GTI Uebungsblatt 3

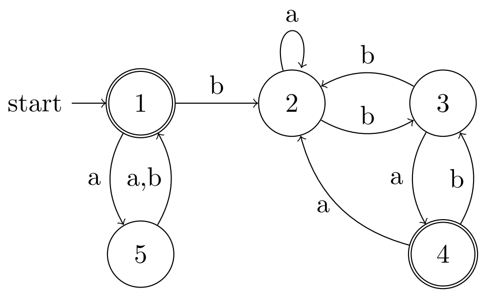
Max Springenberg, 177792

3.1

1.
Das entfernen aller Zustände von A, die nicht von s aus erreichbar sind resultiert in dem Entfernen von 7 und 8.
2.
Die daraus resultierende Relation $N(A)$ und die jewiligen Zustandspaare lassen sich aus folgender Tabelle ablesen.

	1	2	3	4	5	6
1	-	x^0	x^2	x^2	x^0	
2	-	-	x^1	x^0	x^1	x^0
3	-	-	-	x^0	x^1	x^2
4	-	-	-	-	x^1	x^2
5	-	-	-	-	-	x^0
6	-	-	-	-	-	-

3.
Das Verschmelzen der nicht markierten Zustände liefert den Folgenden Automaten.



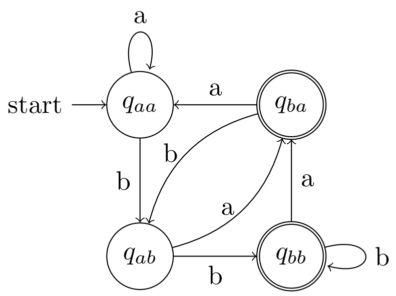
3.2 Sei$L = \{w \in \{a, b\} | |w| > 1$ und der vorletzte Buchstabe in w ist ein b$\}$

3.2.1 Geben Sie fur jede Aquivalenzklasse der Nerode-Relation $\sim_L$ einen Reprasentanten an. Geben Sie auerdem fur je zwei verschiedene dieser Reprasentanten x_i und x_j ein Wort z_{ij} an, das bezeugt, dass x_i und x_j verschiedene Aquivalenzklassen reprasentieren. Es soll also gelten $x_i z_{ij} \in L \Leftrightarrow x_j z_i j \in L$ fur alle Reprasentanten x_i, x_j mit $x_i \neq x_j$.

Mögliche Representationen für die Äquivalenzklassen sind $x_1 = aa, x_2 = ab, x_3 = ba, x_4 = bb$ mit:
$aa \not\sim_L ab$ mit $z = a$
$ba \not\sim_L ab$ mit $z = \epsilon$
$ba \not\sim_L aa$ mit $z = \epsilon$
$bb \not\sim_L aa$ mit $z = \epsilon$
$bb \not\sim_L ab$ mit $z = \epsilon$
$bb \not\sim_L ba$ mit $z = a$

Da dies vier Representationen, zu je einer Äquivalenzklasse angegeben wurden und nach Aufgabenstellung nur 4 Äquivalenzklassen existieren, wurde zu jeder Äquivalenzklasse eine Representation angegeben.

3.2.2 Geben Sie einen minimalen DFA A an, so dass $L(A) = L$ gilt. Begründen Sie sowohl, dass A die Sprache L entscheidet, als auch, dass A minimal ist.



$L(A) = L$:

$L(A) \subseteq L$:
Annahme $L(A) \nsubseteq L$, dann $\exists w \in L(A) : w \notin L$

Alle Wörter $w \in L(A)$ sind länger als 1, da es mindestens 2 Transitionen bedarf um in einen akzeptierenden Zustand zu wechseln. Des weiteren gilt, dass alle Wörter genau dann akzeptiert werden, wenn die vorletzte Transition nach einlesen aller Zeichen, durch ein b erfolgte.
1. Fall $|w| \leq 1$:
$w \notin L(A) \land w \notin L$
2. Fall $w = va\sigma, v \in \{a,b\}^*, \sigma \in \{a,b\}$:
$w \notin L(A) \land w \notin L$
$\lightning$ es muss gelten $L(A) \subseteq L$

$L \subseteq L(A)$:
Annahme $L \nsubseteq L(A)$, dann $\exists w \in L : w \notin L(A)$

Alle Wörter aus L sind definiert als länger als 1 und mit einem b als vorletztes Zeichen.
1. Fall $|w| \leq 1$:
Wörter aus $L(A)$ muessen länger als 1 sein, da es mindestens zwei Transitionen bedarf um in einen akzeptierenden Zustand zu wechseln.
$w \notin L \land w \notin L(A)$
2. Fall $w = va\sigma, v \in \{a,b\}^*, \sigma \in \{a,b\}$:
Es wird nur in einen akzeptierenden Zustand gewechselt, wenn die Vorletzte Transition durch ein b erfolgte.
$w \notin L(A) \land w \notin L$
$\lightning$ es muss gelten $L \subseteq L(A)$

Damit muss dann auch gelten $L(A) = L$

Ein minimaler DFA hat soviele Zustände, wie Äquivalenzklassen zu der Nerode Relation, der durch diesen entschiedene Sprache existieren. A hat vier Zustände und es wurde gezeigt, dass $L(A) = L$

gilt und dass L vier Äquivalenzklassen enthaelt. Damit ist A minimal.

