# Klausur Informatik III

2. Oktober 2004

# Musterlösung

**Aufgabe 1** (16 Punkte)

Es sei $\mathcal{L} = \{a^n b^{n+1} \mid n \geq 0\}$.

a) [3] Geben Sie eine kontextfreie Grammatik $\Gamma$ über dem Terminalalphabet $\{a, b\}$ an mit $L(\Gamma) = \mathcal{L}$.

b) [6] Zeigen Sie mit Hilfe des Earley-Algorithmus, daß $abb \in L(\Gamma)$.

c) [4] Geben Sie einen Kellerautomaten $\mathcal{A}$ an, der diese Sprache durch den Nullkeller akzeptiert, d.h. $N(\mathcal{A}) = \mathcal{L}$.

d) [3] Zeigen Sie, daß $\mathcal{L}$ nicht regulär ist.

**Lösung:**

a) $\Gamma = \langle \{S\}, \{a, b\}, \{S \to aSb, S \to b\}, S \rangle$

b) Füge die Produktion $\overline{S} \to S$ hinzu. Es ergeben sich folgende Zustandsmengen:

$$
\begin{array}{lclllll}
Q_0 & = & \{ & (\overline{S} \to \bullet S; 0), & & (0,1) & \text{Initialisierung} \\
 & & & (S \to \bullet aSb; 0), & & (0,2) & \text{(V) aus (0,1)} \\
 & & & (S \to \bullet b; 0) & \} & (0,3) & \text{(V) aus (0,1)} \\
Q_1 & = & \{ & (S \to a \bullet Sb; 0), & & (1,1) & \text{(E) aus (0,2)} \\
 & & & (S \to \bullet aSb; 1), & & (1,2) & \text{(V) aus (1,1)} \\
 & & & (S \to \bullet b; 1) & \} & (1,3) & \text{(V) aus (1,1)} \\
Q_2 & = & \{ & (S \to b \bullet; 1)\} & & (2,1) & \text{(E) aus (1,3)} \\
 & & & (S \to aS \bullet b; 0) & \} & (2,2) & \text{(R) aus (2,1) und (1,1)} \\
Q_3 & = & \{ & (S \to aSb \bullet; 0), & & (3,1) & \text{(E) aus (2,2)} \\
 & & & (\overline{S} \to S \bullet; 0) & \} & (3,2) & \text{(R) aus (3,1) und (1,0)}
\end{array}
$$

c) $\mathcal{A} = \langle \{q\}, \{a, b\}, \{a, b, S\}, \delta, q, S, \emptyset \rangle$, wobei $\delta$ gegeben ist durch

$$
\begin{array}{lcl}
\delta(q, \epsilon, S) & := & \{(q, aSb), (q, b)\} \\
\delta(q, a, a) & := & \{(q, \epsilon)\} \\
\delta(q, b, b) & := & \{(q, \epsilon)\}
\end{array}
$$

d) Betrachte die syntaktische Kongruenz $\equiv_{\mathcal{L}}$ auf $\{a, b\}^*$. Für alle natürlichen Zahlen $i, j$ mit $i \neq j$ ist $a^i \not\equiv_{\mathcal{L}} a^j$, denn es ist z.B. $a^i b^{i+1} \in \mathcal{L}$, aber $a^j b^{i+1} \notin \mathcal{L}$. Also liegen $a^i$ und $a^j$ in verschiedenen Äquivalenzklassen von $\equiv_{\mathcal{L}}$. Demnach hat $\equiv_{\mathcal{L}}$ unendlich viele Äquivalenzklassen, und darum kann $\mathcal{L}$ nicht regulär sein (Satz von Myhill/Nerode).

Natürlich könnte man an dieser Stelle auch mit Hilfe des Pumping-Lemmas argumentieren. Das Argument auf Seite 33 des Skriptums für die Sprache $\{a^n b^n : n \geq 0\}$ gilt unverändert auch für $\mathcal{L}$.

**Aufgabe 2** (8 Punkte)

Gegeben sei die Grammatik $\Gamma = \langle \{A, D, K, L, S\}, \{a, b, c, \neg, \vee, \wedge\}, \Pi, S \rangle$, wobei $\Pi$ folgende Produktionen umfaßt:

$$\begin{array}{lcl} S & \rightarrow & D \\ D & \rightarrow & L \mid L \vee D \\ K & \rightarrow & L \mid L \wedge K \\ L & \rightarrow & A \mid \neg A \\ A & \rightarrow & a \mid b \mid c \end{array}$$

Konstruieren Sie eine zu $\Gamma$ äquivalente Grammatik in Chomsky-Normalform.

