



Universität Karlsruhe (TH)

Institut für Innovatives Rechnen und Programmstrukturen (IPD)

Informatik I WS 2002/03

Dozent: Prof. Dr.rer.nat. G. Goos

Übungsleiter: Markus L. Noga

<http://eins.info.uni-karlsruhe.de>

goos@ipd.info.uni-karlsruhe.de

noga@ipd.info.uni-karlsruhe.de

Übungsblatt 2

Ausgabe: 25.10.2002

Abgabe: 31.10.2002 13.00 Uhr

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

Aufgabe 1: Von Neumann Rechner (Harvard/Princeton) (10 T- Punkte)

1.1

Beschreiben Sie den den Befehlszyklus einer Von-Neumann-Rechnerarchitektur.

Lösung: 5 Pkt.

1. CPU holt aktuellen Befehl aus dem Speicher
2. CPU holt Daten für den Befehl aus dem Speicher
3. CPU führt den Befehl aus
4. CPU schreibt das Ergebnis des Befehls zurück in den Speicher
5. Weiter mit Schritt 1.

1.2

Legen Sie kurz und prägnant die Unterschiede zwischen der Harvard-und Princeton-Implementierung der Von-Neumann-Architektur dar.

Lösung: 5 Pkt.

Die Harvard-Implementierung besitzt zwei getrennte Speicher (Daten -und Befehlsspeicher) die mit je einem Bus an die CPU gebunden sind. In der Princeton-Implementierung dagegen werden Befehle und Daten im selben Speicher gespeichert. Und natürlich auch über denselben Bus übertragen.

Aufgabe 2: Semi-Thue-Systeme (10 P- Punkte)

2.1

Geben Sie ein Semi-Thue-System zur Verdopplung einer Strichfolge an. Hierbei sei die Strichfolge von zwei s's eingeschlossen. Bsp. s|||s

Im Ergebniswort dürfen die s's nicht mehr vorkommen.

Geben Sie den Ablauf des Algorithmus für die Strichfolge s|||s an.

Lösung: 6 Pkt.

Regeln:

$s \mid \rightarrow \mid \mid s$ (1)

$ss \rightarrow \varepsilon$. (2)

Eingabewort:s|||s

$s \mid \mid \mid s \rightarrow^{(1)} \mid \mid \mid \mid s \rightarrow^{(1)} \mid \mid \mid \mid \mid s \rightarrow^{(1)} \mid \mid \mid \mid \mid \mid s \rightarrow^{(1)} \mid \mid \mid \mid \mid \mid \mid s \rightarrow^{(1)} \mid \mid \mid \mid \mid \mid \mid \mid s \rightarrow^{(2)} \mid \mid \mid \mid \mid \mid \mid \mid \mid \mid$

2.2

Funktioniert die Strichfolgenverdopplung auch ohne die begrenzenden s'? Begründen Sie Ihre Meinung!

Lösung: 4 Pkt.

Die Strichfolgenverdopplung funktioniert nicht ohne die Begrenzung des Eingabewortes durch s's. Ohne die s's, würde das Semi-Thue-System nicht terminieren.

Ein s zu Beginn an den Anfang des Wortes zu setzen (vgl. Markov-Algorithmen) funktioniert hier auch nicht, da Regeln in beliebiger Stelle angewendet werden dürfen und ε zwischen allen Zeichen stehen kann. ($\varepsilon \rightarrow s$ funktioniert also nicht).

Lösung: Je Regel 1 Pkt. Rest 3 Pkt.

- | | |
|--|---|
| (1) $BA\alpha \rightarrow A\alpha B$ | (Ein B wird nach einer Folge von A's nach links sortiert) |
| (2) $\alpha BB \rightarrow B\alpha B$ | (α wird weitergeschoben, um den nächsten Fall zu betrachten) |
| (3) $\alpha AA \rightarrow A\alpha A$ | (α wird weitergeschoben, um den nächsten Fall zu betrachten) |
| (4) $\alpha AB \rightarrow A\alpha B$ | (α wird weitergeschoben, um den nächsten Fall zu betrachten) |
| (5) $\alpha BA \rightarrow A\alpha B$ | (A und B werden getauscht, α wird ein Zeichen weitergeschoben) |
| (6) $\alpha \rightarrow \cdot$ Korrekt | (Halteregel) |
| (7) $\varepsilon \rightarrow \alpha$ | (Schiffchen α wird erzeugt) |

Der Algorithmus sortiert in einem beliebigen Eingabewort aus A's und B's die A's nach links und die B's nach rechts.

