

Anwendung des CYK Algorithmus

Zur Erinnerung: Mit dem CYK-Algorithmus kann man feststellen, ob ein Wort zu einer bestimmten kontextfreien Sprache gehört. Man wendet den Algorithmus auf eine Sprache in Chomsky-Normalform an.

In diesem Kochrezept versuchen wir nachzuweisen, dass sich das Wort **aabbcc** in der folgenden Sprache befindet:

S ? DE
D ? HB
D ? AB
H ? AD
E ? CE
A ? a
B ? b
C ? c
E ? c

Wir bedienen uns dazu einer Dreieckstabelle, deren Felder für die Aufnahme der Mengen dienen. Im ersten Schritt stellen wir eine Tabelle auf die das Wort enthält. Die Bedingung ist erfüllt, wenn das das blau-rot markierte Feld rechts-oben ein S enthält.

Zur Info: Wir fügen Mengen in die Zellen ein. Für diesen Algorithmus bedeutet dies, dass keine Buchstaben doppelt sind und die Reihenfolge egal ist.

easy-coding.de/Attachment/762/...3c8c69cc1b1901fc7a0351894

Grau markiert ist die Zeile auf der wir aktuell arbeiten. Nun suchen wir als erstes alle Produktionen, die in die Nonterminalzeichen zeigen. Bei "a" und "b" ist klar was zu tun ist. Bei "c" stoßen wir schon zum ersten mal auf eine Menge. Sowohl C als auch E zeigen auf "c".

easy-coding.de/Attachment/763/...3c8c69cc1b1901fc7a0351894

Als nächstes ergänzen wir die zweite, jetzt grau hinterlegte Linie. Wir betrachten dazu die unmittelbar benachbarten Zellen.

Für Zelle (2,1) betrachten wir also den linken Nachbar (1,1) "A" und den unteren Nachbarn (2,2) "A". Also suchen wir nach Produktionen die "AA" erzeugen. Es gibt keine.

easy-coding.de/Attachment/764/...3c8c69cc1b1901fc7a0351894

Nach dem selben Prinzip füllen wir die ganze Reihe aus.

(2,1) -> A wird nicht erzeugt
(3,2) -> AB wird durch D erzeugt (Beachten Sie: AB ist nicht BA)
(4,3) -> B wird nicht erzeugt
(5,4) -> BC und BE werden nicht erzeugt
(6,5) -> CC, CE, EC und EE werden erzeugt (E->CE)

easy-coding.de/Attachment/765/...3c8c69cc1b1901fc7a0351894

In der nächsten Zeile kommt nun der eigentliche Algorithmus zum Einsatz. Bisher hat es gereicht die unmittelbaren Nachbarzellen zu vergleichen - das funktionierte aber nur bis hier. Schauen wir uns also im Detail an, wie es nun weitergeht.

easy-coding.de/Attachment/766/...3c8c69cc1b1901fc7a0351894

Wir vergleichen die Zelle in gleicher Höhe, ganz links (1,1) und die Zelle in gleicher Länge (3,2) direkt unter der gefragten Zelle (3,1). Zur Verdeutlichung sind die Zellen, die verglichen werden gelb markiert. Danach wandern wir die Zelle in gleicher Höhe um eine Zelle nach rechts und die Zelle in gleicher Länge um eine Zelle nach unten (grün markiert). Hier gibt es keine Übereinstimmung.

(3,1)

- AD wird durch H erzeugt
- B wird nicht erzeugt

easy-coding.de/Attachment/767/...3c8c69cc1b1901fc7a0351894

Auf die selbe Art fahren wir fort.

(4,2)

- A wird nicht erzeugt
- DB wird nicht erzeugt

(5,3)

- B wird nicht erzeugt
- C oder E werden nicht erzeugt

(6,4)

- BE wird nicht erzeugt
- C oder E werden nicht erzeugt

easy-coding.de/Attachment/768/...3c8c69cc1b1901fc7a0351894

Wesentlich komplexer wird der Algorithmus nicht. Um so weiter wir nach rechts wandern um so mehr Vergleiche müssen wir pro Zelle machen - jedoch werden es immer weniger Zellen die wir überhaupt vergleichen müssen.

In diesem Schritt vergleichen wir nur noch 3 Zellen: (4,1), (5,2), (6,3)

Außerdem sollten wir uns gemerkt haben, dass keine Terminale mit nur einem Zeichen erzeugt werden. Dadurch brauchen wir nicht jedesmal in der Tabelle nachschlagen.

(4,1)

- A wird nicht erzeugt
- ---
- HB wird durch D erzeugt

(5,2)

- A wird nicht erzeugt
- D wird nicht erzeugt
- C oder E werden nicht erzeugt

(6,3)

- B wird nicht erzeugt
- E wird nicht erzeugt
- C oder E werden nicht erzeugt

easy-coding.de/Attachment/769/...3c8c69cc1b1901fc7a0351894

Das selbe Prozedere wenden wir bei der folgenden Zeile an.

easy-coding.de/Attachment/770/...3c8c69cc1b1901fc7a0351894

Im letzten Schritt geht es nur noch darum festzustellen, ob S hergeleitet werden kann. Ist dies der Fall können wir aufhören.

- A wird nicht erzeugt
- ---
- H wird nicht erzeugt
- DE wird durch S erzeugt!!!

easy-coding.de/Attachment/771/...3c8c69cc1b1901fc7a0351894