Universität Karlsruhe (TH)

Institut für Programmstrukturen und Datenorganisation (IPD)

Informatik I WS 2003/04 http://www.infoeins.de

Dozent: Prof. Dr.rer.nat. G. Goos goos@ipd.info.uni-karlsruhe.de

Übungsleiter: Tom Gelhausen gelhausen@fzi.de

Musterlösung 2 - (60T / 0P)

Semi-Thue-Systeme, Markov, Chomsky

Ausgabe: 24.10.2003

Abgabe: 31.10.2003 13:30 Uhr

Einwurf im Keller des Informatik-Hauptbaus (Geb. 50.34)

Aufgabe 1: Semi-Thue-Systeme und Markov-Algorithmen (24T)

1.1 Produkt-Vorzeichen (2T)

Entwerfen Sie ein Semi-Thue-System, das als Eingabe die Vorzeichen einer Multiplikationskette erhält und als Ausgabe das Vorzeichen des Produktes liefert.
(Beispiel: „(-3) · (+4) · (-3)“ bedeutet „- + -“ als Eingabe, die Ausgabe soll in diesem Falle „+“ sein)

Lösung:

Semi-Thue-System (Verwendung der richtigen Pfeile beachten: „↦“ für Regeln, „⇒“ für Ableitungen!)

$$
\begin{aligned}
++ &\mapsto + \\
-- &\mapsto + \\
+- &\mapsto - \\
-+ &\mapsto -
\end{aligned}
$$

1.2 Einerkomplement (5T)

Entwerfen Sie nun ein Semi-Thue-System, das zu einer gegebenen Bitfolge das Einerkomplement bestimmt. (Hinweis: Sie dürfen sich zur Lösung dieser Aufgabe Hilfsmarkierungen in das Eingabewort einbauen.) (3T)

Kann diese Aufgabe ohne Hilfsmarkierungen gelöst werden (Begründung)? (1T)

Führen Sie den Algorithmus für die Bitfolge „101010“ vor. (Fügen Sie ggf. benötigte Hilfsmarkierungen ein, um diese Bitfolge in Ihr Eingabewort zu verwandeln.) (1T)

Lösung:

Semi-Thue-System (Verwendung der richtigen Pfeile beachten: „↦“ für Regeln, „⇒“ für Ableitungen!)

$$
\begin{aligned}
\alpha 0 &\mapsto 1\alpha \\
\alpha 1 &\mapsto 0\alpha \\
\alpha\alpha &\mapsto \varepsilon
\end{aligned}
$$

Die Aufgabe kann nur mit Schiffchen gelöst werden, wegen Eigenschaften des Semi-Thue-Systems:

- Es wird eine beliebige Regel angewandt
- Die Regel wird an einer beliebigen, passenden Stelle angewandt

Beispieldurchführung:
- Bitfolge „101010“ wird zu Eingabewort „α101010α“
- α101010α ⇒ 0α01010α ⇒ 01α1010α ⇒ 010α010α ⇒ 0101α10α ⇒ 01010α0α ⇒ 010101αα ⇒ 010101

1.3 Subtrahieren (4T)

Formulieren Sie einen Markov-Algorithmus zum Subtrahieren von Strichfolgen (z.B. ||| - |||| = -|). Beschreiben Sie auch jeweils kurz, was die jeweilige Regel bewirkt.

Lösung:
Markov-Algorithmus (die Beschreibungen **müssen** Aussagekräftig sein, sonst nur 0,5T pro Regel!):

1. |-| ↦ - „ziehe eins ab“ (1T)
2. |- ↦. | „nichts mehr abzuziehen (HALT)“ (1T)
3. -| ↦. -| „nichts mehr da, um davon abzuziehen (HALT, negatives Ergebnis)“ (1T)
4. - ↦. ε „Ergebnis 0 (HALT)“ (1T)

1.4 Vereinigung von Mengen (6T)

Formulieren Sie ein Semi-Thue-System zur Vereinigung von Mengen, mit Σ={„a“, „b“, „c“ und „∪“}. Jeder Buchstabe darf in der Ergebnismenge nur einmal vorkommen (z.B. ab ∪ ac = abc). (3T)

Beschreiben Sie auch jeweils kurz, was die jeweilige Regel bewirkt und motivieren Sie, wie/warum Ihr Semi-Thue-System funktioniert. (3T)

Lösung:

∪	↦	ε	„Vereinigungssymbol(e) eliminieren“
ba	↦	ab	„Sortieren: a und b in die richtige Reihenfolge bringen“
ca	↦	ac	„Sortieren: a und c in die richtige Reihenfolge bringen“
cb	↦	bc	„Sortieren: b und c in die richtige Reihenfolge bringen“
aa	↦	a	„Mengengesetze: Duplikate von a rauswerfen“
bb	↦	b	„Mengengesetze: Duplikate von b rauswerfen“
cc	↦	c	„Mengengesetze: Duplikate von c rauswerfen“

Die Zeichenmengen sind auf jeden Fall irgendwann vereinigt (Regel 1) und sortiert (Regeln 2-4). Dann ist keine der der ersten vier Regeln mehr anwendbar und die Regeln 5-6 werfen alle Duplikate raus. Ob bereits zuvor Duplikate rausgeworfen wurden, oder nicht ist dabei irrelevant.

1.5 Durchschnitt von Mengen (7T)

Formulieren Sie einen Markov-Algorithmus zum Schneiden zweier Mengen, mit Σ = {„a“, „b“, „c“ und „∩“}. Jeder Buchstabe darf in der Ergebnismenge nur einmal vorkommen (z.B. ab ∩ ac = a). (4T)

Beschreiben Sie auch jeweils kurz, was die jeweilige Regel bewirkt. (3T)

Lösung:

1.	∩	↦	ε	„Durchschnittsymbol eliminieren“
2.	ba	↦	ab	„Sortieren: a und b in die richtige Reihenfolge bringen“
3.	ca	↦	ac	„Sortieren: a und c in die richtige Reihenfolge bringen“
4.	cb	↦	bc	„Sortieren: b und c in die richtige Reihenfolge bringen“
5.	αaa	↦	aα	„Mehrfachvorkommen von a durch Ergebniselement ersetzen“
6.	αbb	↦	bα	„Mehrfachvorkommen von b durch Ergebniselement ersetzen“
7.	αcc	↦	cα	„Mehrfachvorkommen von c durch Ergebniselement ersetzen“
8.	αa	↦	α	„Einzelvorkommen von a eliminieren“
9.	αb	↦	α	„Einzelvorkommen von b eliminieren“
10.	αc	↦	α	„Einzelvorkommen von c eliminieren“
11.	α	↦.	ε	„Halteregel (Schiffchen versenken)“
12.	ε	↦	α	„Schiffchen erzeugen“

2. Kaffeedosen-Spiel (14T)

2.1 Anwendung (2T)

Schauen Sie sich Scholtens Kaffeedosen-Spiel aus der Vorlesung noch einmal an und führen Sie dann folgende Spiele durch (Sie können annehmen, einen genügend großen Vorrat an schwarzen Bohnen außerhalb der Kaffeedose zur Verfügung zu haben.) Machen Sie dabei deutlich, welche Regel Sie an welcher Stelle anwenden:

a. [Schwarz, Weiß, Weiß, Schwarz, Schwarz, Schwarz, Schwarz]
b. [Weiß, Schwarz, Schwarz, Weiß, Schwarz, Weiß, Schwarz]

Lösung:

a. [<u>Schwarz, Weiß</u>, Weiß, Schwarz, Schwarz, Schwarz, Schwarz]

⇒ [<u>**Weiß**, Weiß</u>, Schwarz, Schwarz, Schwarz, Schwarz]

⇒ [**Schwarz**, Schwarz, <u>Schwarz, Schwarz</u>, Schwarz]

⇒ [Schwarz, Schwarz, <u>**Schwarz**, Schwarz</u>]

⇒ [<u>Schwarz, Schwarz</u>, **Schwarz**] ⇒ [<u>**Schwarz**, Schwarz</u>] ⇒ [**Schwarz**]

b. [Weiß, Schwarz, <u>Schwarz, Weiß</u>, Schwarz, Weiß, Schwarz]

⇒ [<u>Weiß, Schwarz</u>, **Weiß**, Schwarz, Weiß, Schwarz]

⇒ [<u>**Weiß**, Weiß</u>, Schwarz, Weiß, Schwarz]

⇒ [<u>**Schwarz**, Schwarz</u>, Weiß, Schwarz]

⇒ [<u>**Schwarz**, Weiß</u>, Schwarz] ⇒ [<u>**Weiß**, Schwarz</u>] ⇒ [**Weiß**]

2.2 Weiterführende Überlegungen (12T)

Beweisen Sie den Satz aus der Vorlesung bezüglich Scholtens Kaffeedosen-Spiel:

> *„Das Ergebnis (Farbe der letzten Bohne) ist unabhängig von der Reihenfolge der Anordnung und der Regelauswahl"*

Hinweis: Beweisen Sie zunächst die Eigenschaften einer einzelnen Ableitung und danach die Gültigkeit dieser Eigenschaften für Ableitungsketten beliebiger Länge!