**Lösung:**

$\Gamma'' = \langle \{A, D, K, L, S, N, O, U, X, Y\}, \{a, b, c, \neg, \vee, \wedge\}, \Pi'', S \rangle$ mit den Produktionen

$$\begin{array}{lcl} S & \rightarrow & a \mid b \mid c \mid NA \mid LX \\ D & \rightarrow & a \mid b \mid c \mid NA \mid LX \\ K & \rightarrow & a \mid b \mid c \mid NA \mid LY \\ L & \rightarrow & a \mid b \mid c \mid NA \\ A & \rightarrow & a \mid b \mid c \\ N & \rightarrow & \neg \\ O & \rightarrow & \vee \\ U & \rightarrow & \wedge \\ X & \rightarrow & OD \\ Y & \rightarrow & UK \end{array}$$

ist eine zu $\Gamma$ äquivalente Grammatik in Chomsky-Normalform.

Der ausführliche Lösungsweg:

- $\Gamma$ ist $\epsilon$-frei.
- Eliminiere die Kettenproduktionen.

  Modifiziere die Menge der Produktionen $\Pi$ sukzessive auf folgende Weise:

  1. Entferne $L \rightarrow A$ und füge $L \rightarrow a \mid b \mid c$ hinzu.
  2. Entferne $K \rightarrow L$ und füge $K \rightarrow a \mid b \mid c \mid \neg A$ hinzu.
  3. Entferne $D \rightarrow L$ und füge $D \rightarrow a \mid b \mid c \mid \neg A$ hinzu.
  4. Entferne $S \rightarrow D$ und füge $S \rightarrow a \mid b \mid c \mid \neg A \mid L \vee D$ hinzu.

- Die Grammatik $\Gamma'$ unterscheidet sich von $\Gamma$ nur durch ihre Produktionen $\Pi'$. Diese sind:

$$\begin{array}{lcl} S & \rightarrow & a \mid b \mid c \mid \neg A \mid L \vee D \\ D & \rightarrow & a \mid b \mid c \mid \neg A \mid L \vee D \\ K & \rightarrow & a \mid b \mid c \mid \neg A \mid L \wedge K \\ L & \rightarrow & a \mid b \mid c \mid \neg A \\ A & \rightarrow & a \mid b \mid c \end{array}$$

- Nun kann die eigentliche Konstruktion der CNF-Grammatik erfolgen.

  1. Für jedes Terminalzeichen, das zusammen mit anderen Zeichen auf der rechten Seite einer Produktion auftaucht, wird eine neue Variable eingeführt: $N$ für $\neg$, $O$ für $\vee$ und $U$ für $\wedge$. Die Produktionen $N \rightarrow \neg$, $O \rightarrow \vee$ und $U \rightarrow \wedge$ werden zu $\Pi'$ hinzugefügt. Es ergeben sich folgende Produktionen:

$$\begin{array}{lcl} S & \rightarrow & a \mid b \mid c \mid NA \mid LOD \\ D & \rightarrow & a \mid b \mid c \mid NA \mid LOD \\ K & \rightarrow & a \mid b \mid c \mid NA \mid LUK \\ L & \rightarrow & a \mid b \mid c \mid NA \\ A & \rightarrow & a \mid b \mid c \\ N & \rightarrow & \neg \\ O & \rightarrow & \vee \\ U & \rightarrow & \wedge \end{array}$$

  2. Die Produktionen $S \rightarrow LOD$, $D \rightarrow LOD$ und $K \rightarrow LUK$ habe noch nicht die richtige Form. Führe neue Variablen $X, Y$ ein und nimm die Produktion $X \rightarrow OD$ und $Y \rightarrow UK$ in die Menge der Produktionen auf. Ersetze $S \rightarrow LOD$ durch $S \rightarrow LX$, $D \rightarrow LOD$ durch $D \rightarrow LX$ und $K \rightarrow LUK$ durch $K \rightarrow LY$.

