Theoretical Computer Science
Exercise Sheet 5

Prof. Dr. Roland Meyer
René Maseli
TU Braunschweig
Winter Semester 2025/26

Release: 2025-12-20 Due: 2025-01-15 23:59

Homework Exercise 5.1: Closure properties of regular languages [3 points]

Let Σ an alphabet and $X, L, R \subseteq \Sigma^*$ languages. The remainder $L \lhd X \rhd R$ is defined as

$$L \lhd X \rhd R := \left\{ v \in \Sigma^* \mid \exists u \in L, w \in R : u.v.w \in X \right\}.$$

Let X, L and R regular languages. Show that $L \lhd X \rhd R$ is regular.

Homework Exercise 5.2: Pumping lemma for regular languages [4 points]

Consider $\Sigma = \{a, b, c\}$ For all words v and w let $|w|_v$ be the number of occurrences of v in w.
By using the Pumping Lemma, prove that the following languages are not regular.

a) [1 point] $L_\mathrm{a} = \left\{ xb^my \in \{a, b\}^* \mid \exists n \in \mathbb{N} : |y| = n \text{ and } x \in (a^*b)^n \text{ and } m \geq 2 \right\}$

b) [1 point] $L_\mathrm{b} = \{ u.v \in \{a, b\}^* \mid |u|_a + u_b = |v|_a + v_c \}$

c) [1 point] $L_\mathrm{c} = \{ a^{m \cdot n}b^n \mid m, n \in \mathbb{N} \}$

d) [1 point] $L_\mathrm{d} = \{ w \in \{a, b, c\}^* \mid |w|_{ab} < |w|_{ac} \}$

Homework Exercise 5.3: Replacement systems [5 points]

Consider $\Sigma = \{a, b\}$. Give context free grammars G_1, G_2, G_3 and G_4, which produce the following languages:

a) [1 point] $L_\mathrm{a} := \{ a^nb^mw \mid w \in \Sigma^* \text{ and } m > 2 \text{ and } |w|_a = n \}$.

b) [1 point] $L_\mathrm{b} := \{ w \in \Sigma^* \mid |w|_a < |w|_b \}$.

c) [1 point] $L_\mathrm{c} := \{ a^mb^n \mid m, n \in \mathbb{N}, m \neq n \}$.

d) [1 point] $L_\mathrm{d} := \left\{ w_1 \dots w_n \in \Sigma^n \mid n \in \mathbb{N}, \forall 1 \leq k \leq n\colon |w_k \dots w_n|_a \leq |w_k \dots w_n|_b \right\}$.

e) [1 point] $L_\mathrm{e} = \bigcup_{m \in \mathbb{N}} M^m.c^m$, where $M = \{ a^nb^n \mid n \in \mathbb{N} \}$ is already known to be context-free (see Example 8.18 from the lecture notes). E.g. $bbabaacc \in L_\mathrm{e}$.

Remark: Die Sprache der wohlgeformten einfachen ‚Klammerterme' aus ab ist $L_\mathrm{d} \cap \{ w \in \Sigma^* \mid |w|_a = |w|_b \}$.

Homework Exercise 5.4: Programm Languages [3 points]

A control path of a while-program can be seen as a walk in the controlflow graph, i.e. a block sequence, such that all consecutive pairs are connected with a flow edge. In program analysis, those walks are usually called paths (acyclic walks), because e.g. their index in the sequence are also considered.

Consider the following programs P_a, P_b and P_c with blocks $B = \{0,1,2,3,4,5,6,7\}$.

P_a:

```
[x := 0]^0
while [x^2 < y]^1 do
    [z := x + 1]^2
    [x := z^2 + z]^3
end while
if [x^2 = y]^4 then
    [z := 1]^5
else
    [z := y]^6
end if
[x := z]^7
```

P_b:

```
[x := 0]^0
while [x < 2^4]^1 do
    [y := 3x + 2]^2
    while [y < 5x]^3 do
        [y := y + 2]^4
        if [3x < y]^5 then
            [x := x + 1]^6
        end if
    end while
end while
[x := x − 14]^7
```

```
procedure P_c(x, y) do
    [z := 1]^0
    while [z · (z + 1) < y]^1 do
        [z := z + 1]^2
    end while
    if [z = y]^3 then
        [z := 1]^4
    else
        [P_c(z, y)]^5
    end if
end procedure
```

a) [1 point] Construct a **right-linear** grammar G_a over the alphabet $\Sigma = B$, that produces exactly all control paths of P_a starting in block 0 and ending in block 7. E.g. $01457 \in \mathcal{L}(G_\mathrm{a})$ and $01231467 \in \mathcal{L}(G_\mathrm{a})$, are valid, but e.g. $01234567 \notin \mathcal{L}(G_\mathrm{a})$ must not be producable.

b) [1 point] Construct a **left-linear** grammar G_b over the alphabet $\Sigma = B$, that produces exactly all control paths of P_b starting in block 0 and ending in block 7. E.g. $017 \in \mathcal{L}(G_\mathrm{b})$ is the shortest word of the language. Another element is e.g. $0123456317 \in \mathcal{L}(G_\mathrm{b})$, but e.g. $01234567 \notin \mathcal{L}(G_\mathrm{b})$ must not be produceable.

c) [1 point] Construct a **linear** grammar G_c for the control paths of P_c, including the blocks inside the recursion. Here, block 5 shall be positioned directly **after** the recursive call. E.g. $030345 \in \mathcal{L}(G_\mathrm{c})$.

