Informatik I WS 07/08

Tutorium 24

10.01.08

Bastian Molkenthin

E-Mail: infotut@sunshine2k.de

Web: http://infotut.sunshine2k.de

Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Anmeldung IPK

Eine *inoffizielle* Info-1 Probeklausur findet am Samstag, 19.1.2008 statt.

Wann?	13.00 Uhr - ca. 14.30 Uhr
Wo?	Audimax

⇨ Anmeldung notwendig! Bis spätestens Do, 17.1.08

⇨ http://anmeldungipk.webhop.net

Was braucht man?

- Studentenausweis
- Kuli, Füller (blau oder schwarz)
- Unkostenbeitrag von 1 Euro

Keine anderen Hilfsmittel erlaubt!

Chomsky - Grammatiken

- (Chomsky-) Grammatiken besitzen einen ähnlichen Aufbau wie schon die Semi-Thue-Systeme und Markov-Algorithmen:
  - Der Zeichenvorrat wird unterteilt in die Mengen der Terminalsymbole (**Σ**) und der Nichtterminalsymbole (**N**).
  - Regeln nennen wir Produktionen und sind in der Menge **P**.
  - Alle Ableitungen beginnen bei dem Startsymbol (ein *Nicht-Terminal!*) **A** (auch Axiom genannt)
- Der Viertupel aus Σ, N, P und Axiom bildet die Grammatik:

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

- Sämtliche Worte aus Σ^*, welche sich mit den gegebenen Regeln aus A ableiten lassen, bilden die von der Grammatik erzeugte Sprache:

$$L(G) = \{w \in \Sigma^* \mid A \Rightarrow^* w\}$$

Chomsky - Hierarchie

- **Chomsky-0-Grammatik (CH-0)**
  - allgemeiner Produktionstyp: $l \to r$ wobei $l, r \in V^*$ beliebig
  - insbesondere auch ε-Produktionen ($r = \varepsilon$)
- **Chomsky-1-Grammatik (CH-1)**
  - längenbeschränkt: $l \to r$ wobei $l, r \in V^*$, $1 \leq |l| \leq |r|$
  - kontextsensitiv: $uAv \to urv$ wobei $A \in N$, $u, v \in V^*$, $r \in V^+$, d.h. $r \neq \varepsilon$
- **Chomsky-2-Grammatik (CH-2)**
  - kontextfrei: $A \to r$ wobei $A \in N$, $r \in V^*$
- **Chomsky-3-Grammatik (CH-3)**
  - entweder linkslinear: $A \to Bx$ oder $A \to x$ wobei $A, B \in N$, $x \in \Sigma$
  - oder rechtslinear: $A \to xB$ oder $A \to x$ wobei $A, B \in N$, $x \in \Sigma$

Aufgabe Chomsky

Gegeben sei die Grammatik $G = (\Sigma, N, P, A)$ mit

- $\Sigma = \{a, b, c\}$
- $N = \{A, B, C\}$
- $P = \{ \quad A \rightarrow B \mid C, \quad B \rightarrow bBc \mid bc, \quad C \rightarrow aC \mid aB \quad \}$

a) Gebe den maximalen Chomsky-Typ der Grammatik an! CH-2

b) Geben sie die von der oben angegebenen Grammatik G erzeugte Sprache L(G) an.

$L(G) = \{a^n b^m c^m \mid n,m \in \mathbb{N}_0, n \geq 0, m \geq 1\}$

c) Geben sie eine Grammatik $G = (\Sigma, N, P, A)$ vom maximalen Chomsky–Typ an, die die Sprache $L(G) = \{a^n b^i c^j \mid n, i, j\ n \in \mathbb{N}_0, n, i, j \geq 1\}$ erzeugt. Dabei ist $\Sigma = \{a, b, c\}$. Geben sie den Chomsky–Typ ihrer Grammatik an.