**Aufgabe 3** (8 Punkte)

Geben Sie Maschinenschemata an, die folgende Funktionen berechnen:

a) [4] die totale Vorgängerfunktion

$$V(x) := \begin{cases} x - 1 & \text{falls } x > 0 \\ 0 & \text{falls } x = 0 \end{cases}$$

b) [4] die partielle Vorgängerfunktion

$$V_p(x) := \begin{cases} x - 1 & \text{falls } x > 0 \\ \text{undefiniert} & \text{falls } x = 0 \end{cases}$$

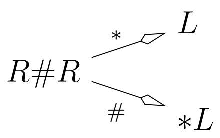
Beachten Sie dabei diese beiden Spezifikationen:

- Eine Berechnung führt eine Anfangskonfiguration $(\underline{\#}w)$ in eine Haltekonfiguration $(\underline{\#}v)$ über.
- Eine natürliche Zahl $n$ werde *unär* codiert, d.h. durch $n + 1$ aufeinanderfolgende Markierungen $\underbrace{*\cdots*}_{n+1}$ auf dem Band dargestellt.

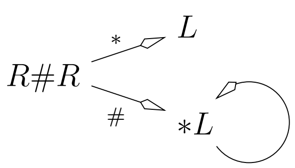
Hinweis: Es ist Ihnen gestattet, bei höchstens *einer* der Teilaufgaben eine Turingmaschine anstatt eines Maschinenschemas anzugeben.

**Lösung:**

a) das Maschinenschema zur Berechnung von $V$:




b) das Maschinenschema zur Berechnung von $V_p$:




Zunächst muß eine einzelne Markierung vom Band gelöscht werden. Dies wird von dem Schema $R\#$ erledigt, welches zunächst eine "gehe.nach.rechts"-Maschine startet und unmittelbar danach eine "schreibe.#"-Maschine. Das weitere Verhalten der Automaten, welches zur Unterscheidung eines Eingabewertes $x = 0$ von Eingabewerten $x > 0$ erforderlich ist, wird durch Inspektion des Bandinhaltes zur Rechten ausgewählt. Dazu wird zunächst der Arbeitskopf durch eine $R$-Maschine einen weiteren Schritt nach rechts bewegt. Dann ergibt sich folgende Fallunterscheidung:

- Sollte sich auf der aktuellen Bandposition eine Markierung befinden, dann war der Eingabewert $> 0$. Der Arbeitskopf muß wieder nach links gefahren werden, was eine $L$-Maschine erledigt. Dadurch ist die Endkonfiguration erreicht.
- Sollte sich auf der Bandposition keine Markierung befinden, dann war anfangs nur ein $*$ auf dem Band, d.h. der Eingabewert war $= 0$.

  a) Für die totale Vorgängerfunktion ist es erforderlich, wieder eine Markierung auf das Band zu schreiben, nach links zu gehen und zu terminieren. Dies erledigt das Teilschema $*L$.

  b) Die partielle Vorgängerfunktion darf in diesem Falle jedoch kein Ergebnis liefern! Es muß also verhindert werden, daß die Maschine terminiert. Man erreicht dies, indem man die finale $L$-Maschine aus der Teilaufgabe a) durch einen zyklischen Aufruf immer wieder startet. Auf diese Weise erhält man die Lösung der Teilaufgabe b) durch Hinzufügen von nur einem einzigen Übergang zur Lösung von a). Selbstverständlich ergibt sich durch jede andere Realisierung einer Endlosschleife nach der Feststellung, dass die Eingabe nur aus einer Markierung bestand, genauso das gewünschte Verhalten.

**Aufgabe 4** (4 Punkte)

Geben Sie eine partiell rekursive Funktionsdefinition für $x \dot{-} y$ an, so daß folgende Gleichung erfüllt ist:

$$x \dot{-} y := \begin{cases} x - y & \text{falls } x \geq y \\ \text{undefiniert} & \text{sonst} \end{cases}$$

Sie dürfen dazu verwenden, daß $+$ eine partiell rekursive Funktion und $=$ ein partiell rekursives Prädikat ist.