Lösung:
Bei ungerader Anzahl weißer Bohnen ist das Ergebnis eine weiße Bohne, ansonsten eine schwarze Bohne. Die Anzahl an schwarzen Bohnen ist unerheblich. Außerdem ist das Ergebnis unabhängig von der Reihenfolge der Bohnen. Sei s die Anzahl der schwarzen Bohnen und w die Anzahl der weißen Bohnen, „mod" bezeichne die Modulo-Operation.

Zur Erinnerung:

Regel 1:	schwarz schwarz	$\mapsto$	schwarz
Regel 2:	weiß weiß	$\mapsto$	schwarz
Regel 3:	schwarz weiß	$\mapsto$	weiß
Regel 4:	weiß schwarz	$\mapsto$	weiß

Lemma (1T): Für jede einzelne Regelanwendung mit $(w,s) \Rightarrow (w', s')$ gilt:

- $w' + s'+1 = w + s$
- $w' \bmod 2 = w \bmod 2$

Beweis (4T):

- Regel 1 $\Rightarrow s' = s - 1, w' = w$. Somit ist $w' + s' + 1 = w' + s - 1 + 1 = w + s$ und $w' \bmod 2 = w \bmod 2$.
- Regel 2 $\Rightarrow s' = s + 1, w' = w - 2$
  Somit ist $w'+s'+1 = w-2+s+1+1 = w+s$ und $(w' \bmod 2) = (w-2 \bmod 2) = (w \bmod 2)$.
- Regel 3 $\Rightarrow s' = s - 1, w' = w$. Somit ist $w' + s' + 1 = w' + s - 1 + 1 = w + s$ und $w' \bmod 2 = w \bmod 2$.
- Regel 4 $\Rightarrow s' = s - 1, w' = w$. Somit ist $w' + s' + 1 = w' + s - 1 + 1 = w + s$ und $w' \bmod 2 = w \bmod 2$.

Satz (1T): Für eine Reihe von k beliebigen Regelanwendungen $(w,s) \Rightarrow^k (w',s')$ gilt:

- $w' + s' + k = w + s$
- $w' \bmod 2 = w \bmod 2$

Beweis: Vollständige Induktion

Induktionsanfang (IA) k=1 (1T):	genau eine Regelanwendung → siehe Lemma.
Induktionsvoraussetzung (IV) (1T):	die Behauptumg gelte für k Anwendungen
Induktionsschritt (IS) k→k + 1 (1T):	
Es gilt:	$(w,s) \Rightarrow^{k+1} (w',s') \equiv (w,s) \Rightarrow^k (w'',s'') \Rightarrow (w',s')$
Für bel. Abl.-ketten der Länge k gilt (IV) (1T):	$w'' + s'' + k = w + s$ und $w'' \bmod 2 = w \bmod 2$.
Nach Lemma gilt (1T):	$w' + s' + 1 = w'' + s''$ und $w'' \bmod 2 = w' \bmod 2$.
Zusammengenommen (1T) gilt:	$w+s = w''+s''+k = ((w'+s'+1)+k) = (w'+s')+(k+1)$
Und	$(w \bmod 2) = (w'' \bmod 2) = (w' \bmod 2)$

Hinweis an Tutoren:
Es gab den Hinweis in der Vorlesung, dass diese Aufgabe mit Vollständiger Induktion zu lösen sei. Auch eine grobe Beweis-Skizze wurde angegeben. Es muss daher ein formaler Beweis wie hier geführt werden. Ein-Satz-Antworten reichen definitiv nicht aus!

3. Chomsky-Grammatiken (22T)

3.1 Grammatik 1 (5T)

Gegeben sei die folgende Grammatik:

$G_1 = \{ \Sigma, N, P, S \}$
$N = \{ S, I, N \}$
$\Sigma = \{ n, i \}$
$P = \{ S \mapsto N,$
$I \mapsto iN \mid i,$
$N \mapsto nI \}$

Geben Sie den einschränkensten Chomsky Typ an (1T) und begründen Sie ihre Entscheidung (2T). Geben Sie außerdem die erzeugte Sprache an (2T).