- $G = (\Sigma, N, P, A)$
- $\Sigma = \{a, b, c\}$
- $N = \{A, B, C\}$
- $P = \{ \quad A \rightarrow aA \mid aB, \quad B \rightarrow bB \mid bC, \quad C \rightarrow cC \mid c \quad \}$

CH-3

Rückblick EBNF

Metazeichen dienen zur Beschreibung der Grammatikregeln, durch die die zu einer Sprache gehörenden Sätze festgelegt werden

Aufbau der EBNF:

=	trennt linke und rechte Regelseite
.	schließt Regel ab

|	trennt Alternativen
Beispiel: x | y	beschreibt: x, y

()	klammert Alternativen
Beispiel: (x | y) z	beschreibt: xz, yz

[]	wahlweises Vorkommen
Beispiel: [x] y	beschreibt: xy, y

{}	kein- bis n-maliges Vorkommen
Beispiel: {x} y	beschreibt: y, xy, xxy, xxxxy, ...

Rückblick EBNF (2)

Es wird zwischen Terminal- und Nichtterminalzeichen unterschieden.

Terminalzeichen:	“a“... “z“	⟵ In Anführungszeichen!
Nichtterminalzeichen:	Zahl, Ziffer	

Zusammenhang EBNF ⟷ Chomsky-Hierarchie:

Kontextfreie Sprachen (CH-2) können in EBNF formuliert werden!

EBNF Aufgabe

Gebe, falls möglich, die von folgender Grammatik erzeugte Sprache in EBNF an.

$G_1 = (\Sigma, N, P, A)$ mit

- $\Sigma = \{a, c\}$
- $N = \{A, B, C\}$
- $P = \{ \quad A \rightarrow B,$
  $B \rightarrow CC,$
  $B \rightarrow a,$
  $C \rightarrow c \}$

Lösung:

A = B.
B = CC | “a“.
C = “c“.

EBNF Aufgabe (2)

Gebe, falls möglich, die von folgender Grammatik erzeugte Sprache in EBNF an.

$G_2 = (\Sigma, N, P, A)$ mit

- $\Sigma = \{a, c\}$
- $N = \{A, B, C\}$
- $P = \{$ $A \rightarrow aB \mid bCA,$
$cB \rightarrow bB,$
$aB \rightarrow Ca,$
$C \rightarrow c\ \}$

Lösung: Nicht möglich, $L(G_2)$ nicht kontextfrei.

EBNF Aufgabe (3)

Gebe zu der vorliegenden EBNF eine Grammatik an. (A ist Startsymbol)

A = (AT)|(A“ts“)|({“b“}“b“).
T = “a”

Lösung: $G_3 = (\Sigma, N, P, A)$ mit

- $\Sigma = \{a, b, t, s\}$
- $N = \{A, B, T\}$
- $P = \{ \quad A \rightarrow AT \mid Ats \mid B, \quad B \rightarrow Bb \mid b, \quad T \rightarrow a \quad \}$

Definition – Reguläre Ausdrücke

Reguläre Ausdrücke sind eine verbreitete und geeignete Notation, um reguläre Sprachen (CH-3) zu beschreiben. Ein regulärer Ausdruck R über einem Zeichenvorrat C ist rekursiv definiert durch:

- Für jedes $c \in C$ ist c ein regulärer Ausdruck
- Ist R ein regulärer Ausdruck, dann auch (R)*.
- Sind R und S reguläre Ausdrücke, so sind auch (RS) und (R + S) reguläre Ausdrücke.
- Klammern darf man ggf. weglassen.
- Außerdem gelten folgende Vorrangregeln:
    - * bindet stärker als Verkettung
    - Verkettung (RS) bindet stärker als ”oder“ (R + S)

Aufgabe – Reguläre Ausdrücke

Gib für folgende Sprachen einen regulären Ausdruck an, sowie eine EBNF-Regel, welche die Sprache aus dem Startsymbol S erzeugt.

$$L = \{a^3 b^{3n} c^m \mid n, m \in \mathbb{N}_0, m \leq 2\} \cup \{d^j \mid j \in \mathbb{N}, j \geq 2\}$$

Lösung: EBNF: A = "aaa" {"bbb"} ["c"] ["c"] | "dd" {"d"} .

