Klausur Informatik III

2. März 2004

Musterlösung

Aufgabe 1 (6 Punkte)

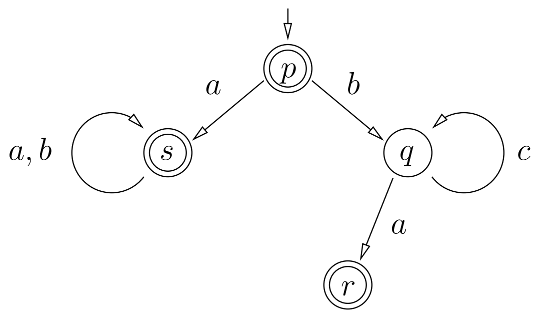
Geben Sie zu dem regulären Ausdruck

$$\gamma = (ab^*)^* \cup bc^*a$$

einen reduzierten deterministischen endlichen Automaten (DEA) $\mathcal{A}$ über dem Alphabet $\{a, b, c\}$ an mit $L(\mathcal{A}) = \langle \gamma \rangle$.

Lösung:

[6] Der DEA $\mathcal{A} = \langle \{p, q, r, s\}, \{a, b, c\}, \delta, p, \{p, r, s\} \rangle$ mit folgendem Übergangsgraphen leistet das gewünschte und ist bereits reduziert:



Aufgabe 2 (8 Punkte)

Gegeben sei die Grammatik $\Gamma = \langle\{S,T,U,V,W,X,A,B,C\},\{a,b,c\},\Pi,S\rangle$, wobei Π durch folgende Produktionen erklärt ist:

$$
\begin{array}{lcl@{\qquad}lcl@{\qquad}lcl}
S & \longrightarrow & AT & V & \longrightarrow & BW \mid CX \mid CA & A & \longrightarrow & a \\
T & \longrightarrow & AU & W & \longrightarrow & CX \mid CA & B & \longrightarrow & b \\
U & \longrightarrow & AV & X & \longrightarrow & AV & C & \longrightarrow & c
\end{array}
$$

a) Entscheiden Sie mithilfe des Algorithmus von Coke, Younger und Kasami (CYK), ob die folgenden Wörter in $L(\Gamma)$ enthalten sind:

1. *aacabca*
2. *aaabcaca*

b) Ist die Sprache $L(\Gamma)$ regulär?

Lösung:

a) 1. [3] $aacabca \notin L(\Gamma)$:

$$
\begin{array}{cccccccc}
 & A & - & - & T & - & - & T \\
a & & A & - & U,X & - & - & U,X \\
 & a & & C & V,W & - & - & V,W \\
 & & c & & A & - & - & U,X \\
 & & & a & & B & - & V \\
 & & & & b & & C & V,W \\
 & & & & & c & & A \\
 & & & & & & a &
\end{array}
$$

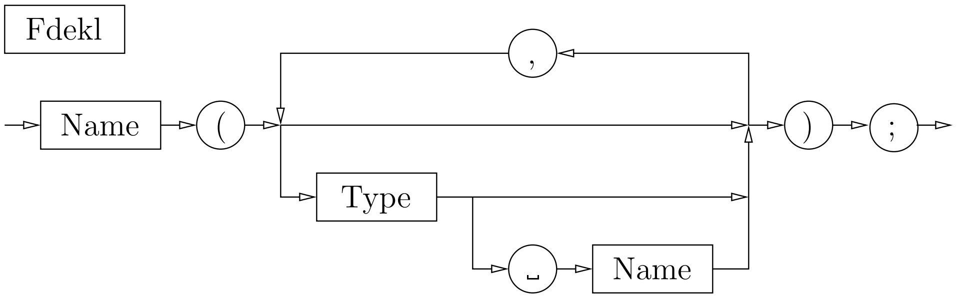
2. [3] $aaabcaca \in L(\Gamma)$:

$$
\begin{array}{ccccccccc}
 & A & - & - & - & - & S & - & S \\
a & & A & - & - & - & T & - & T \\
 & a & & A & - & - & U,X & - & U,X \\
 & & a & & B & - & V & - & V \\
 & & & b & & C & V,W & - & V,W \\
 & & & & c & & A & - & U,X \\
 & & & & & a & & C & V,W \\
 & & & & & & c & & A \\
 & & & & & & & a &
\end{array}
$$

b) [2] $L(\Gamma)$ ist regulär. Ersetze in Γ bei allen Produktionen, auf deren rechter Seite zwei Variablen stehen, die linke der beiden Variablen (immer A, B oder C) durch a, b oder c. In der resultierenden regulären Grammatik sind alle Wörter ableitbar, die in Γ ableitbar sind.