**Lösung:**

Durch $\mu$-Rekursion kann nach der kleinsten natürlichen Zahl $z$ gesucht werden, welche die Gleichung $x = y + z$, (also $x - y = z$) erfüllt. Falls $x \geq y$, dann gibt es eine solche Zahl $z$, und sie wird durch die $\mu$-Rekursion gefunden. Falls $x < y$, dann gibt es keine Zahl $z$, welche die Gleichung erfüllt, d.h. eine $\mu$-rekursive Suche, die, bei 0 beginnend, jede natürliche Zahl als Kandidaten für die Lösung der o.g. Gleichung untersucht, wird nicht terminieren.

Die Lösung der Aufgabe lautet demnach:

$$x \dot{-} y := (\mu z)(x = y + z)$$

**Aufgabe 5** (4 Punkte)

Welche der folgenden Behauptungen über Mengen von natürlichen Zahlen ist richtig?

a) [2] $A$ ist genau dann primitiv rekursiv, wenn sowohl $A$ als auch $\overline{A}$ primitiv rekursiv aufzählbar sind.

b) [2] $A$ ist genau dann rekursiv, wenn sowohl $A$ als auch $\overline{A}$ rekursiv aufzählbar sind.

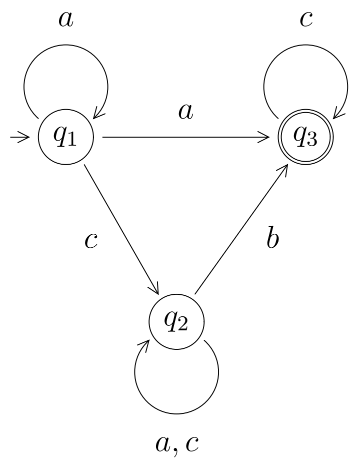
Begründen Sie Ihre Antworten unter Verweis auf Sätze der Vorlesung.

**Lösung:**

Aussage b) gilt nach Theorem 15.6 des Skriptums. Weiterhin sind nach Theorem 15.4 "rekursiv aufzählbar" und "primitiv rekursiv aufzählbar" äquivalente Begriffe, während Definition 15.5 die Begriffe "rekursiv" sowie "primitiv rekursiv" auf die entsprechenden Eigenschaften von Prädikaten zurückführt, welche *nicht* äquivalent sind. Demzufolge muß Aussage a) falsch sein.

**Aufgabe 6** (10 Punkte)

Geben Sie zu dem abgebildeten NDEA $\mathcal{A}$ einen regulären Ausdruck $\gamma$ mit $\langle\gamma\rangle = L(\mathcal{A})$ an. Stellen Sie dazu entweder das zu $\mathcal{A}$ gehörende Gleichungssystem auf und lösen Sie dieses, oder verwenden Sie die Kleene-Myhill Konstruktion der $R^k_{i,j}$-Mengen.




**Lösung:**

- Ermittlung von $\gamma$ durch Lösen des Gleichungssystems:

  Das Gleichungssystem für $\mathcal{A}$ lautet:

  $$\begin{aligned} X_1 &= aX_1 \cup cX_2 \cup aX_3 \\ X_2 &= (a \cup c)X_2 \cup bX_3 \\ X_3 &= cX_3 \cup \epsilon \end{aligned}$$

  Die Lösung der dritten Gleichung lautet nach Arden's Regel $X_3 = c^*\epsilon$, d.h. $X_3 = c^*$. Einsetzung in die beiden ersten Gleichungen liefert:

  $$\begin{aligned} X_1 &= aX_1 \cup cX_2 \cup ac^* \\ X_2 &= (a \cup c)X_2 \cup bc^* \end{aligned}$$

  Die Lösung der zweiten Gleichung lautet nach Arden's Regel $X_2 = (a \cup c)^*bc^*$. Einsetzung in die erste Gleichungen liefert:

  $$X_1 = aX_1 \cup c(a \cup c)^*bc^* \cup ac^*$$

  Die Lösung dieser Gleichung ist $X_1 = a^*(c(a \cup c)^*bc^* \cup ac^*)$. Damit ist der gesuchte reguläre Ausdruck $\gamma = a^*(c(a \cup c)^*bc^* \cup ac^*)$. Durch Ausklammern von $c^*$ kann man den Ausdruck zu $\gamma = a^*(c(a \cup c)^*b \cup a)c^*$ vereinfachen.