Regulärer Ausdruck: $aaa(bbb)^*(\varepsilon + c + cc) + ddd^*$

$$L_2 = \{w \in \Sigma^* : |w| \geq 2\} \text{ mit } \Sigma = \{a, b, c\}.$$

Lösung: EBNF: A = (”a” | ”b” | ”c”)(”a” | ”b” | ”c”){”a” | ”b” | ”c”}.

Regulärer Ausdruck: $(a+b+c)(a+b+c)(a+b+c)^*$

Aufgabe – EBNF & RegEx

Formen Sie folgenden regulären Ausdruck in eine EBNF–Regel (Startsymbol A) um:

(((a+b)*(cd)*)+e)*

Lösung: A = {({"a" | "b"}{"cd"}) | "e"}.

Formen Sie die durch folgende EBNF–Regel beschriebene Sprache in einen regulären Ausdruck um:

A = "a" | ("abc" ["a" | "b"] A).

Lösung: (abc + abca + abcb)*a

Einordnung Endlicher Automat

- Ein endlicher Automat ist eine Art von (abstrakter) Maschine.
- Es besteht ein enger Zusammenhang zwischen den durch Grammatiken erzeugten Sprachen und den Maschinen:
    - Maschine soll zu einer als Eingabe anliegenden Zeichenreihe feststellen, ob diese Zeichenreihe zur Sprache gehört oder nicht.
- Zu jedem Chomsky-Grammatiktyp lässt sich ein diese Sprachklasse bearbeitbarer Maschinentyp angeben:
    - CH-0 Turing-Maschine
    - CH-1 Linear beschränkter Automat
    - CH-2 Kellerautomat
    - CH-3 Endlicher Automat

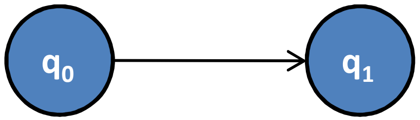
Endlicher Automat

Bestandteile:

- *endliche Menge Q von Zuständen* mit einem Anfangszustand $q_0 \in Q$
- *Zeichenvorrat Σ*
- *Zustandsübergänge*: Lesen eines Zeichens $a \in \Sigma$ führt zu einem Zustandsübergang vom aktuellen Zustand $q \in Q$ in einen neuen Zustand $q' \in Q$
    - Notation: $qa \rightarrow q'$
    - bei gegebenem q bestimmt a den Nachfolgerzustand bzw. bei gegebenem a hängt die Wirkung q´ vom bisherigen Zustand q ab
    - der Zustand läßt sich als ein (endliches) Gedächtnis über die Vorgeschichte und die bisher eingegebenen Zeichen auffassen

Ein endlicher Automat kann als Graph aufgefasst werden:

- Zustände $Q = \{q_0, q_1, \ldots, q_n\}$ des endlichen Automaten lassen sich als Ecken eines Graphen auffassen
- Zustandsübergänge $q_i a \rightarrow q_j$ mit $a \in \Sigma$ entsprechen markierte gerichtete Kanten

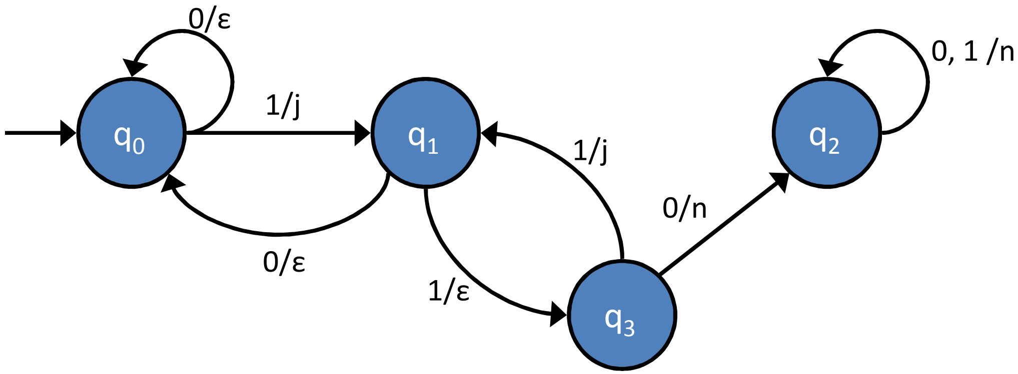


Moore/Mealy Automaten

Mealy-Automat:

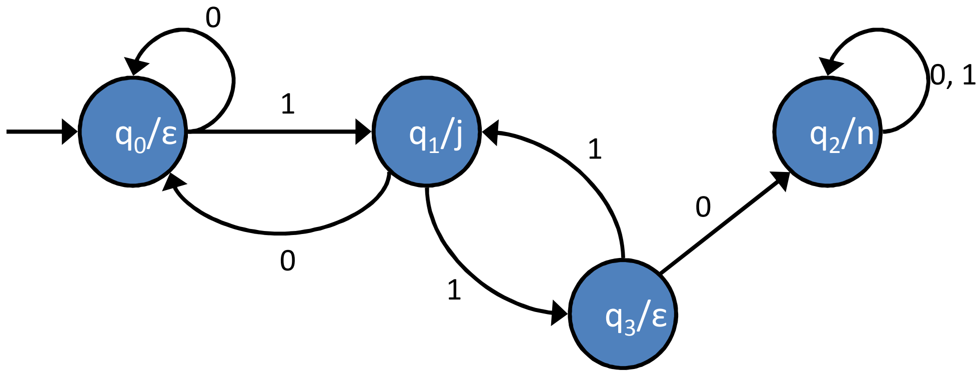
Erzeugung einer Ausgabe bei jedem Zustandsübergang

Markieren der Kanten mit a / t



Moore-Automat:

Erzeugung einer Ausgabe bei Erreichen eines Zustandes



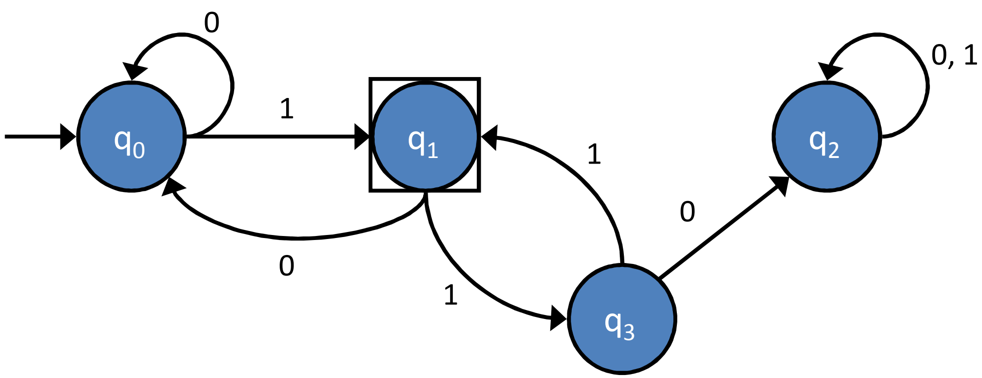
In *beiden* Fällen :

Ausgabe ist ein Wort $t=t_0 \ldots t_{n-1}$ über einem Ausgabezeichenvorrat T. Die Ausgabe kann offensichtlich nicht länger sein als das Eingabewort.

Akzeptor

Akzeptor:

- häufigster Spezialfall eines Moore-Automaten
- Ausgabe nicht bei allen Zuständen
- Zustände $F \in Q$, bei denen eine Ausgabe erfolgt, heißen Endzustände
- Ausgegebenes Wort $t \in T^n$ hängt vom erreichten Endzustand $q \in F$ ab



Akzeptor (2)

Ein Automat heißt ein *vollständiger Akzeptor*, wenn die Übergangsmatrix vollständig um Übergänge in einen Fehlerzustand ergänzt ist.