Aufgabe 3 (6 Punkte)

Syntaxdiagramme sind eine alternative Form der Darstellung kontextfreier Grammatiken, bei der die rechte Seite einer Produktion durch einen Graphen dargestellt wird, der sogar Schleifen enthalten kann. Das folgende Beispiel ist die Produktion, die zu der Variable "Fdekl" gehört, wobei die Kästchen mit den Bezeichnungen "Name" und "Type" als Variablen auf weitere Syntaxdiagramme verweisen, während die Kreise Terminale enthalten.



Durch dieses Syntaxdiagramm lassen sich z.B. folgende Wörter erzeugen:

Name ( ) ;
Name (Type) ;
Name (Type ␣ Name) ;
⋮
Name (Type, Type ␣ Name) ;
⋮

Geben Sie eine Grammatik $\Gamma = \langle \mathcal{V}, \mathcal{T}, \Pi, \mathrm{Fdekl} \rangle$ an, wobei $\mathcal{V} \supseteq \{\mathrm{Fdekl}, \mathrm{Name}, \mathrm{Type}\}$, $\mathcal{T} = \{(,), ;, \textvisiblespace, ,\}$, so daß in Γ alle Wörter ableitbar sind, die durch das Syntaxdiagramm erzeugt werden können.

Lösung:

[6] Idee: Füge an dem Knoten unmittelbar nach "(" die Variable A und an dem Knoten unmittelbar vor ")" die Variable B ein. Dann läßt sich das Syntaxdiagramm wie folgt linearisieren.

Es sei $\mathcal{V} = \{\mathrm{Fdekl}, \mathrm{Name}, \mathrm{Type}, A, B\}$. Π enthält folgende Produktionen:

$$\begin{array}{ccl} \mathrm{Fdekl} & \longrightarrow & \mathrm{Name}\,(\,A \\ A & \longrightarrow & B \mid \mathrm{Type}\,B \mid \mathrm{Type}\textvisiblespace\mathrm{Name}\,B \\ B & \longrightarrow &)\,; \mid\; ,\,A \end{array}$$

Aufgabe 4 (6 Punkte)

Gegeben sei die kontextfreie Grammatik $\Gamma = \langle \{S, T\}, \{0, 1\}, \Pi, S\rangle$, wobei Π durch folgende Produktionen erklärt ist:

$$\begin{array}{rcl} S & \longrightarrow & 0S11 \mid T \\ T & \longrightarrow & 0T0 \mid 1T1 \mid 00 \end{array}$$

a) Konstruieren Sie einen zu Γ äquivalenten Kellerautomaten $\mathcal{A}$ mit $N(\mathcal{A}) = L(\Gamma)$.

b) Konstruieren Sie einen Kellerautomaten $\mathcal{A}'$ mit $L(\mathcal{A}') = N(\mathcal{A})$.

Lösung:

a) [4] Für den Automaten $\mathcal{A} = \langle \{q\}, \{0, 1\}, \{S, T, 0, 1\}, \delta, q, S, \emptyset\rangle$ mit

$$\begin{array}{rcl} \delta(q, \epsilon, S) & = & \{(q, 0S11), (q, T)\} \\ \delta(q, \epsilon, T) & = & \{(q, 0T0), (q, 1T1), (q, 00)\} \\ \delta(q, 0, 0) & = & \{(q, \epsilon)\} \\ \delta(q, 1, 1) & = & \{(q, \epsilon)\} \end{array}$$

gilt nach Theorem 6.1 $N(\mathcal{A}) = L(\Gamma)$.

b) [2] Für den Automaten $\mathcal{A}' = \langle \{q_s, q, q_f\}, \{0, 1\}, \{Z, S, T, 0, 1\}, \delta \cup \delta', q_s, Z, \{q_f\}\rangle$ mit

$$\begin{array}{rcl} \delta'(q_s, \epsilon, Z) & = & \{(q, SZ)\} \\ \delta'(q, \epsilon, Z) & = & \{(q_f, Z)\} \end{array}$$

gilt nach Satz 6.2 $L(\mathcal{A}') = N(\mathcal{A})$.