Homework Exercise 5.5: Regular grammars [4 points]

Prove that the regular languages exactly coincide with the languages that are produced by some right-linear grammar.

a) [2 points] Explain how to construct a right linear grammar G from a given NFA A such that $\mathcal{L}(G) = \mathcal{L}(A)$ holds.

b) [2 points] Explain how to construct an NFA A from a given right linear grammar G such that $\mathcal{L}(G) = \mathcal{L}(A)$ holds.

Tutorial Exercise 5.6:

Consider abstract programs on boolean variables. An assignment $[x := *]^\ell$ non-deterministically guesses the value for x, as if read from an input stream.

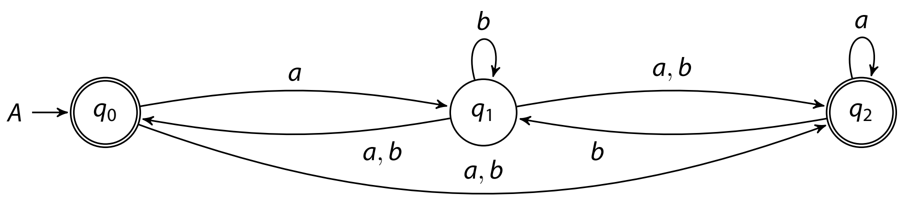
Consider the following program P with blocks $B = \{0, 1, 2, 3, 4, 5, 6, 7\}$ and variables $\text{Vars} = \{x, y, z\}$.

$[x := *]^0$
while [not x or not $z]^1$ **do**
  $[y :=$ not x and not $z]^2$
  $[x := *]^3$
  if $[y]^4$ **then**
    $[x :=$ not $x]^5$
  end if
  $[z := x]^6$
end while
$[\text{skip}]^7$

Construct a finite automaton A_P that accepts the assigned values as a language over $\Sigma = \text{Vars} \times \{0, 1\}$.
$\varepsilon \notin \mathcal{L}(A_P)$, since every execution must pass through Block b_0.
$x1 \notin \mathcal{L}(A_P)$, since executions always start with $z = 0$ and therfore have to iterate at least once.
$x1.y0.x1.z1 \in \mathcal{L}(A_P)$, because there is an execution that first reads 1 and then 0, breaking the loop.

Tutorial Exercise 5.7:

Consider the following NFA A over $\{a, b\}$.



a) From A, construct a language equivalent DFA $\mathcal{P}(A)$ using the Rabin-Scott power set construction. Make sure that $\mathcal{P}(A)$ has no unreachable states.

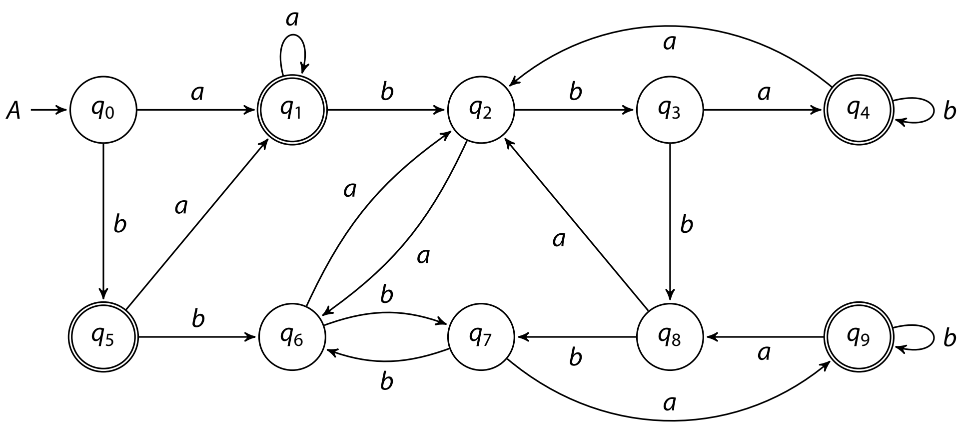
b) Determine the ~-equivalence classes on the states of $\mathcal{P}(A)$ by using the Table-Filling-Algorithm from the lecture. Make clear in which order the cells of the table were marked.

c) Give the minimal DFA B for $\mathcal{L}(A)$. Make use of the ~-equivalence classes.

d) Find all equivalence classes of the Nerode right-congruence $\equiv_{\mathcal{L}(A)}$.

Tutorial Exercise 5.8:

Consider the following NFA A over $\{a, b\}$.



a) Determine the ~-equivalence classes on the states of A by using the Table-Filling-Algorithm from the lecture. Make clear in which order the cells of the table were marked.

b) Give the minimal DFA B for $\mathcal{L}(A)$. Make use of the ~-equivalence classes.

c) Find all equivalence classes of the Nerode right-congruence $\equiv_{\mathcal{L}(A)}$. Find an expression for $\mathcal{L}(A)$ as a union of a certain subset of those classes.

d) Use $\overline{B}$ to construct a regular expression for $\overline{\mathcal{L}(A)}$.