that for any regular language M, there is a transducer T_M with $T_M(L) = L \cap M$.

d) [4 points] Now show that the translation under any transducer T can be expressed as a combination of the three operations mentioned above.

Homework Exercise 3: Equivalence classes [9 points]

Let $\Sigma = \{a, b\}$ be an alphabet.

a) [5 points] Consider $L = \{ a^n b a^m \mid n, m \in \mathbb{N}, n \geq m \}$. Prove that

$$[a^n]_{\equiv_L} = \{a^n\} \text{ for all } n \in \mathbb{N}$$
$$[a^n.a.b]_{\equiv_L} = \left\{ a^{\ell+1}.b^{\ell+1-n} \;\middle|\; \ell \in \mathbb{N}, \ell \geq n \right\} \text{ for all } n \in \mathbb{N}$$

holds. With infinite congruence classes, L is not regular by the Theorem of Myhill & Nerode. Find all remaining equivalence classes with respect to $\equiv_L$.

b) [2 points] Consider the language $M = \{a, b\}^*.\{aab, abb\}.\{a, b\}^*$. Find all equivalence classes of $\equiv_M$. Construct the equivalence class automaton A_M.

c) [2 points] Consider the language $N = \{a, b\}^*.\{a\}.\{a, b\}^* \cup (\{a, b\}.\{a, b\})^*$. Find all equivalence classes of $\equiv_N$. Construct the equivalence class automaton A_N.

Homework Exercise 4: Theorem of Myhill & Nerode [10 points]

Let $L \subseteq \Sigma^*$ be a regular language with $\text{Index}(\equiv_L) = k \in \mathbb{N}$ and let $A\langle Q, q_0, \to, Q_F\rangle$ be a DFA with $L = \mathcal{L}(A)$ and $|Q| = k$. Let further $A_L = \langle Q_L, q_{0L}, \to_L, Q_{FL}\rangle$ be the equivalence automaton for L with $\mathcal{L}(A_L) = L$ and $u_1, \ldots, u_k$ be the representants of the equivalence classes of $\equiv_L$.

Show Theorem 6.11 from the script: A and A_L are isomorphic. The isomorphism $\beta : Q_L \to Q$ is defined as: $\beta([u_i]_{\equiv_L}) = q$ with $q_0 \xrightarrow{u_i} q$ in A.

a) [2 points] Consider the equivalence relation $\equiv_A$. Show that $\text{Index}(\equiv_A) = \text{Index}(\equiv_L)$ holds. With the result $\equiv_A \subseteq \equiv_L$ from the lecture, this implies $\equiv_A = \equiv_L$.

b) [3 points] Show that β is well-defined.
Hint: The function β was defined on equivalence classes. You have to show, that β is independent of the choice of the representant $u_1, \dots, u_k$. Let us assume $\hat{u}_i \equiv_L u_i$ and show that $\beta([\hat{u}_i]_{\equiv_L}) = \beta([u_i]_{\equiv_L})$ holds.

c) [2 points] Show that β is a bijection between Q_L and Q.

d) [3 points] Show that β is isomorphic. It remains to show, that $\beta(q_{0L}) = q_0$, $\beta(Q_{FL}) = Q_F$ and for all $p, p' \in Q_L$ and $a \in \Sigma$ the property $p \xrightarrow{a}_L p'$ iff $\beta(p) \xrightarrow{a} \beta(p')$ holds.

Exercise 5:

In this exercise we want to show that some languages that admit a description by small NFAs do not admit a description by small DFAs; every DFA for that language is necessarily large.

For a number $k \in \mathbb{N}, k > 0$ let $L_{a@k} = \Sigma^*.a.\Sigma^{k-1}$ be the language of words over $\Sigma = \{a, b\}$ that have an a at the k-th last position.

a) Show how to construct for any $k \in \mathbb{N}, k > 0$ an NFA $A_k = \langle Q_k, q_0, \to_k, F_k \rangle$ with $\mathcal{L}(A_k) = L_{a@k}$ and $|Q_k| = k + 1$. Give the automaton formally as a tuple.
You do not have to show correctness of your construction.

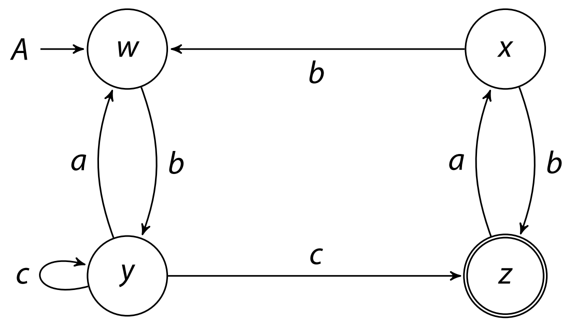
b) Now draw A_3 and its determinization A_3^{det} via Rabin-Scott-power set construction.
Compare the number of states of A_3 and A_3^{det}.

c) Let $k \in \mathbb{N}, k > 0$ be arbitrary. Prove that for $L_{a@k}$ there is no DFA B with less than 2^k many states such that $\mathcal{L}(B) = L_{a@k}$ holds.
Proceed as follows:

1. Assume there is a finite automaton $B = \langle Q', q'_0, \to', Q'_F \rangle$ with $\mathcal{L}(B) = L_{a@k}$ and $|Q'| < 2^k$.
2. Consider the set Σ^k of words of length k. How many such words are there?
3. Now consider to each word $w \in \Sigma^k$ the (unique) state q_w in the DFA B after it read the word w.
4. Now derive a contradiction.