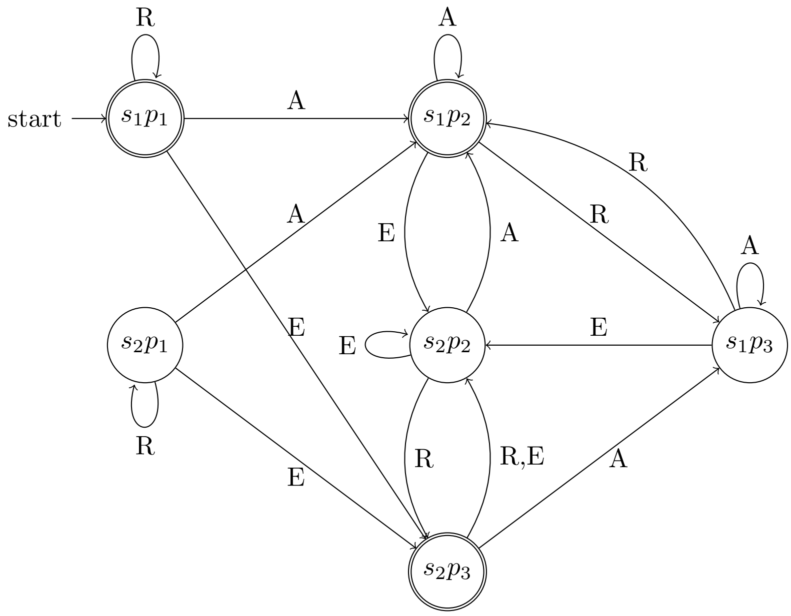
3.3

3.3.1

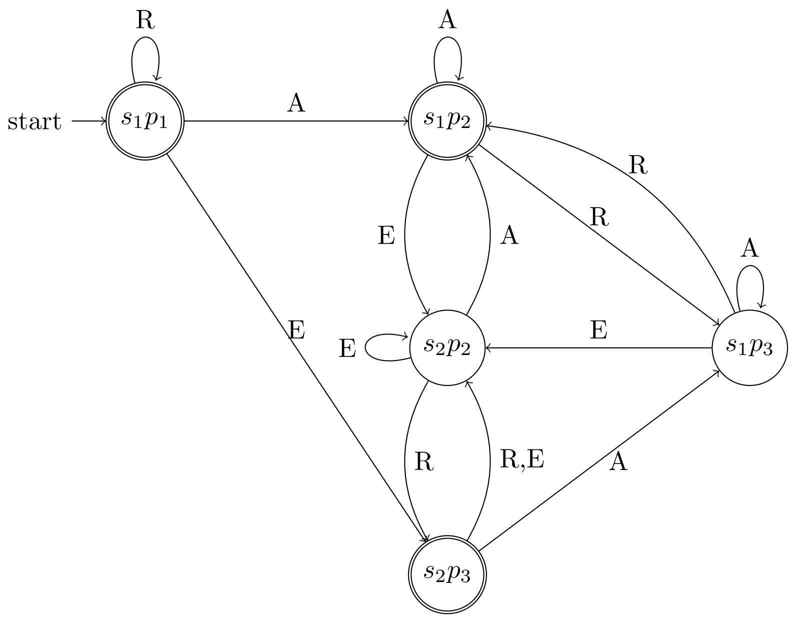
Der Produktautomat A mit den Akzeptierenden Zuständen

$$F_A = \{(p, q) \in Q_S \times Q_P | (q \in F_S \wedge p \notin F_P) \vee (q \notin F_S \wedge p \in F_P)\}$$

ergibt sich zu:



Ohne den unerreichbaren Zustand s_2p_1:



3.3.2

Die Sprache $L(A)$ ist nicht leer, damit sind die Automaten nach dem Äquivalenztest der Vorlesung über einen Produktautomaten mit akzeptierenden Zuständen bei zusammen gedassten Zuständen mit ungleicher Akzeptiereigenschaft nicht äquivalent.

Ferner erfüllt P damit auch nicht die Spezifikation S.

3.3.3

Es bietet sich, wie auch in den vorhergehenden Teilaufgaben die Konstruktion eines Produktautomaten und ein Leerheitstest für diesen an.

Wie in der Vorlesung vorgestellt existiert ein Algorithmus für den Leerheitstest mit:

1. Vergiss die Kantenmarkierung
2. Füge einen Zielknoten t und für jeden akzeptierenden Zustand eine Transition zu t hinzu.
3. Teste, ob ein Weg von s noch t existiert
4. Wenn ja, so gilt $L(A) \neq \emptyset$

In der Laufzeit $O(|\delta|)$
Die Konstruktion eines Produktautomaten A aus den Automaten A_1, A_2 kann durch $A = A_1 \times A_2$ mit

$$F_A = \{(p, q) \in Q_1 \times Q_2 | (q \in F_1 \wedge p \notin F_2) \vee (q \notin F_1 \wedge p \in F_2)\}$$

Nach vorlesung beläuft sich diese Konstruktion in ihrer Laufzeit auf $O(|Q_1| \times |Q_2| \times |\Sigma|)$

Der Algorithmus umfasst diese Konstruktion gefolgt von dem Leerheitstest und beläuft dementsprechend auf eine Laufzeit von

$$O(|Q_1| \times |Q_2| \times |\Sigma| + |\delta|) = O(|Q_1| \times |Q_2| \times |\Sigma|)$$

3.3.4

Zunächst wandelt man jeden RA α_i in einen $\epsilon - NFA$ um, mit dem Baukastenprinzip beläuft sich das auf $O(|\alpha_i|)$
Im folgendem wird der Produktautomat mit durch die $\epsilon - NFA$s konstruiert und deer Leerheitstest berechnet.
Da $O(|\alpha|)$ linear ist bleibt die Laufzeit bei

$$O(|Q_1| \times |Q_2| \times |\Sigma|)$$