

# Übungsaufgaben Theoretische Informatik I

## Blatt 9

Marco Möller <Marco.Moeller@gmxpro.de>  
Matrikel Nummer: 23220320

20. Dezember 2004

### 1 CYK-Algorithmus - Step By Step

#### 1.1 Umschreiben einer Tabelle

Die Definition von  $x_{n,m}$  lässt sich auch umgangssprachlich ausdrücken.  $x_{n,m}$  ist ein Teil des Wortes  $x$  und zwar beginnend ab dem  $n$ -ten Terminal und mit einer Länge von  $m$  Terminalen.

$a_1$	$a_2$	$a_3$	$a_4$
$a_1a_2$	$a_2a_3$	$a_3a_4$	
$a_1a_2a_3$	$a_2a_3a_4$		
$a_1a_2a_3a_4$			

#### 1.2 Nochmal umschreiben einer Tabelle

... ersetzen der Terminale ...

$\alpha$	$\beta$	$\gamma$	$\delta$
$\alpha\beta$	$\beta\gamma$	$\gamma\delta$	
$\alpha\beta\gamma$	$\beta\gamma\delta$		
$\alpha\beta\gamma\delta$			

#### 1.3 Umschreiben der Definition

Die Definition für  $i = 1, \dots, n+1-j$  lässt sich auf beiden Seiten der Gleichung um ein  $j$  erweitern. Dann erhalten wir  $i+j = 1+j, \dots, n+1$ . Daraus folgt, dass  $i+j \leq n+1$  und nicht  $<$ . Also ist die Umschreibung nur *fast* richtig, zu deutsch: *falsch*.

#### 1.4 Ausfüllen der Tabelle laut CYK-Algorithmus

In jede Zelle der Tabelle muss nun die Menge der Variablen eingetragen werden, aus denen man den Inhalt der korrespondierenden Originalzelle ableiten kann.

$\{A\}$	$\{B\}$	$\{A, C\}$	$\{C, D\}$
$\{E\}$	$\emptyset$	$\{F\}$	
$\emptyset$	$\emptyset$		
$\{S\}$			

## 1.5 Sprache der Grammatik $G$

Durch Zeichnen des Ableitungsbaumes erhält man die 4 verschiedenen Wörter die der Grammatik  $G$  entsprechen, und somit Element von  $L(G)$  sind.

$$L(G) = \{\alpha\beta\gamma\delta, \alpha\beta\delta\delta, \gamma\beta\gamma\delta, \gamma\beta\delta\delta\}$$

## 1.6 Typ-3 Grammatik zu $G$

$$\begin{aligned} G' &= (V', \Sigma, P', S) \\ V' &= \{S, A, B, C\} \\ \Sigma &= \{\alpha, \beta, \gamma, \delta\} \\ P' &= \{S \rightarrow \alpha A, \gamma A \\ &\quad A \rightarrow \beta B, \\ &\quad B \rightarrow \gamma C | \delta C, \\ &\quad C \rightarrow \delta\} \end{aligned}$$

## 1.7 Zustandsgraph zu DFA von $G'$



## 2 Something about the CYK-Algorithm

Der CYK-Algorithmus ist eine Methode, die es bei kontextfreien Sprachen erlaubt, das Wortproblem in  $O(n^3)$  zu lösen, was *wesentlich* schneller als die allgemeine Methode ( $O(c^n)$ ) ist. Hierzu wird ausgenutzt, dass sich jede kontextfreie Sprache in eine Chomsky-Normalform überführen lässt. Diese hat die besondere Eigenschaft, dass es nur zwei Arten von Regeln gibt. Als erstes erstellt man beim CYK-Algorithmus zu jedem Terminal des zu untersuchenden Wortes eine Liste (Menge) von Variablen, von denen das jeweilige Terminal erreichbar ist. Nun wird iterativ die Wortlänge verkürzt, dafür werden alle Möglichkeiten aufgelistet (mit konkatenierten Mengen) aus welchen Variablen sich zwei benachbarte Variablen ableiten lassen. Die funktioniert, da beim Ableiten eines Wortes aus einer Grammatik in Chomsky Normalform eben genau das umgekehrte geschieht. Dort können immer nur zwei Variablen pro Ableitungsschritt aus einer Vorgängervariablen entstehen. Wenn man die nun bis zu einer Wortlänge von 1 fortführt, erhält man eine Menge von Variablen, aus denen sich das zu untersuchende Wort ableiten lassen würde. Das Wort ist genau dann wenn die Startvariable in dieser Menge enthalten ist, ein Wort der Sprache. Um diesen Algorithmus möglichst fehlerfrei per Hand durchzuführen sind natürlich weiterführende Schemata nötig, die aber letztendlich auf genau diesen Überlegungen fußen.