Lösung:
Die Grammatik ist vom Typ Chomsky 2.
Auf den linken Seiten stehen nur einzelne Nichtterminale, was auf Typ 2 oder Typ 3 schließen lässt. Allerdings kommt auf der rechten Seite kommt eine sog. *Kettenproduktion* vor, d.h. ein Nichtterminal wird in ein anderes überführt. Dies ist nur in Grammatiken vom Typ 2 erlaubt.

$\mathcal{L}(G_1) = \{ (ni)^k \mid k > 0 \}$

3.2 Grammatik 2 (5T)

Gegeben sei die folgende Grammatik:

$G_2 = \{ \Sigma, N, P, S \}$
$N = \{ S, A, B \}$
$\Sigma = \{ 0, 1 \}$
$P = \{ S \mapsto 0B \mid 1A,$
$A \mapsto 0 \mid 0S,$
$B \mapsto 1 \mid 1S \}$

Geben Sie den einschränkensten Chomsky Typ an (1T) und begründen Sie ihre Entscheidung (2T). Geben Sie außerdem die erzeugte Sprache an (2T).

Lösung:
Die Grammatik ist vom Typ Chomsky 3.
Alle Produktionen sind terminierend bzw. rechtlinear, damit handelt es sich um eine *reguläre Grammatik*, also um eine vom Typ 3.

$\mathcal{L}(G_2) = \{ x \mid x \in T^+, T = \{01, 10\} \}$

3.3 Klassifikation von Grammatiken (8T)

Geben Sie zu den folgenden Grammatiken den einschränkensten Chomsky Typ an (mit Begründung).

a. $G_a = \{ \Sigma, N, P, S \}$ (2T)

$N = \{ S, A, B \}$

$\Sigma = \{ 0, 1 \}$

$P = \{ S \mapsto 0SA \mid 0B,$

$BA \mapsto 1B0,$

$0A \mapsto A1,$

$B \mapsto 10 \}$

b. $G_b = \{ \Sigma, N, P, S \}$ (2T)

$N = \{ S \}$

$\Sigma = \{ 0, 1 \}$

$P = \{ S \mapsto 01 \mid 0S1 \}$

c. $G_c = \{ \Sigma, N, P, S \}$ (2T)

$N = \{ S, A, B \}$

$\Sigma = \{ 0, 1 \}$

$P = \{ S \mapsto 0B \mid 1A,$

$A \mapsto 0 \mid 0S \mid 1AA,$

$B \mapsto 1 \mid 1S \mid 0BB \}$

d. $G_d = \{ \Sigma, N, P, S \}$ (2T)

$N = \{ S, A, B, C, D \}$

$\Sigma = \{ 0, 1, 2, 3 \}$

$P = \{ S \mapsto ABCD,$

$CD \mapsto E,$

$E \mapsto D \mid \varepsilon,$

$A \mapsto 0 \mid 0A,$

$B \mapsto 012 \mid 0B12,$

$D \mapsto 3,$

$B \mapsto \varepsilon \}$

Lösung:

a. Die Grammatik ist vom Typ Chomsky 0, die Sprache vom Typ CH-1, $\mathcal{L}(G_a) = \{ 0^k1^k0^k \mid k > 0 \}$

b. Chomsky 2, $\mathcal{L}(G_b) = \{ 0^k1^k \mid k > 0 \}$

c. Chomsky 2, $\mathcal{L}(G_c) = \{ x \mid x \in \Sigma^*$, Anzahl 0 und 1 in x ist gleich $\}$

d. Chomsky 0, $\mathcal{L}(G_d) = \{ 0^{k+1}(12)^l 3^m \mid k > 0, l > 1, m \in \{0, 1\} \}$

3.4 Klassifikation von Produktionen (4T)

Von welchem Typ ist eine Grammatik höchstens, die folgende Produktionen enthält (je 1T):

a. caABa $\mapsto$ caAcAa

b. caABa $\mapsto$ caBAa

c. B $\mapsto$ cC

d. cAd $\mapsto$ da

Lösung:
a. Chomsky 1 (kontextsensitiv)
b. Die Grammatik ist höchstens vom Typ Chomsky 0 (beschränkt, aber nicht kontextsensitiv)
c. Chomsky 3 (linear)
d. Chomsky 0 (weder beschränkt noch kontextsensitiv)