René Maseli
Prof. Dr. Roland Meyer

Theoretische Informatik 1
Übungsblatt 3

TU Braunschweig
Wintersemester 2022/23

Ausgabe: 29.11.2022 Abgabe: 09.12.2022, 09:45

Geben Sie Ihre Lösungen bis Freitag, 09.12 09:45 Uhr, im Vips-Verzeichnis der StudIP-Veranstaltung ab. Fertigen Sie dazu ihre Hausaufgaben direkt in .pdf Form an oder scannen ihre handschriftlichen Hausaufgaben ein. Geben Sie in Gruppen von **4 Personen** ab.

Aufgabe 1: Eigenschaften regulärer Sprachen [8 Punkte]

Zeigen Sie die folgenden Aussagen:

a) [4 Punkte] Der Kleene-Stern ist tatsächlich ein Hüllen-Operator: $(L^*)^* = L^*$.

b) [4 Punkte] Es gibt für jedes Paar von NFAs A und B (andere) NFA $A \cup B$ und $A.B$, welche $\mathcal{L}(A \cup B) = \mathcal{L}(A) \cup \mathcal{L}(B)$ und $\mathcal{L}(A.B) = \mathcal{L}(A).\mathcal{L}(B)$ erfüllen.

   Hinweis: Geben Sie dazu jeweils eine Konstruktion an, die unabhängig von A und B funktioniert, und beweisen Sie, dass die entsprechenden Sprachen übereinstimmen.

Aufgabe 2: Boolsche Programme sind regulär [9 Punkte]

Wir betrachten Programme nur mit boolschen Variablen. Hier werden Ausdrücke $e \in \text{Exp}$ nichtdeterministisch auf Variabel-Zuständen $\sigma \in \{0,1\}^V$ ausgewertet: Für $v \in \{0,1\}$ soll $S_v(e)$ die Menge der Variablen-Zustände sein, auf denen e den Wert v zurückgeben kann. Für Variable $x \in V$ ist $S_v(x) = \{\sigma \in \{0,1\}^V \mid \sigma(x) = v\}$. Beispielsweise gilt auch $\sigma \in S_v(y \lor \neg z)$ genau dann, wenn $v = \max(\sigma(y), 1 - \sigma(z)) \in \{0,1\}$ ist. Außerdem gibt es den sogenannten *havoc*-Ausdruck $*$, welcher nicht-deterministisch sowohl 0 als auch 1 zurückgeben kann: $S_0(*) = S_1(*) = \{0,1\}^V$.

Sei V die Menge der Variabeln und B die Menge der Blöcke des boolschen Programms P. Einige Wörter aus $\{s\} \cup (V \times \{0,1\})$ entsprechen (endlichen) Ausführungen von P. Es gibt einen NFA $\langle (B \times \{0,1\}^V), \langle b_0, 0^V\rangle, \rightarrow, \{f\} \times \{0,1\}^V\rangle$, dessen Sprache exakt diese Wörter enthält. Dieser startet beim initialen Block $i \in B$ mit allen Variablen auf Wert 0 und akzeptiert auf dem (einzigen) finalen Block $f \in B$, ungeachtet der Werte.

Die Transition $\langle b, \sigma\rangle \xrightarrow{\langle x,v\rangle} \langle c, \tau\rangle$ existiert genau dann, wenn für jedes $y \in V \setminus \{x\}$ die Gleichheit $\sigma(y) = \tau(y)$ gilt, $\tau(x) = v$ ist, $b = [x := e]^\ell$ eine Zuweisung ist, c der (nach Programmrichtung eindeutige) Nachfolger von b ist und wenn $\sigma \in S_v(e)$.

Die Transition $\langle b, \sigma\rangle \xrightarrow{s} \langle c, \tau\rangle$ existiert genau dann, wenn $\sigma = \tau$ gilt, $b = [e]^\ell$ die Bedingung einer Verzweigung oder Schleife ist, und entweder

- c der erste else-Block oder falls dieser nicht existiert, der nächste Block ist und $\sigma \in S_0(e)$ ist.
- c der erste then-Block oder der erste innere Schleifenblock ist und $\sigma \in S_1(e)$ ist.

a) [8 Punkte] Betrachten Sie das folgende boolsche Programm P mit den Blöcken $B = \{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$ und den Variablen $V = \{x, y, z\}$.

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

Transform this boolean while program into a nondeterministic finite automaton A_P that accepts words of $\{s\} \cup V \times \{0, 1\}$: $\Sigma = \{s, x0, x1, y0, y1, z0, z1\}$.

