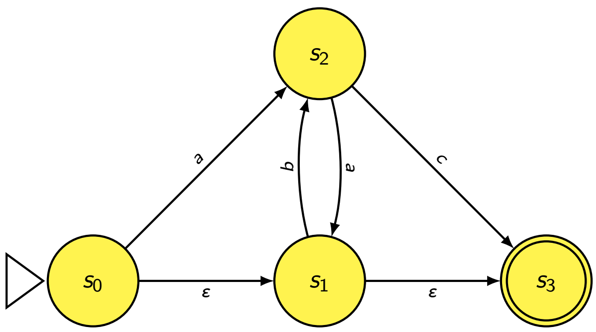
A100 – Formale Grundlagen der Informatik I
Beispielaufgaben zur Klausurvorbereitung*

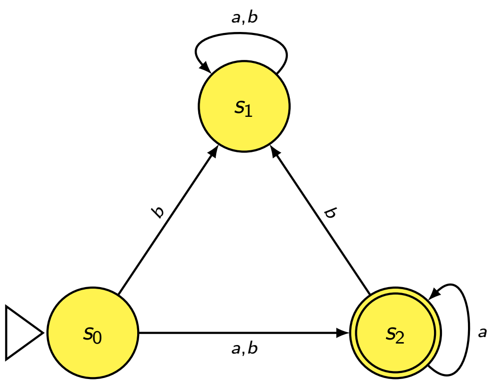
T21 (2022)

1. Wandeln Sie den folgenden ε-Automaten in einen äquivalenten gewöhnlichen endlichen Automaten um:



   - Bei systematischer Anwendung aller Schritte des allgemeinen Algorithmus wird einer der ursprünglich definierten Zustände entfernt. Skizzieren Sie ein entsprechendes Zwischenergebnis, markieren Sie den zu entfernenden Zustand, und begründen Sie kurz, warum die Entfernung den Automaten nicht wesentlich verändert.
   - Zeichnen Sie den fertig umgewandelten Automaten.
   - Geben Sie alle akzeptierten Wörter der Länge <3 in Standardordnung an.

2. Wandeln Sie den folgenden nichtdeterministischen in einen äquivalenten deterministischen Automaten um:

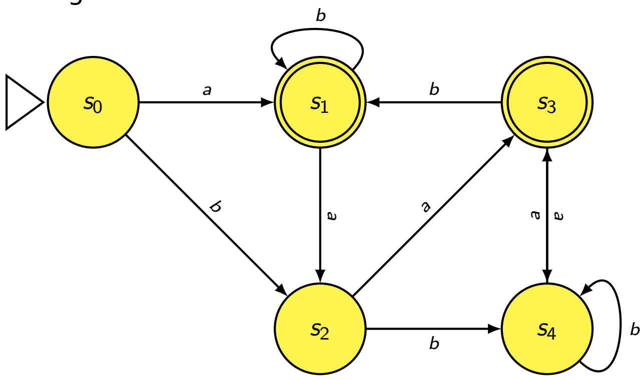


   - Geben Sie die vollständige formale Definition des gegebenen Automaten als Quintupel an.

*Der Gesamtumfang entspricht rund zweimal dem der 60-minütigen Teilklausur

- Erstellen Sie eine Übergangstabelle für den umgewandelten Automaten.
- Zeichnen Sie den umgewandelten Automaten.

3. Minimieren Sie den folgenden endlichen Automaten:



- Welche Bedingungen stellt der Minimierungsalgorithmus an seine Eingabe?
- Prüfen Sie, ob der gegebene Automat diese erfüllt; andernfalls ergänzen Sie den Automaten zunächst entsprechend.
- Erstellen Sie eine Unterscheidungsmatrix für die Zustände des Automaten.
- Zeichnen Sie den minimierten Automaten.

4. Gegeben sei das Alphabet $\Sigma = \{a, b\}$. Geben Sie reguläre Ausdrücke für folgende Sprachen an:

   (a) Alle Wörter aus Σ^*, die zwei aufeinanderfolgende as enthalten.
   (b) Alle Wörter aus Σ^*, die *nicht* zwei aufeinanderfolgende as enthalten.
   (c) Alle Wörter aus Σ^*, die mit verschiedenen Buchstaben beginnen und enden.

5. Gegeben sei die folgende Typ-3-Grammatik:

$$\begin{array}{llll} S \to aS & A \to aA & B \to bB & C \to c \\ S \to aA & A \to aB & B \to bC & \end{array}$$

- Geben Sie die vollständige formale Definition der Grammatik als Quadrupel an.
- Handelt es sich um eine links- oder rechtslineare Grammatik?
- Wandeln Sie diese in eine äquivalente Grammatik der jeweils umgekehrten Art um.
- Geben Sie einen äquivalenten regulären Ausdruck an.
- Skizzieren Sie einen äquivalenten endlichen Automaten.

