CYK: Datenstruktur

Die Mengen $T_{i,j}$ sind Teilmengen der Variablenmenge V. Damit handelt es sich um endliche Mengen. Es gibt in der Größenordnung n^2 viele solche Mengen, wenn das Eingabewort die Länge n hat. Also kann man auf $\mathcal{O}(n^2)$ Platz alle berechneten solchen Mengen abspeichern!

Wir organisieren diese Daten in einer Tabelle, wie nebenstehend gezeigt.

Länge	w_1	w_2	w_3	$\dots$	w_n
1	$T_{1,1}$	$T_{2,1}$	$T_{3,1}$	$\dots$	$T_{n,1}$
2					
				$\dots$	
$n-1$					
n					

CYK: Der Algorithmus in Pseudo-Code

```
INPUT:   w = w_1 w_2 ... w_n
```

INPUT: $w = w_1 w_2 \ldots w_n$

FOR $i := 1$ TO n DO $T[i,1] := \{A \in V \mid A \to w_i \in P\}$ END;

FOR $j := 2$ TO n DO

  FOR $i := 1$ TO $n+1-j$ DO

    $T[i,j] := \emptyset$;

    FOR $k := 1$ TO $j-1$ DO

      $T[i,j] := T[i,j] \cup \{A \in V \mid \exists B, C:\ A \to BC \in P \wedge$
      $B \in T[i,k] \wedge C \in T[i+k, j-k]\}$

    ENDFOR

  ENDFOR

ENDFOR

OUTPUT: JA, falls $S \in T[1,n]$, NEIN, sonst.

Korrektheit und Effizienz

Zur Korrektheit muss man nach den Vorüberlegungen nicht mehr viel sagen. Ein Induktionsbeweis könnte leicht nachweisen, dass die Mengen $T_{i,j}$ nach dem entsprechenden Durchlauf der zugehörigen FOR-Schleife korrekt berechnet sind. Damit ist der am Ende gegebene Output definitiv korrekt.

Laufzeit: Die Laufzeit wird im wesentlichen bestimmt durch die dreifach geschachtelte FOR-Schleife. Da sich diese jeweils höchstens über n Fälle erstreckt, erhalten wir für die Laufzeit die Beschränkung $\mathcal{O}(n^3)$.

Speicherbedarf: Offenbar in $\mathcal{O}(n^2)$.

CYK: Ein Beispiel

Die Grammatik G in CNF ist gegeben durch die folgenden Regeln:

$S \to AX|YB$ $A \to XA|AB|a$ $B \to XY|BB$ $X \to YA|a$ $Y \to XX|b$

Eingabe: $a\ b\ b\ a\ a\ b$

Länge	a	b	b	a	a	b
1	$\{A,X\}$	$\{Y\}$	$\{Y\}$	$\{A,X\}$	$\{A,X\}$	$\{Y\}$
2	$\{B\}$	$\{\}$	$\{X\}$	$\{S,A,Y\}$	$\{B\}$	
3	$\{\}$	$\{\}$	$\{X,A,Y\}$	$\{A\}$		
4	$\{\}$	$\{X\}$	$\{X,B\}$			
5	$\{S,Y\}$	$\{S,B\}$				
6	$\{A,B\}$					

$T_{1,6} = \{A,B\}$,

also gilt $S \notin T_{1,6}$

Und daher: $abbaab \notin L(G)$.

Noch ein Beispiel

Ein weiteres interessantes Beispiel findet man im Buch. Es geht um die Sprache $\{a^n b^n c^m \mid m, n \geq 1\}$, gegeben durch die Typ-2 Grammatik G mit den Regeln $S \to AB$, $A \to ab|aAb$ und $B \to c|cB$.

Führen Sie die Umformung in CNF durch und prüfen Sie dann, ob die folgenden Wörter zu dieser Sprache gehören oder nicht:

aaabbbcc

aaabbccc

aaaabbbbcccc

Kellerautomaten

Ein PDA *(Kellerautomat)* ist definiert wie ein NEA, allerdings mit ein paar nicht ganz unwesentlichen Veränderungen.

Die wichtigste Neuerung beim PDA: Er hat einen sogenannten

Kellerspeicher

Wozu?

Zum Beispiel, um a's *einzukellern*...

(und sie dann später gegen b's zu *matchen*...)

Details hierzu gibt's gleich...