Theoretische Informatik 1
Übungsblatt 5

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

Ausgabe: 2025-12-20 Abgabe: 2025-01-15 23:59

Hausaufgabe 5.1: Abschlusseigenschaften regulärer Sprachen [3 Punkte]
Seien Σ ein Alphabet und $X, L, R \subseteq \Sigma^*$ Sprachen. Der Rest $L \lhd X \rhd R$ sei definiert als

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

Seien X, L und R reguläre Sprachen. Zeigen Sie, dass $L \lhd X \rhd R$ regulär ist.

Hausaufgabe 5.2: Pumping-Lemma für reguläre Sprachen [4 Punkte]
Sei $\Sigma = \{a, b, c\}$ ein Alphabet.Für alle Wörter v und w beschreibe $|w|_v$ die Anzahl der Vorkommen von v in w.
Beweisen Sie unter Verwendung des Pumping-Lemmas, dass die folgenden Sprachen nicht regulär sind.

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

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

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

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

Hausaufgabe 5.3: Ersetzungssysteme [5 Punkte]
Sei $\Sigma = \{a, b\}$ ein Alphabet. Geben Sie kontextfreie Grammatiken G_1, G_2, G_3 und G_4 an, welche die folgenden Sprachen ableiten:

a) [1 Punkt] $L_a := \{ a^n b^m w \mid w \in \Sigma^* \text{ und } m > 2 \text{ und } |w|_a = n \}$.

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

c) [1 Punkt] $L_c := \{ a^m b^n \mid m, n \in \mathbb{N}, m \neq n \}$.

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

e) [1 Punkt] $L_e = \bigcup_{m \in \mathbb{N}} M^m.c^m$, wobei $M = \{ a^n b^n \mid n \in \mathbb{N} \}$ bekanntermaßen kontextfrei ist (siehe Bsp. 8.18 im Skript). Z.B. $bbabaacc \in L_e$.

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

Hausaufgabe 5.4: Programmsprachen [3 Punkte]

Ein Kontrollpfad eines While-Programms kann als ein Weg im Kontrollfluss-Graphen verstanden werden, d.h. eine Block-Folge, in der alle aufeinanderfolgenden Paare durch eine Fluss-Kante verbunden sind. In der Programmanalyse bezeichnet man sie üblicherweise als Pfade (Wege ohne Kreise), weil man z.B. auch deren Index in der Folge beachtet.

Betrachten Sie die folgenden Programme P_a, P_b und P_c jeweils mit Blöcken $B = \{0,1,2,3,4,5,6,7\}$.

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
```

P_c:

```
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 Punkt] Geben Sie eine **rechtslineare** Grammatik G_a über dem Alphabet $\Sigma = B$ an, welche genau die Kontrollpfade von P_a produziert, die in Block 0 beginnen und in Block 7 enden. Z.B. ist $01457 \in \mathcal{L}(G_a)$ und $01231467 \in \mathcal{L}(G_a)$, sind gültig, aber z.B. $01234567 \notin \mathcal{L}(G_a)$ darf nicht akzeptiert werden.

b) [1 Punkt] Konstruieren Sie eine **linkslineare** Grammatik G_b über dem Alphabet $\Sigma = B$, welche genau die Kontrollpfade von P_b produziert, die in Block 0 beginnen und in Block 7 enden. Z.B. ist $017 \in \mathcal{L}(G_b)$ das kleinste Wort der Sprache. Ein weiteres Element ist z.B. $0123456317 \in \mathcal{L}(G_b)$, aber z.B. $01234567 \notin \mathcal{L}(G_b)$ darf nicht ableitbar sein.

c) [1 Punkt] Konstruieren Sie eine **lineare** Grammatik G_c für die Kontrollpfade von P_c, einschließlich der Blöcke innerhalb der Rekursion. Dabei soll sich Block 5 unmittelbar **nach** dem rekursiven Aufruf befinden. Z.B. $030345 \in \mathcal{L}(G_c)$.

Hausaufgabe 5.5: Reguläre Grammatiken [4 Punkte]

Beweisen Sie, dass die regulären Sprachen genau die Sprachen sind, die von einer rechtslinearen Grammatik erzeugt werden.

a) [2 Punkte] Erklären Sie, wie man zu einem gegebenen NFA A eine rechtslineare Grammatik G mit $\mathcal{L}(G) = \mathcal{L}(A)$ konstruieren kann.

b) [2 Punkte] Erklären Sie, wie man zu einer gegebenen rechtslineare Grammatik G einen NFA A mit $\mathcal{L}(G) = \mathcal{L}(A)$ konstruieren kann.

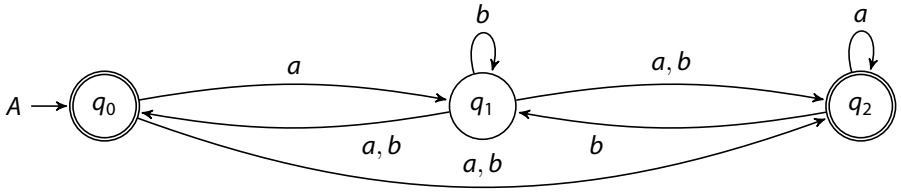
Übungsaufgabe 5.6:
Wir betrachten abstrakte Programme über boolschen Variablen. Eine Zuweisung $[x := *]^\ell$ rät den Wert für x nicht-deterministisch, so als ob es von einer Eingabe gelesen wird.