- Ermittlung von $\gamma$ durch Konstruktion der $R^k_{i,j}$-Mengen:

  Es müssen nicht alle $R^k_{i,j}$-Mengen berechnet werden. Vielmehr genügt es, ausgehend von $R^3_{1,3}$ (Startzustand $q_1$, Endzustand $q_3$) zunächst rekursiv die Dependenzen aufzulösen:

$$
\begin{array}{rcl}
R^3_{1,3} &=& R^2_{1,3} \cup (R^2_{1,3} \circ (R^2_{3,3})^* \circ R^2_{3,3}) \\[1ex]
R^2_{1,3} &=& R^1_{1,3} \cup (R^1_{1,2} \circ (R^1_{2,2})^* \circ R^1_{2,3}) \\
R^2_{3,3} &=& R^1_{3,3} \cup (R^1_{3,2} \circ (R^1_{2,2})^* \circ R^1_{2,3}) \\[1ex]
R^1_{1,2} &=& R^0_{1,2} \cup (R^0_{1,1} \circ (R^0_{1,1})^* \circ R^0_{1,2}) \\
R^1_{1,3} &=& R^0_{1,3} \cup (R^0_{1,1} \circ (R^0_{1,1})^* \circ R^0_{1,3}) \\
R^1_{2,2} &=& R^0_{2,2} \cup (R^0_{2,1} \circ (R^0_{1,1})^* \circ R^0_{1,2}) \\
R^1_{2,3} &=& R^0_{2,3} \cup (R^0_{2,1} \circ (R^0_{1,1})^* \circ R^0_{1,3}) \\
R^1_{3,2} &=& R^0_{3,2} \cup (R^0_{3,1} \circ (R^0_{1,1})^* \circ R^0_{1,2}) \\
R^1_{3,3} &=& R^0_{3,3} \cup (R^0_{3,1} \circ (R^0_{1,1})^* \circ R^0_{1,3})
\end{array}
$$

  Nun können die relevanten Mengen, ausgehend von den Grundmengen, berechnet werden:

$$
\begin{array}{rclcl}
R^0_{1,1} &=& a \cup \epsilon \\
R^0_{1,2} &=& c \\
R^0_{1,3} &=& a \\
R^0_{2,1} &=& \emptyset \\
R^0_{2,2} &=& a \cup c \cup \epsilon \\
R^0_{2,3} &=& b \\
R^0_{3,1} &=& \emptyset \\
R^0_{3,2} &=& \emptyset \\
R^0_{3,3} &=& c \cup \epsilon \\[1ex]
R^1_{1,2} &=& c \cup ((a \cup \epsilon)(a \cup \epsilon)^* c) &=& a^* c \\
R^1_{1,3} &=& a \cup ((a \cup \epsilon)(a \cup \epsilon)^* a) &=& a^* a \\
R^1_{2,2} &=& (a \cup c \cup \epsilon) \cup (\emptyset(a \cup \epsilon)^* c) &=& (a \cup c \cup \epsilon) \\
R^1_{2,3} &=& b \cup (\emptyset(a \cup \epsilon)^* a) &=& b \\
R^1_{3,2} &=& \emptyset \cup (\emptyset(a \cup \epsilon)^* c) &=& \emptyset \\
R^1_{3,3} &=& (c \cup \epsilon) \cup (\emptyset(a \cup \epsilon)^* a) &=& (c \cup \epsilon) \\[1ex]
R^2_{1,3} &=& a^* a \cup (a^* c(a \cup c \cup \epsilon)^* b) &=& a^* a \cup (a^* c(a \cup c)^* b) \\
R^2_{3,3} &=& (c \cup \epsilon) \cup (\emptyset(a \cup c \cup \epsilon)^* b) &=& (c \cup \epsilon)
\end{array}
$$

  Es ergibt sich somit für die Lösungsmenge $R^3_{1,3}$:

$$
\begin{array}{rcl}
R^3_{1,3} &=& (a^* a \cup (a^* c(a \cup c)^* b)) \cup ((a^* a \cup (a^* c(a \cup c)^* b))(c \cup \epsilon)^*(c \cup \epsilon)) \\
&=& (a^* a \cup (a^* c(a \cup c)^* b))c^*
\end{array}
$$

  Durch Ausklammern von $a^*$ (und das Weglassen unnötiger Klammern) erhält man den regulären Ausdruck $\gamma = a^*(a \cup c(a \cup c)^* b)c^*$.