6. Gegeben sei die folgende Typ-2-Grammatik:

$$\begin{array}{llll} S \to VC & V \to aV & C \to RRC & R \to r \\ & V \to \varepsilon & C \to \varepsilon & \end{array}$$

- Ist diese Grammatik in Greibach-Normalform? Falls nein, markieren Sie die nicht-konformen Regeln.
- Wandeln Sie die Grammatik in eine äquivalente Chomsky-Normalform um.

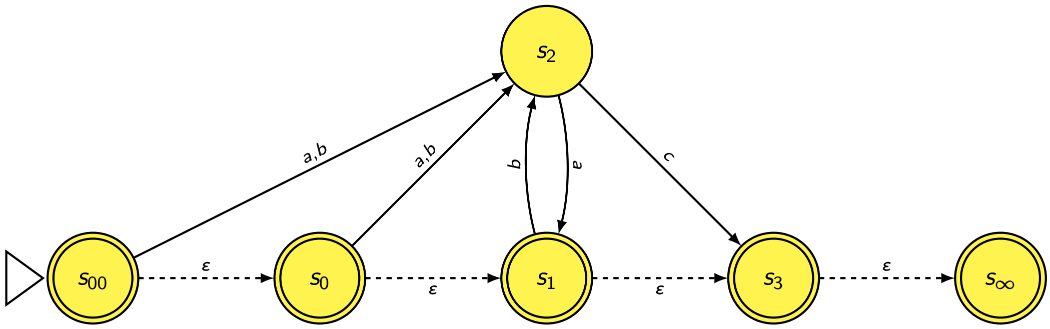
7. Gegeben sei die folgende Sprache über dem Alphabet $\Sigma = \{a, b, c, d\}$:

$$L = \left\{ w \in \Sigma \;\middle|\; w \in \{a, b\}^+\{c, d\}^+; |w|_a = |w|_c \right\}$$

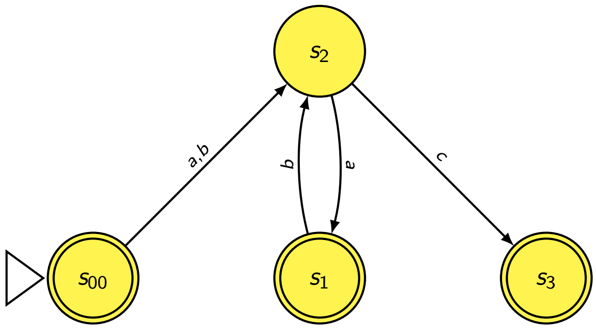
- Konstruieren Sie eine Typ-2-Grammatik, die diese Sprache erzeugt.
- Konstruieren Sie einen Kellerautomaten, der diese Sprache akzeptiert.
- Skizzieren Sie grob, wie man mit dem Pumping-Lemma zeigen würde, dass diese Sprache nicht regulär ist. Der Beweis muss nicht formal exakt ausgeführt werden.

Musterlösungen

1. Bei der Entfernung der ε-Übergänge wird s_0 unerreichbar:



Der neue Startzustand s_{00} übernimmt dessen Funktion aber vollständig.



Akzeptierte Wörter der Länge <3 in Standardordnung: ε, aa, ac, ba, bc

2. Formale Darstellung:

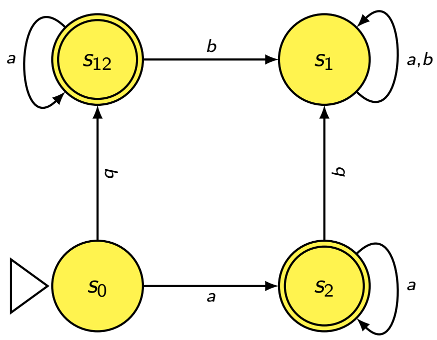
$$A = (\Sigma, S, \delta, s_0, F)$$

$$\Sigma = \{a, b\}$$
$$S = \{s_0, s_1, s_2\}$$
$$\delta = \left\{ \begin{array}{c} (s_0, a) \mapsto \{s_2\}, (s_0, b) \mapsto \{s_1, s_2\} \\ (s_1, a) \mapsto \{s_1\}, (s_1, b) \mapsto \{s_1\} \\ (s_2, a) \mapsto \{s_2\}, (s_2, b) \mapsto \{s_1\} \end{array} \right\}$$
$$F = \{s_2\}$$

Potenzmengenkonstruktion, Übergangstabelle:

	a	b	$\in F?$
$\{s_0\}$	$\{s_2\}$	$\{s_1, s_2\}$	
$\{s_2\}$	$\{s_2\}$	$\{s_1\}$	•
$\{s_1\}$	$\{s_1\}$	$\{s_1\}$	
$\{s_1, s_2\}$	$\{s_1, s_2\}$	$\{s_1\}$	•

Diagramm:



3. Ein zu minimierender Automat muss deterministisch und vollständig sein. Das ist hier schon der Fall, wie man aus der Übergangstabelle gut sieht:

	a	b
s_0	s_1	s_2
s_1	s_2	s_1
s_2	s_3	s_4
s_3	s_4	s_1
s_4	s_3	s_4

(Andernfalls wäre zunächst die Pontenzmengenkonstruktion eines äquivalenten DEA bzw. die Einführung eines Totzustandes nötig.)

