Probeklausur Automatentheorie & Formale Sprachen WiSe 2012/13, Wiebke Petersen

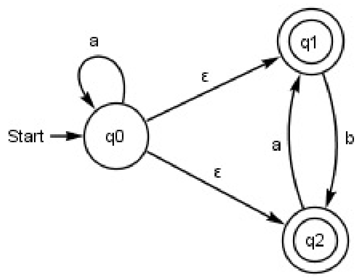
Name:

Matrikelnummer:

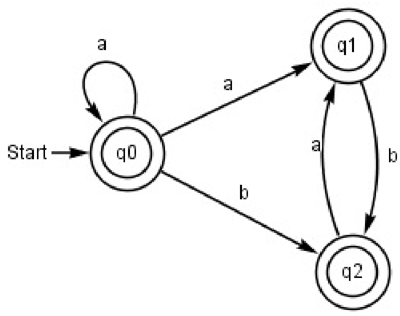
Aufgabe A: Typ3-Sprachen

1. Konstruieren Sie einen endlichen Automaten, der die Sprache aller Wörter über dem Alphabet $\{a, b\}$ akzeptiert, die mindestens drei a's enthalten. Es genügt, wenn Sie das Übergangsnetz des Automaten zeichnen. Vergessen Sie bitte nicht, End- und Startzustände zu markieren.

2. Gegeben sei folgender nichtdeterministischer endlicher Automat mit ϵ-Übergängen:



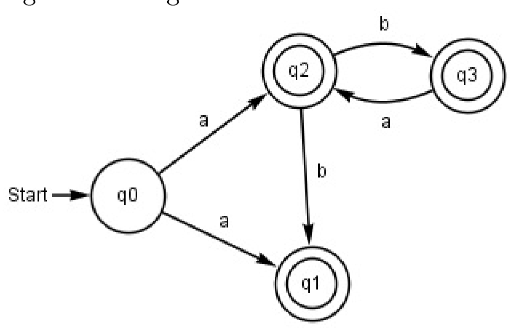
Lösung:



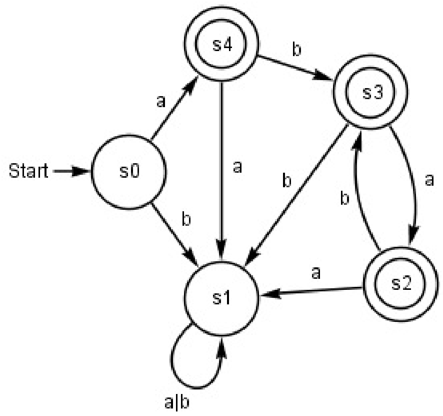
   (a) Zeichnen Sie einen äquivalenten endlichen Automaten ohne ϵ-Übergänge.

   (b) Geben Sie den Automaten zusätzlich als 5-Tupel an.

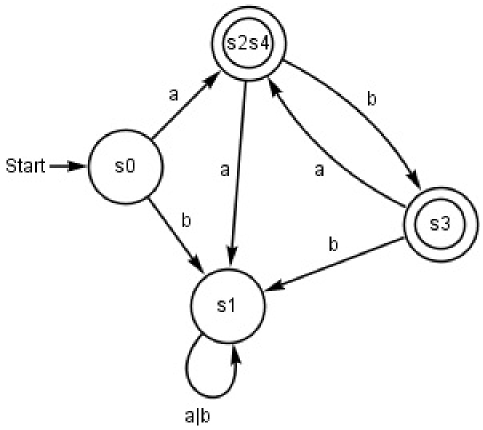
3. Gegeben sei folgender nichtdeterministischer endlicher Automat:



   (a) Konstruieren sie einen äquivalenten DEA mit minimaler Zustandszahl.
   Lösung:



DEA:



MinDEA:

(b) Geben Sie einen regulären Ausdruck an, der die von dem Automaten akzeptierte Sprache beschreibt.
Lösung:
$a(ba)^*(b|\epsilon)$

(c) Geben Sie eine rechtslineare Grammatik an, die dieselbe Sprache generiert, die der Automat akzeptiert.
Lösung:

$$\begin{array}{rcl} S & \rightarrow & aT|a \\ T & \rightarrow & bR|b \\ R & \rightarrow & aT|a \end{array}$$

4. Ist folgende Sprache regulär? Beweisen Sie Ihre Antwort.

   (a) Sprache aller Wörter über dem Alphabet $\{a, b\}$, die wenigstens ein a mehr als b's enthalten.