Betrachten Sie das folgende boolsche Programm P mit den Blöcken $B = \{0, 1, 2, 3, 4, 5, 6, 7\}$ und den Variablen Vars $= \{x, y, z\}$.

$[x := *]^0$
while $[$ nicht x oder nicht $z]^1$ **do**
  $[y :=$ nicht x und nicht $z]^2$
  $[x := *]^3$
  if $[y]^4$ **then**
    $[x :=$ nicht $x]^5$
  end if
  $[z := x]^6$
end while
$[\text{skip}]^7$

Konstruieren Sie einen endlichen Automaten A_P, welcher die zugewiesenen Werte als eine Sprache über $\Sigma = \text{Vars} \times \{0, 1\}$ auffasst.
$\varepsilon \notin \mathcal{L}(A_P)$, da jede Ausführung mindestens durch Block b_0 führen muss.
$x1 \notin \mathcal{L}(A_P)$, da jede Ausführung mit $z = 0$ startet und deswegen mindestens einmal die Schleife durchlaufen muss.
$x1.y0.x1.z1 \in \mathcal{L}(A_P)$, da die Ausführungen zuerst 1 und dann 0 lesen können, wodurch die Austrittsbedingung erfüllt wird.

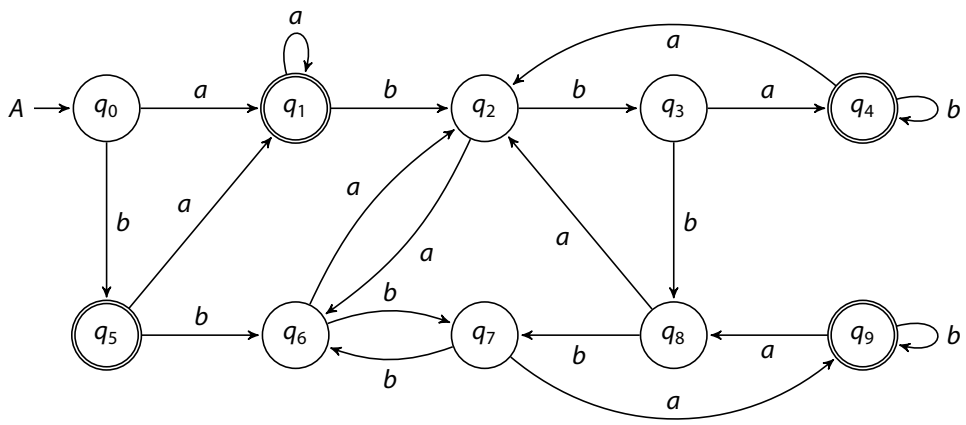
Übungsaufgabe 5.7:
Betrachten Sie den folgenden NFA A über $\{a, b\}$.



a) Konstruieren Sie einen zu A sprachäquivalenten DFA $\mathcal{P}(A)$ unter Verwendung der Rabin-Scott-Potenzmengenkonstruktion. Stellen Sie sicher, dass $\mathcal{P}(A)$ keine unerreichbaren Zustände enthält.
b) Bestimmen Sie alle ~-Äquivalenzklassen auf den Zuständen von $\mathcal{P}(A)$ unter Verwendung des Table-Fillings-Algorithmus aus der Vorlesung. Geben Sie an, in welcher Reihenfolge Sie die Zellen in der Tabelle markiert haben.
c) Geben Sie den minimalen DFA B für $\mathcal{L}(A)$ an. Verwenden Sie hierzu die ~-Äquivalenzklassen.
d) Bestimmen Sie alle Äquivalenzklassen der Nerode-Rechtskongruenz $\equiv_{\mathcal{L}(A)}$.

Übungsaufgabe 5.8:

Betrachten Sie den folgenden NFA A über $\{a, b\}$.



a) Bestimmen Sie auf den Zuständen von A alle ~-Äquivalenzklassen unter Verwendung des Table-Fillings-Algorithmus aus der Vorlesung. Geben Sie an, in welcher Reihenfolge Sie die Zellen in der Tabelle markiert haben.

b) Geben Sie den minimalen DFA B für $\mathcal{L}(A)$ an. Verwenden Sie hierzu die ~-Äquivalenzklassen.

c) Bestimmen Sie alle Äquivalenzklassen der Nerode-Rechtskongruenz $\equiv_{\mathcal{L}(A)}$. Stellen Sie $\mathcal{L}(A)$ als Vereinigung einer bestimmten Teilmenge dieser Äquivalenzklassen dar.

d) Nutzen Sie $\overline{B}$, um einen regulären Ausdruck für $\overline{\mathcal{L}(A)}$ zu finden.