Aufgabe 5 (8 Punkte)

Beantworten Sie folgende Fragen (mit Begründung):

a) Sind alle Sprachen über einem einelementigen Alphabet regulär?

b) Ist das Komplement einer kontextfreien Sprache selbst kontextfrei?

c) In welcher Sprachklasse liegt der Schnitt einer kontextfreien Sprache mit einer regulären Sprache?

d) Kann der Schnitt zweier kontextfreier Sprachen regulär sein?

e) Besitzt jedes Wort einer durch eine kontextfreie Grammatik Γ gegebenen Sprache genau eine Linksableitung?

Lösung:

a) [2] Nein. Gegenbeispiel: $L = \{a^p \mid p \text{ ist Primzahl}\}$.

b) [1] Nein. Die kontextfreien Sprachen sind unter Vereinigung abgeschlossen, nicht aber unter Schnitt. Weil aber $L_1 \cap L_2 = \Sigma^* \setminus ((\Sigma^* \setminus L_1) \cup (\Sigma^* \setminus L_2))$, können die kontextfreien Sprachen nicht unter Komplement abgeschlossen sein (Lemma 7.2).

c) [1] Der Schnitt einer kontextfreien Sprache und einer regulären Sprache ist kontextfrei (Satz 7.1 - Kreuzproduktautomat aus DEA und KA).

d) [2] Ja. Beispiel: Seien $L_1 = \{a^n b^m c^m \mid m, n \geq 0\}$ und $L_2 = \{a^n c^m b^m \mid m, n \geq 0\}$. Dann ist $L_1 \cap L_2 = \{a^n \mid n \geq 0\} = \langle a^* \rangle$, also regulär.

e) [2] Nein. Gegenbeispiel: Sei $\Gamma = \langle \{S, A, B\}, \{a\}, \Pi, S \rangle$ mit den Produktionen

$$\begin{array}{lcl} S & \longrightarrow & A \mid B \\ A & \longrightarrow & a \\ B & \longrightarrow & a \end{array}$$

Dann sind sowohl $S \Longrightarrow A \Longrightarrow a$ als auch $S \Longrightarrow B \Longrightarrow a$ Linksableitungen von a.

Aufgabe 6 (6 Punkte)

Geben Sie eine Turingmaschine $\mathcal{M}$ über dem Alphabet $\{0, 1\}$ an, welche für die Darstellung natürlicher Zahlen als Binärstrings die Funktion

$$f(x) = 2x + 1$$

berechnet. Das heißt, wenn $[n]_b$ den Binärstring bezeichnet, der zur Zahl n gehört, und dessen niedrigstwertiges Bit ganz rechts steht, dann soll $\mathcal{M}$ für jedes n den folgenden Übergang realisieren:

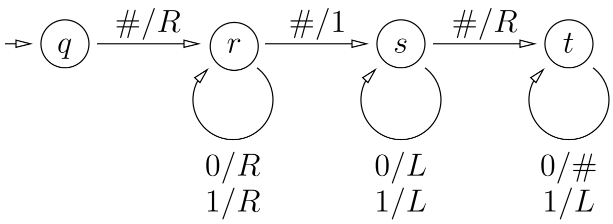
$$(\underline{\#}[n]_b) \underset{\mathcal{M}}{\Rightarrow} (\underline{\#}[2n + 1]_b)$$

Schreiben Sie $\mathcal{M}$ entweder als Maschinentafel oder Übergangsgraphen auf, also *nicht* als Maschinenschema.

Lösung:

[6] Idee: Die Multiplikation $2x$ wird auf Binärstrings durch ein Verschieben des Strings um eine Stelle nach links bei nachrückender Null realisiert. Eine nachrückende Eins resultiert demzufolge in dem Binärstring, der dem Wert $2x + 1$ entspricht.

Die Maschine $\mathcal{M} = \langle \{q, r, s, t\}, \{0, 1, \#\}, \delta, q\rangle$ mit folgendem δ leistet das gewünschte:



$\mathcal{M}$ läuft auf den String (Zustand q), dann bis einen Schritt über das rechte Ende des Binärstrings hinaus (Zustand r), schreibt dort eine Eins und läuft zurück auf das erste Blank links vom Wortanfang (Zustand s). Zuletzt sollte noch eine evtl. führende Null entfernt werden (Zustand t). Zwar kann man davon ausgehen, daß die Eingabe ohne führende Nullen auf dem Band steht, aber auf diese Weise ist auch folgender Sonderfall korrekt behandelt:

$$(\underline{\#}0) \underset{\mathcal{M}}{\Rightarrow} (\underline{\#}1)$$

Aufgabe 7 (6 Punkte)

Schreiben Sie ein LOOP-Programm, das folgende Funktion berechnet:

$$\arg(x,y) = \begin{cases} 0 & \text{falls } x = y \\ 1 & \text{falls } x > y \\ 2 & \text{falls } x < y \end{cases}$$

Sie dürfen dabei Zuweisungen der Form $x_i := x_j \dot{-} x_k$ verwenden, ansonsten nur Grundoperationen.