   (b) Sprache aller Wörter über dem Alphabet $\{a, b\}$, in denen auf jedes b ein a folgt.

   (c) Sprache aller Wörter über dem Alphabet $\{a, b\}$, die mit ab beginnen und die wenigstens drei a's beinhalten.

Aufgabe B: Typ2-Sprachen und Grammatiken

1. Geben Sie die folgende Grammatik in der Chomsky-Normalform an:

$$\begin{array}{rcl} S & \rightarrow & aaB|bA \\ A & \rightarrow & aB \\ B & \rightarrow & AC|a \\ C & \rightarrow & AC|\epsilon \end{array}$$

Lösung:
Schritt 1: Elimination der ϵ-Regeln.

$$\begin{aligned} S &\rightarrow aaB|bA \\ A &\rightarrow aB \\ B &\rightarrow AC|a|A \\ C &\rightarrow AC|A \end{aligned}$$

Schritt 2: Elimination der Kettenregeln.

$$\begin{aligned} S &\rightarrow aaB|bA \\ A &\rightarrow aB \\ B &\rightarrow AC|a|aB \\ C &\rightarrow AC|aB \end{aligned}$$

Schritt 3: Entfernung nutzloser Variablen.
in dem Beispiel gibt es keine nutzlosen Variablen.

Schritt 3: Regeln für die Terminalsymbole.

$$\begin{aligned} S &\rightarrow V_aV_aB|V_bA \\ A &\rightarrow V_aB \\ B &\rightarrow AC|a|V_aB \\ C &\rightarrow AC|V_aB \\ V_a &\rightarrow a \\ V_b &\rightarrow b \end{aligned}$$

Schritt 4: Reduktion auf binäre rechte Regelseiten.

$$\begin{aligned} S &\rightarrow X_{V_aV_a}B|V_bA \\ A &\rightarrow V_aB \\ B &\rightarrow AC|a|V_aB \\ C &\rightarrow AC|V_aB \\ X_{V_aV_a} &\rightarrow V_aV_a \\ V_a &\rightarrow a \\ V_b &\rightarrow b \end{aligned}$$

2. Gegeben sei die folgende Grammatik mit Startsymbol S:

$$\begin{aligned} S &\rightarrow aA|bB \\ A &\rightarrow aAA|bS|b \\ B &\rightarrow bBB|aS|a \end{aligned}$$

   (a) Wählen Sie ein Wort das von der Grammatik generiert wird und mindestens die Länge 4 hat und zeichnen Sie den Ableitungsbaum.
   (b) Beschreiben Sie die von der Grammatik generierte Sprache in eigenen Worten.
   Lösung:

   Die Menge aller Wörter über dem Alphabet $\{a, b\}$, in denen genausoviele a's wie b's vorkommen.
   (c) Prüfen Sie mit dem CYK-Algorithmus, ob die Grammatik die Worte 'abab' und 'ababa' generiert. Fassen Sie in einem Satz das Ergebnis zusammen.

a	b	a	b

a	b	a	b	a

Lösung:

$$\begin{aligned} S &\rightarrow aA|bB \\ A &\rightarrow aAA|bS|b \\ B &\rightarrow bBB|aS|a \end{aligned}$$