Ein endlicher Akzeptor lässt sich als Quintupel (Σ, Q, q_0, F, P) auffassen:

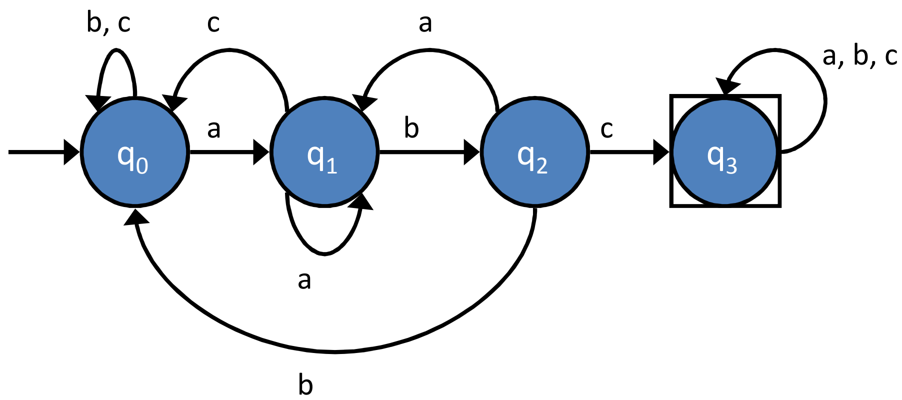
- Σ : Zeichenvorrat
- Q : nichtleere endliche Zustandsmenge
- q_0: Anfangszustand aus Q
- F: nichtleere Menge von Endzuständen aus Q
- P: Übergänge $q_a \to q'$ mit $q, q' \in Q$, $a \in \Sigma$

Sprache, die der Akzeptor akzeptiert:

$$L(A) = \{x \mid x \in \Sigma^*, q_0x \Rightarrow^* q_e, q_e \in F\}$$

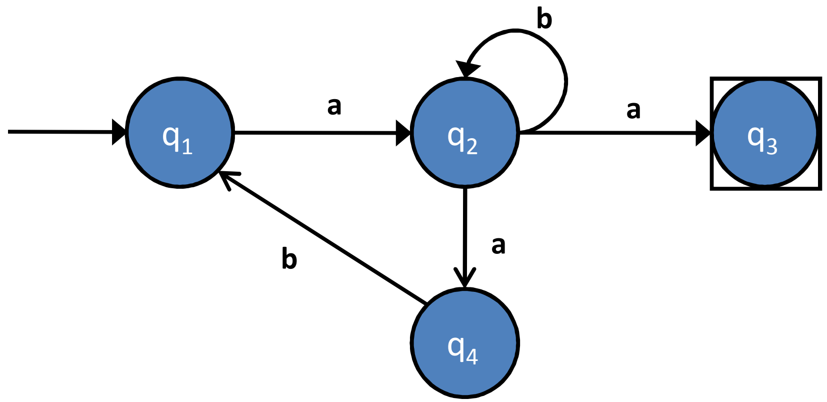
Aufgabe – Endlicher Automat

Erstelle einen vollständigen, deterministischen Akzeptor, der sämtliche Worte über dem Alphabet $\Sigma = \{a, b, c\}$ akzeptiert, welche die Zeichenfolge "abc" enthalten!



Aufgabe – Graphen und Automaten

Gegeben ist folgender Graph, der einen endlichen Automaten darstellt:



a) Gebe die formale Beschreibung des endl. Automaten an!

$A=(\Sigma,Q,q_1,F,P)$, $\Sigma=\{a,b\}$, $Q=\{q_1,q_2,q_3,q_4\}$, $F=\{q_3\}$, Startzustand = q_1

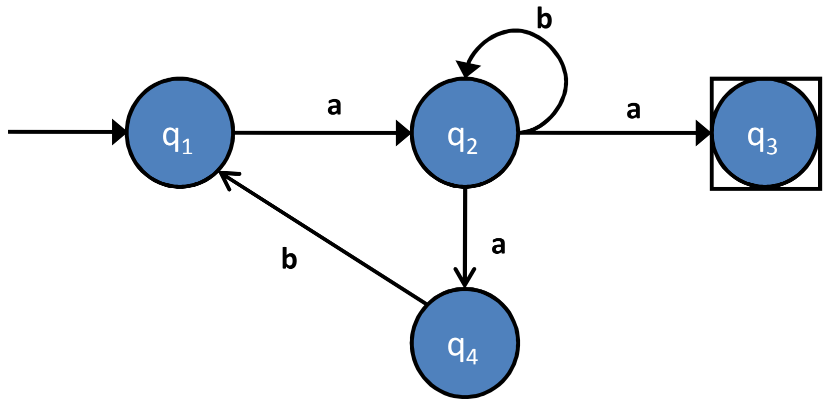
$P=\{ q_1a\rightarrow q_2, q_2a\rightarrow q_3, q_2a\rightarrow q_4, q_2b\rightarrow q_2, q_4b\rightarrow q_1\}$

b) Um was für einen Automatentyp handelt es sich hier? Begründe die Antwort!