Lösung:

[6] Folgendes LOOP-Programm leistet das gewünschte:

$x_3 := x_1 \dot{-} x_2;$	es ist $x_3 > 0$, falls $x_1 > x_2$
$x_4 := x_2 \dot{-} x_1;$	es ist $x_4 > 0$, falls $x_1 < x_2$
$x_1 := 0;$	initialisiere die Ausgabe mit 0
loop x_3 **do**	es wird x_3-mal wiederholt:
$x_1 := 0;$	Ausgabe mit 0 initialisieren
$x_1 := N(x_1)$	Ausgabe um Eins erhöhen
end	
loop x_4 **do**	es wird x_4-mal wiederholt:
$x_1 := 0;$	Ausgabe mit 0 initialisieren
$x_1 := N(x_1);$	Ausgabe um Eins erhöhen
$x_1 := N(x_1)$	Ausgabe um Eins erhöhen
end	

Aufgabe 8 (6 Punkte)

Ist die Funktion f, die den ganzzahligen Anteil der Wurzel einer natürlichen Zahl x berechnet, d.h. $f(x) = \lfloor\sqrt{x}\rfloor$, primitiv rekursiv definierbar?

Falls ja, geben Sie eine entsprechende Definition an.

Falls nicht, geben Sie eine partiell rekursive Funktionsdefinition für f an.

Sie dürfen auf Funktionen zurückgreifen, die in der Vorlesung als primitiv rekursiv bzw. partiell rekursiv nachgewiesen wurden.

Lösung:

[6] Zu einer gegebenen Zahl x suchen wir die kleinste Zahl y mit $(y+1)^2 > x$. Damit ergibt sich unmittelbar folgende partiell rekursive Definition:

$$f(x) = (\mu y)\ ((y+1)\cdot(y+1) > x)$$

Weiterhin ist für natürliche Zahlen x stets $0 \leq \lfloor\sqrt{x}\rfloor \leq x$, d.h. x ist eine obere Schranke für $\lfloor\sqrt{x}\rfloor$. Insgesamt kann man f demnach durch folgende primitiv rekursive Definition beschreiben:

$$f(x) = (\mu y \leq x)\ ((y+1)\cdot(y+1) > x)$$

Aufgabe 9 (8 Punkte)

Betrachten Sie folgende Eigenschaften von Mengen:

1. Turing-entscheidbar
2. Turing-aufzählbar
3. partiell rekursiv
4. partiell rekursiv aufzählbar
5. rekursiv
6. rekursiv aufzählbar
7. primitiv rekursiv
8. primitiv rekursiv aufzählbar
9. LOOP-entscheidbar*
10. WHILE-entscheidbar*
11. grammatisch

a) Welche dieser Eigenschaften sind äquivalent, welche nicht? Geben Sie die Antwort dadurch, daß Sie Äquivalenzklassen auf der Menge $\{1, 2, 3, \ldots, 11\}$ angeben.

b) Ordnen Sie die in a) gewonnenen Äquivalenzklassen nach der Stärke der Eigenschaften, die sie beschreiben.

Lösung:

Folgende Begriffe sind jeweils äquivalent und von schwach nach stark angeordnet:

- [4] Turing-aufzählbar, partiell rekursiv aufzählbar, rekursiv aufzählbar, primitiv rekursiv aufzählbar und grammatisch (Theoreme 11.1, 15.1 und 15.2)
- [2] Turing-entscheidbar, partiell rekursiv und WHILE-entscheidbar (Theoreme 13.1, 13.3 und 14.14)
- [1] rekursiv heißt total und partiell rekursiv (siehe auch Satz 12.7, Lemma 12.8)
- [1] primitiv rekursiv und LOOP-entscheidbar (Theorem 14.6)

Insgesamt kann man die Äquivalenzklassen also wie folgt anordnen:

$$\{7, 9\} \succ \{5\} \succ \{1, 3, 10\} \succ \{2, 4, 6, 8, 11\}$$

*Eine Menge heißt LOOP- bzw. WHILE-*entscheidbar*, falls deren charakteristische Funktion LOOP- bzw. WHILE-*berechenbar* ist.