CNF:

$$\begin{aligned} S &\rightarrow V_aA|V_bB \\ A &\rightarrow X_{V_aA}A|V_bS|b \\ B &\rightarrow X_{V_bB}B|V_aS|a \\ X_{V_bB} &\rightarrow V_bB \\ X_{V_aA} &\rightarrow V_aA \\ V_a &\rightarrow a \\ V_b &\rightarrow b \end{aligned}$$

a	b	a	b
B, V_a	A, V_b	B, V_a	A, V_b
S, X_{V_aA}	S, X_{V_bB}	S, X_{V_aA}	
B	A		
S, X_{V_aA}			

CYK gelingt, das Wort lässt sich aus S generieren.

a	b	a	b	a
B, V_a	A, V_b	B, V_a	A, V_b	B, V_a
$S, X_{V_a A}$	$S, X_{V_b B}$	$S, X_{V_a A}$	$S, X_{V_b B}$	
B	A	B		
$S, X_{V_a A}$	$S, X_{V_b B}$			
B				

CYK scheitert, das Wort lässt sich aus B aber nicht aus S generieren.

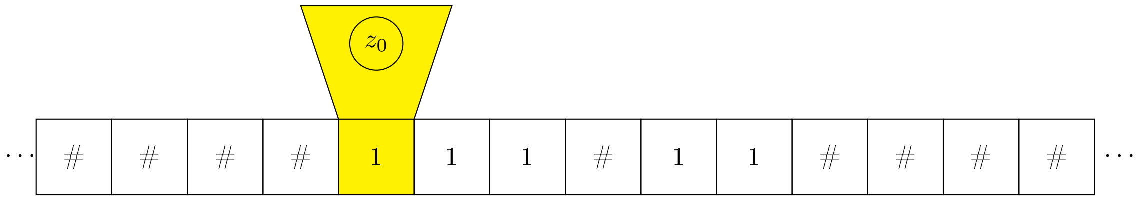
3. Zeigen Sie, dass die folgende Sprache nicht kontextfrei ist: $L = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$
4. Konstruieren Sie einen Kellerautomaten zu einer gegebenen Sprache und stellen Sie die Verarbeitung eines gegebenen Wortes dar (Aufgabe 4.15). Gibt es einen deterministischen Kellerautomaten zu der Sprache?
5. Begründen Sie, warum die Klasse der kontextfreien Sprachen abgeschlossen ist unter Konkatenation.

Aufgabe C: Typ0-Sprachen

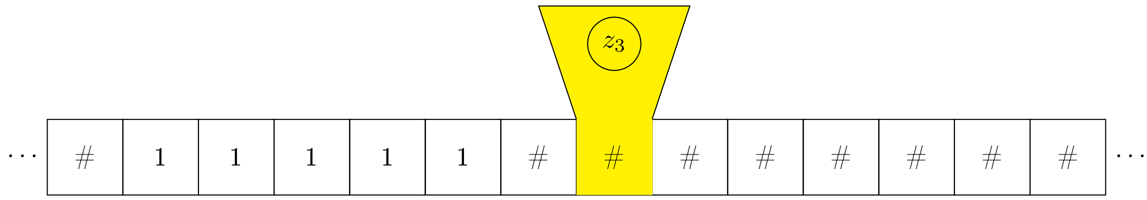
Gegeben sei die Turingmaschine mit $Z = \{z_0, z_1, z_2, z_3\}$, $\Sigma = \{1\}$, $\Gamma = \{1, \#\}$, Startzustand z_0, Endzustandsmenge: $\{z_3\}$, Leersymbol $\#$ und Übergangsfunktion δ mit:

$$\begin{array}{rcl}
\delta(z_0, 1) & = & (z_0, 1, R) \\
\delta(z_0, \#) & = & (z_1, 1, R) \\
\delta(z_1, 1) & = & (z_1, 1, R) \\
\delta(z_1, \#) & = & (z_2, \#, L) \\
\delta(z_2, 1) & = & (z_3, \#, R)
\end{array}$$

Die Turingmaschine startet mit der folgenden Eingabe:



1. Tragen Sie die Bandbeschriftung und die Position des Lesekopfes sowie den aktuellen Zustand zum Ende der Berechnung bei der Eingabe der oben abgebildeten Ausgangsbandbeschriftung in die folgende Skizze ein.