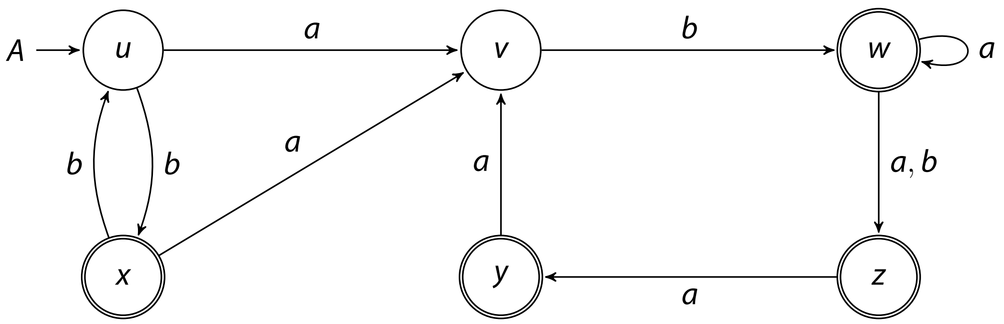
$\varepsilon \notin \mathcal{L}(A_P)$, da jede Ausführung mindestens durch Block b_0 führen muss.

$x1.s \notin \mathcal{L}(A_P)$, da jede Ausführung mit $z = 0$ startet und deswegen mindestens einmal die Schleife durchlaufen muss.

$x1.s.y0.x1.s.z1.s \in \mathcal{L}(A_P)$, da die Ausführungen zuerst 1 und dann 0 lesen können, wodurch die Austrittsbedingung erfüllt wird.

b) [1 Punkt] Ist Ihr Automat A_P *partiell* deterministisch (indem alle fehlenden Kanten zu einem neuen Zustand $\varnothing$ führen)?

Aufgabe 3: Ardens Lemma und Rabin und Scott's Potenzmengenkonstruktion [10 Punkte]
Gegeben sei der folgende NFA A über dem Alphabet $\{a, b\}$:



a) [1 Punkt] Geben Sie das zu A gehörige Gleichungssystem an.

b) [3 Punkte] Bestimmen Sie einen regulären Ausdruck für $\mathcal{L}(A)$, indem Sie das Gleichungssystem unter Verwendung von Ardens Lemma lösen.

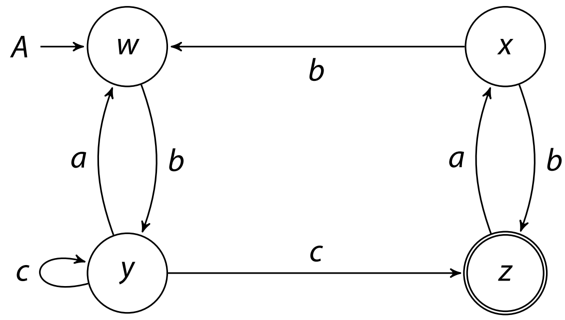
c) [2 Punkte] Geben Sie insbesondere einen Ausdruck für jede Variable an.

d) [3 Punkte] Bestimmen Sie den einen DFA B mit $\mathcal{L}(B) = \mathcal{L}(A)$ mithilfe der Rabin-Scott-Potenzmengenkonstruktion.

e) [1 Punkt] Geben Sie einen regulären Ausdruck für $\overline{\mathcal{L}(A)}$ an.

Aufgabe 4: Homomorphismen [8 Punkte]

Gegeben sei der folgende NFA A über dem Alphabet $\Sigma = \{a, b, c\}$:



Betrachten Sie den folgenden Homomorphismus $f\colon \Sigma \to \{0, 1\}$.

$$f(a) = \varepsilon$$
$$f(b) = 10$$
$$f(c) = 01$$

a) [3 Punkte] Konstruieren Sie den Bild-Automaten $f(A)$ mit $\mathcal{L}(f(A)) = f(\mathcal{L}(A))$ als einen gewöhnlichen NFA.

b) [1 Punkt] Zeigen Sie, dass $1001011010100110 \in \mathcal{L}(f(A))$ gilt, indem Sie einen entsprechenden Lauf durch A angeben.

Betrachten Sie jetzt den folgenden Homomorphismus $g\colon \{d, e\} \to \Sigma$.

$$g(d) = bcc$$
$$g(e) = ab$$

c) [3 Punkte] Konstruieren Sie den Urbild-Automaten $g^{-1}(A)$ mit $\mathcal{L}(g^{-1}(A)) = g^{-1}(\mathcal{L}(A))$ als einen gewöhnlichen NFA.

c) [1 Punkt] Zeigen Sie, dass $deeded \in \mathcal{L}(g^{-1}(A))$ gilt, indem Sie einen entsprechenden Lauf durch A angeben.