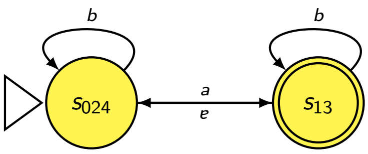
Unterscheidungstabelle für den Markierungsalgorithmus; markiere zunächst Paare von akzeptierendem und nicht akzeptierendem Zustand:

	s_0	s_1	s_2	s_3
s_1	■			
s_2		■		
s_3	■		■	
s_4		■		■

Bei diesem Automaten kann keine weitere Markierung gefunden werden! Es ergibt sich die Partitionierung:

$$\Pi = \big\{\{s_0, s_2, s_4\}, \{s_1, s_3\}\big\}$$

Minimierter Automat:



4. (a) Alle Wörter aus Σ^*, die zwei aufeinanderfolgende as enthalten:

$$(a \mid b)* \cdot aa \cdot (a \mid b)*$$

(b) Alle Wörter aus Σ^*, die *nicht* zwei aufeinanderfolgende as enthalten:

$$(ab \mid b)* \cdot (a \mid J)$$

(c) Alle Wörter aus Σ^*, die mit verschiedenen Buchstaben beginnen und enden:

$$a(a \mid b)b* \mid b(a \mid b)*a$$

5. Formale Darstellung:

$$G = (N, \Sigma, P, S)$$

$$N = \{S, A, B, C\}$$
$$\Sigma = \{a, b, c\}$$
$$P = \big\{(S, aS), (S, aA), (A, aA), (A, aB), (B, bB), (B, bC), (C, c)\big\}$$

Diese Grammatik ist rechtslinear. Zur Umwandlung muss die letzte Regel umgeformt werden:

$$C \to cT$$
$$T \to \varepsilon$$

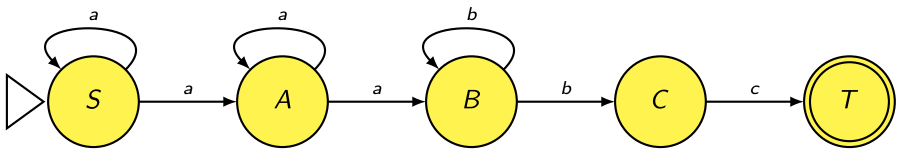
Dann können alle Regeln umgekehrt werden, T wird neues Startsymbol:

$$S \to Sa \qquad A \to Aa \qquad B \to Bb \qquad T \to Cc$$
$$A \to Sa \qquad B \to Aa \qquad C \to Bb \qquad S \to \varepsilon$$

Direkt abgelesener regulärer Ausdruck (könnte noch vereinfacht werden):

$$(a*)a(a*)a(b*)bc$$

Äquivalenter Automat:



6. Nein, die Regeln

$$S \to VC \qquad V \to \varepsilon \qquad C \to RRC \qquad C \to \varepsilon$$

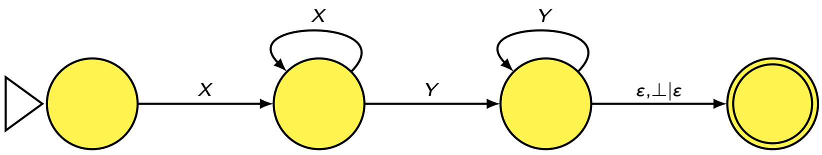
sind in der Greibach-Normalform nicht erlaubt. Die äquivalente Chomsky-Normalform:

$$\begin{array}{lllll} S \to VC & V \to V_aV & C \to RX & R \to r & X \to RC \\ S \to \varepsilon & S \to V_aV & S \to RX & V_a \to a & \end{array}$$

7. Grammatik:

$$\begin{array}{llll} S \to BaBTDcD & T \to BaBTDcD & B \to bB & D \to dD \\ & T \to \varepsilon & B \to \varepsilon & D \to \varepsilon \end{array}$$

Kellerautomat:



wobei

$$X = \begin{array}{c} a, \bot \mid a\bot \\ a, a \mid aa \\ b, \bot \mid \bot \\ b, a \mid a \end{array} \qquad Y = \begin{array}{c} c, a \mid \varepsilon \\ d, a \mid a \\ d, \bot \mid \bot \end{array}$$

In der negativen Form des Pumping-Lemmas wähle für jede Zahl n das Wort $x = a^n b^n$. Dann gilt für jede in Frage kommende Zerlegung $x = uvw$ mit $|uv| \leq n$, dass v nur aus as besteht. Dann ist nach dem Aufblasen zu uv^iw mit $i \neq 1$ die Anzahl nicht mehr gleich der Anzahl der bs.