Es handelt sich um den häufigsten Spezialfall eines Moore-Automaten, den Akzeptor, da keine Ausgaben bei den Zustandsübergängen (Mealy) vermerkt sind.
Die Ausgabe des Automaten erfolgt dabei nicht bei allen Zuständen sondern nur bei den Endzuständen.

Aufgabe – Graphen und Automaten (2)

Gegeben ist folgender Graph, der einen endlichen Automaten darstellt:



c) Ist der Automat vollständig? Begründe die Antwort!

Nein, beispielsweise fehlt im vierten Zustand ein Verhalten für die Eingabe a.

d) Ist der Automat deterministisch? Begründe die Antwort!

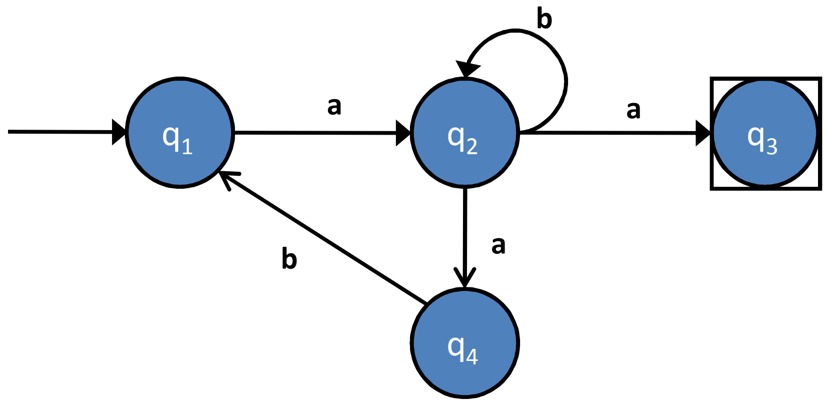
Nein, im zweiten Zustand ist das Verhalten für die Eingabe a nicht eindeutig!

e) Welche Sprache akzeptiert der Automat?

L={a(b*[aba])*a}

Aufgabe – Graphen und Automaten (3)

Gegeben ist folgender Graph, der einen endlichen Automaten darstellt:



f) Welchen Chomsky-Typ hat die Sprache? Begründe die Antwort!

CH-3, da sie von einem endlichen Automaten akzeptiert wird!

g) Geben Sie die Grammatik zu der akzeptierten Sprache an!