Exercise 6:

Examine the following NFA A over the alphabet $\Sigma = \{a, b, c\}$:



Consider the homomorphism $f\colon \Sigma \to \{0, 1\}$. $g(a) = \varepsilon \quad g(b) = 10 \quad g(c) = 01$

a) Construct the image-automaton $g(A)$ with $\mathcal{L}(g(A)) = g(\mathcal{L}(A))$. Show $1001011010100110 \in \mathcal{L}(g(A))$ by giving a corresponding run through A.

Consider the homomorphism $g\colon \{d, e\} \to \Sigma$ with $g(d) = bcc \quad g(e) = ab$.

b) Construct the co-image-automaton $g^{-1}(A)$ with $\mathcal{L}(g^{-1}(A)) = g^{-1}(\mathcal{L}(A))$.

c) Show $deeded \in \mathcal{L}(g^{-1}(A))$ by giving a corresponding run through A.

d) Consider the following homomorphism $g\colon \{c, d, e\} \to \Sigma$.

$$g(c) = \varepsilon$$
$$g(d) = bbb$$
$$g(e) = ba$$

Konstruieren Sie den Urbild-Automaten $g^{-1}(A)$ mit $\mathcal{L}(g^{-1}(A)) = g^{-1}(\mathcal{L}(A))$.
Show $ceddc \in \mathcal{L}(g^{-1}(A))$ by giving a corresponding run through A.

Exercise 7:

Let $\equiv\, \subseteq \Sigma^* \times \Sigma^*$ be an equivalence relation on words. As usual, we write $u \equiv v$ (instead of $\langle u, v\rangle \in\, \equiv$) to express that u and v are equivalent with respect to $\equiv$.

a) Prove formally the following basic properties about equivalence relations:

- Every word is contained in its own equivalence class: $u \in [u]_\equiv$.
- The equivalence classes of equivalent words are equal: $u \equiv v \implies [u]_\equiv = [v]_\equiv$.
- The equivalence classes of non-equivalent words are disjoint: $u \not\equiv v \implies [u]_\equiv \cap [v]_\equiv = \varnothing$.

b) Let $L \subseteq \Sigma^*$ and $\equiv_L$ be the Nerode right-congruence, known from the lecture, with

$$u \equiv_L v \quad \text{gdw.} \quad \forall w \in \Sigma^*\colon u.w \in L \text{ iff } v.w \in L\,.$$

Prove that $\equiv_L$ is indeed an equivalence relation and a right-congruence. The latter means, that for all u, v with $u \equiv_L v$ and all $x \in \Sigma^*$ it holds that: $u.x \equiv_L v.x$.

c) Let $A = (Q, q_0, \rightarrow, Q_F)$ be a DFA. The relation $\equiv_A \subseteq \Sigma^* \times \Sigma^*$ is defined by:

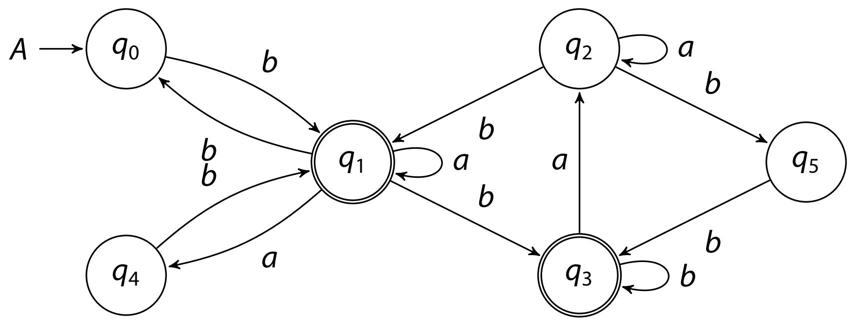
$$u \equiv_A v \quad \text{gdw.} \quad \exists q \in Q\colon q_0 \xrightarrow{u} q \text{ und } q_0 \xrightarrow{v} q.$$

Show that $\equiv_A$ is an equivalence relation.

d) Is $\equiv_A$ from c) still an equivalence relation if A is a an NFA instead? Explain your answer!

Exercise 8:

Minimalise the following automaton A.



a) Construct the powerset automaton $\mathcal{P}(A)$.

b) Perform the *Table-Filling Algorithm* on $\mathcal{P}(A)$. Annote each marked cell of the table with the step, when that pair of states was marked. (starting with 0 for final/nonfinal states.)

c) Draw the minimal DFA for $\mathcal{L}(A)$.