2. Geben sie ausgehend von der Ausgangskonfiguration $(\#, z_0, 11\#11)$ die Konfigurationsübergänge bis zum Erreichen des Endzustands an.
   Lösung:
   $(\#, z_0, 11\#11) \to (1, z_0, 1\#11) \to (11, z_0, \#11) \to (111, z_1, 11) \to (1111, z_1, 1) \to (11111, z_1, \#) \to (1111, z_2, 1\#) \to (1111\#, z_3, \#)$

3. Welche Berechnung führt die Turingmaschine durch? Beschreiben sie bitte genau die Bedingungen für die Bandbeschriftung zu Beginn der Berechnung und beschreiben sie die Bandbeschriftung zum Ende der Berechnung.

4. Bitte nutzen Sie für die folgenden Aufgaben diese Gödelkodierung:
   $X_1 = 0,\ X_2 = 1,\ X_3 = \#$
   $z_0 = Z_1,\ z_1 = Z_2,\ z_2 = Z_3,\ z_3 = Z_4$
   $L = R_1,\ R = R_2$
   $\delta(Z_i, X_j) = (Z_k, X_l, R_m)$ wird kodiert als $0^i10^j10^k10^l10^m$.

   (a) Geben Sie die Gödelnummern für die folgenden beiden Übergänge an:
   $\delta(z_0, 1) = (z_0, 1, R)$ und $\delta(z_2, \#) = (z_1, 1, L)$.

   Lösung:
   Die Gödelnummer für $\delta(z_0, 1) = (z_0, 1, R)$ ist 010010100100
   und für $\delta(z_2, \#) = (z_1, 1, L)$ 000100010010010.

   (b) Bitte dekodieren Sie den folgenden Übergang: 010001001001001

   Lösung:
   $\delta(z_0, \#) = (z_1, 1, R)$

5. Wie würden Sie eine Turingmaschine konstruieren, die zwei Zahlen in Unärcode multipliziert? Beschreiben sie den erwarteten Input und beschreiben Sie, wie die Multiplikation berechnet wird. Sie müssen die Maschine nicht formal spezifizieren.

6. Geben Sie die formale Spezifikation einer Turingmaschine an, die die
   Lösung:
   Nachfolgefunktion im Unärcode berechnet. Die Nachfolgefunktion wandelt zum Beispiel den Input #111# in #1111# um. Um sich die Aufgabe zu vereinfachen dürfen Sie die Funktion einschränken und bestimmte Eingaben ausschließen (vielleicht mag ihre Maschine ja zum Beispiel nur Zahlen größer 10?). Ihre Maschine sollte allerdings für unendlich viele Eingabewerte die korrekte Ausgabe produzieren. Geben Sie bitte die Anforderungen an die Eingabe an.
   Der Input besteht aus einem Band, auf dem eine Zahl im Binärcode steht. Der Lesekopf ist am Anfang über einem beliebigen Zeichen der Zahl.
   Die Turingmaschine hat z_1 als Endzustand und lediglich zwei Übergänge, nämlich: $\delta(z_0, 1) = (z_0, 1, R)$ und $\delta(z_0, \#) = (z1, 1, R)$.

Aufgabe D: Querbeet

Beachten Sie bitte, dass es für falsche Antworten Punktabzug gibt, daher lohnt sich Raten nicht. Für jede richtige Antwort gibt es einen Punkt und für jede falsche einen Punkt Abzug. Unbeantwortete Fragen geben keinen Punktabzug (aber natürlich auch keine Punkte).

1. Sind die folgenden Aussagen wahr oder falsch?

	wahr/falsch
Jede reguläre Sprache ist kontextfrei	