**Aufgabe 7** (10 Punkte)

Gegeben sei das folgende LOOP$_2$-Programm $P$:

$$x_2 := 1;\ \textbf{loop}\ x_1\ \textbf{do}\ x_2 := 0\ \textbf{end};\ x_1 := x_2$$

a) [2] Welche aus der Vorlesung bekannte Funktion berechnet $P$?

b) [8] Es sei $\beta_1$ die Belegung $\{x_1 := 2\}$ und $\beta_2$ die Belegung $\{x_1 := 0\}$. Geben Sie jeweils die Konfigurationsfolge an, welche die Anfangskonfiguration

$$\begin{pmatrix} \beta_i \\ P; \square \end{pmatrix}$$

in eine Endkonfiguration

$$\begin{pmatrix} \beta_i' \\ \square \end{pmatrix}$$

überführt. Verwenden Sie dazu die folgenden vereinfachten Übergangsregeln (also *nicht* diejenigen aus Vorlesung/Skript):

- $\begin{pmatrix} \beta \\ x_i := n; P \end{pmatrix} \Longrightarrow \begin{pmatrix} \beta[x_i := n] \\ P \end{pmatrix}$ für jede natürliche Zahl $n$
- $\begin{pmatrix} \beta \\ x_i := x_j; P \end{pmatrix} \Longrightarrow \begin{pmatrix} \beta[x_i := \beta(x_j)] \\ P \end{pmatrix}$
- $\begin{pmatrix} \beta \\ \textbf{loop}\ x_i\ \textbf{do}\ Q\ \textbf{end}; P \end{pmatrix} \Longrightarrow \begin{pmatrix} \beta \\ \underbrace{Q; \ldots; Q}_{\beta(x_i)-\text{mal}}; P \end{pmatrix}$

**Lösung:**

a) Das Programm berechnet das Cosignum $\overline{sg}$ der Eingabe.

b) Die Konfigurationsfolge für die Belegung $\beta_1$ lautet:

$$\begin{aligned}
& \begin{pmatrix} \{x_1 := 2\} \\ x_2 := 1;\ \textbf{loop}\ x_1\ \textbf{do}\ x_2 := 0\ \textbf{end};\ x_1 := x_2;\ \square \end{pmatrix} \\
\Longrightarrow\ & \begin{pmatrix} \{x_1 := 2; x_2 := 1\} \\ \textbf{loop}\ x_1\ \textbf{do}\ x_2 := 0\ \textbf{end};\ x_1 := x_2;\ \square \end{pmatrix} \\
\Longrightarrow\ & \begin{pmatrix} \{x_1 := 2; x_2 := 1\} \\ x_2 := 0;\ x_2 := 0;\ x_1 := x_2;\ \square \end{pmatrix} \\
\Longrightarrow\ & \begin{pmatrix} \{x_1 := 2; x_2 := 0\} \\ x_2 := 0;\ x_1 := x_2;\ \square \end{pmatrix} \\
\Longrightarrow\ & \begin{pmatrix} \{x_1 := 2; x_2 := 0\} \\ x_1 := x_2;\ \square \end{pmatrix} \\
\Longrightarrow\ & \begin{pmatrix} \{x_1 := 0; x_2 := 0\} \\ \square \end{pmatrix}
\end{aligned}$$

Die Konfigurationsfolge für die Belegung $\beta_2$ lautet:

$$
\begin{aligned}
& \left( \begin{array}{c} \{x_1 := 0\} \\ x_2 := 1;\ \textbf{loop}\ x_1\ \textbf{do}\ x_2 := 0\ \textbf{end};\ x_1 := x_2;\ \square \end{array} \right) \\
\Longrightarrow\ & \left( \begin{array}{c} \{x_1 := 0; x_2 := 1\} \\ \textbf{loop}\ x_1\ \textbf{do}\ x_2 := 0\ \textbf{end};\ x_1 := x_2;\ \square \end{array} \right) \\
\Longrightarrow\ & \left( \begin{array}{c} \{x_1 := 0; x_2 := 1\} \\ x_1 := x_2;\ \square \end{array} \right) \\
\Longrightarrow\ & \left( \begin{array}{c} \{x_1 := 1; x_2 := 1\} \\ \square \end{array} \right)
\end{aligned}
$$