Aufgabe 4: Chomsky-Grammatiken (20 T- Punkte)

Geben Sie den einschänkensten Chomsky-Typ der folgenden Grammatiken an, und beründen sie Ihre Aussage durch Analyse einzelner Produktionen.

4.1

Gegeben ist die folgende Grammatik.

Grammatik $G_1 = \{N, \Sigma, P, S\}$

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

$\Sigma = \{a, b\}$

$P = \{S \rightarrow X$

$X \rightarrow AXB \mid ab$

$A \rightarrow a$

$B \rightarrow b\}$

Lösung: 5 Pkt.

Typ CH-2., zum Beispiel wegen $X \rightarrow AXB$. CH-0 und CH-1 Produktionen sind nicht enthalten.

Es werden die Wörter $ab, aabb, aaabbb, \dots$ produziert.

Sprache $L(G_1) = \{a^n b^n, n \geq 1\}$

4.2

Gegeben ist die folgende Grammatik.

Grammatik $G_2 = \{N, \Sigma, P, S\}$

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

$\Sigma = \{a, b\}$

$P = \{S \rightarrow aX \mid B \mid a$

$X \rightarrow aX \mid B$

$B \rightarrow bB \mid b\}$

Lösung: 5 Pkt.

Typ CH-2, Die erzeugte Sprache ist zwar regulär aber in CH-3 sind Kettenproduktionen($S \rightarrow B$) nicht vorgesehen. CH-1, CH-0 Produktionen sind nicht enthalten.

Es werden z.B. die Wörter $a, aabb, abbb, \dots$ produziert.

Sprache $L(G_2) = \{a^n b^m, n \geq 1, m \geq 0\}$

4.3

Gegeben ist die folgende Grammatik.

Grammatik $G_3 = \{N, \Sigma, P, S\}$

$N = \{S, L, R, X\}$

$\Sigma = \{a, b, c, d\}$

$P = \{S \rightarrow X$

$$\begin{aligned}
 X &\rightarrow LXR \mid LR \mid abcd \\
 LR &\rightarrow Lcd \\
 dR &\rightarrow Rd \\
 cR &\rightarrow ccd \\
 Lc &\rightarrow abc \\
 La &\rightarrow aL \\
 Lb &\rightarrow abb
 \end{aligned}$$

Lösung: 5 Pkt.

Typ CH-1 z.B. wegen $dR \rightarrow Rd$, CH-0 Produktionen kommen nicht vor.

Es werden Wörter der Form $abcd$, $aabccdd$, usw. produziert.

Sprache $L(G_3) = \{a^n b^n c^n d^n, n > 1\}$

4.4

Gegeben ist eine Grammatik, die Palindromwörter erzeugt. Ein Palindrom ist ein Wort, das vorwärts wie rückwärts gelesen gleich ist. (Bsp. aba , $abcbba$, $abaccaaccaba$)

$$\begin{aligned}
 \text{Grammatik } G_4 &= \{N, \Sigma, P, S\} \\
 N &= \{S, A, B, C\} \\
 \Sigma &= \{a, b, c\} \\
 P &= \{S \rightarrow ASA \mid BSB \mid CSC \mid a \mid b \mid c \mid \varepsilon \\
 &\quad A \rightarrow a \\
 &\quad B \rightarrow b \\
 &\quad C \rightarrow c\}
 \end{aligned}$$

Zeigen Sie, dass mit Hilfe des Ableitungsbaumes das Wort $accbccbcca$ graphisch beschrieben werden kann.

Die Palindrom-Grammatik kann ohne weiteres auch auf das ganze Alphabet erweitert werden. Einfach die Produktionen $S \rightarrow \#S\#$ und $\# \rightarrow *$ ($\#$ steht für alle Grossbuchstaben des Alphabets und $*$ für den jeweiligen Kleinbuchstaben)

In den Tutorien findet noch ein kleines Gewinnspiel statt:

Wer findet das längste sinnvolle Palindrom? Beispiele: $Anna$, $Otto$, $Vitaler Nebel$ mit $Sinn$ ist im $Leben$ $Relativ$.

Lösung: 5 Pkt.

Typ CH-2 kontextfrei.

Es werden Palindrome erzeugt wie z.B. aa , bb , cc , aaa , aba , aca , bab , bbb , bcb , cac , ccc , $aabbaa$, usw.