$G_2 = (\Sigma, N, P, A)$ mit

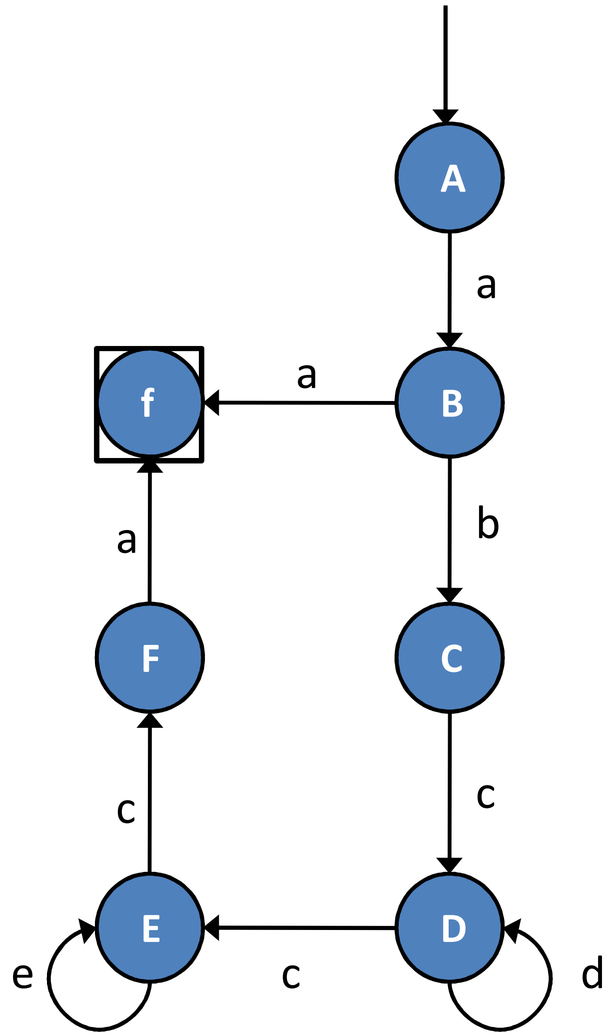
- $\Sigma = \{a, b\}$
- $N = \{A, B\}$
- $P = \{$ $A \rightarrow aB,$ $B \rightarrow bB \mid aC \mid a,$ $C \rightarrow bD,$ $D \rightarrow aB \}$

Aufgabe Akzeptor

Gib den Graphen des Akzeptors für die durch G erzeugte Sprache an.

G = (Σ, N, P, A) mit

- Σ = {a, b, c, d, e}
- N = {S, A, B, C, D, E}
- P = { A → aB,
  B → bC | a,
  C → cD,
  D → dD | cE,
  E → eE | cF,
  F → a}

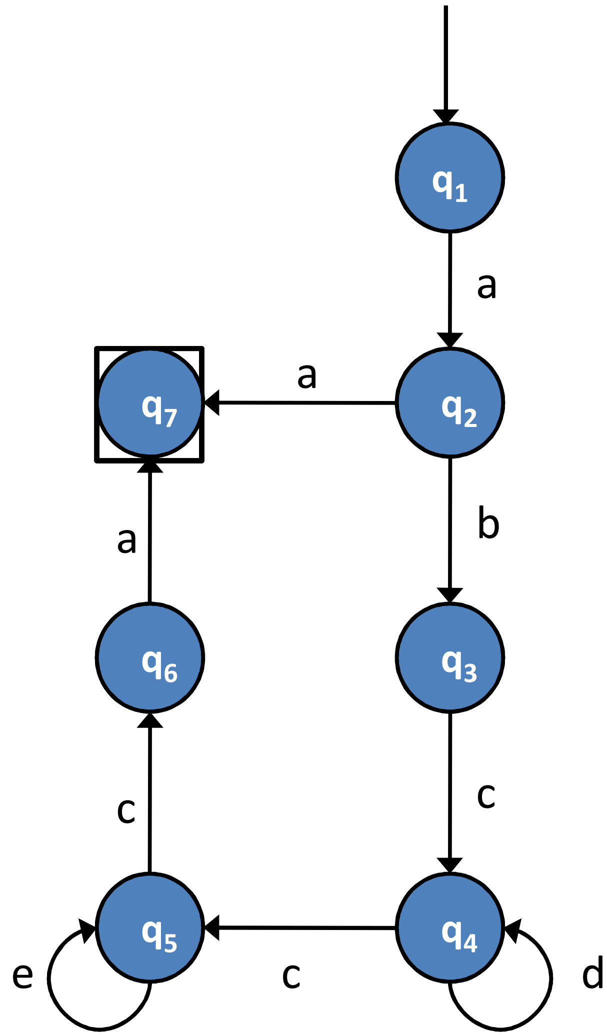


Aufgabe Akzeptor

Gib den Graphen des Akzeptors für die durch G erzeugte Sprache an.

G = (Σ, N, P, A) mit

- Σ = {a, b, c, d, e}
- N = {S, A, B, C, D, E}
- P = { A → aB,
  B → bC | a,
  C → cD,
  D → dD | cE,
  E → eE | cF,
  F → a}



Werbung ;-)

Eulenfest: 15. Januar, Infobau

Fragen ???

Viel Spaß mit dem Übungsblatt!