Ringableitungen entfernen

Eine Ringableitung liegt vor, wenn es Variablen $B_1, \ldots, B_r$ gibt, so dass in P Regeln $B_i \to B_{i+1}$ für $i = 1, \ldots, r-1$ existieren, und außerdem $B_r \to B_1$ eine Regel aus P ist (und $r \geq 2$).

Wenn eine solche Ringableitung existiert, streichen wir alle Variablen B_i und ersetzen sie in allen Regeln durch die *neue* Variable B. Die Regel $B \to B$ wird entfernt.

Alle Ableitungen der ursprünglichen Grammatik können mit der abgewandelten Regelmenge immer noch gebildet werden.

Andererseits gilt in der ursprünglichen Grammatik $B_i \Rightarrow_G^* B_j$ für alle Paare i, j. Daher können wir Ableitungen gemäß der abgewandelten Grammatik zurück übersetzen in Ableitungen der ursprünglichen Grammatik, wenn an den „Nahtstellen“ diese Übergänge benutzt werden.

Variablen anordnen

Wenn alle Ringableitungen eliminiert sind, können wir die Variablen in einem Graph ohne Zyklen darstellen: Eine Kante (A, B) existiert genau dann, wenn $(A, B) \in P$ gilt.

Dieser endliche zyklenfreie Graph muss Knoten vom Ausgangsgrad 0 haben. Wir wählen einen solchen als *letzten* Knoten in der Ordnung und entfernen ihn aus dem Graph. Der neue Graph hat wieder einen solchen Knoten vom Ausgangsgrad 0, der der vorletzte in der Ordnung wird, usw.

Nun seien die Knoten geordnet: $V = \{A_1, \ldots, A_n\}$ und in P kann $A_i \to A_j$ nur dann vorkommen, wenn $i < j$ gilt.

Jetzt eliminieren wir sukzessive Regeln der Form $A_i \to A_j$, indem wir *Abkürzungen* einführen, beginnend bei $i = n - 1$.

Abkürzungen und Pseudoterminale

Die Beseitigung der Regel $A_{n-1} \to A_n$ schaffen wir, indem wir für jede Regel $A_n \to v$ auch die Regel $A_{n-1} \to v$ einführen.

Das tun wir natürlich nur, wenn die Regel $A_{n-1} \to A_n$ wirklich vorhanden war!

Dieses Vorgehen iterieren wir so lange, wie noch Regeln $A_i \to A_j$ mit $i < j$ vorhanden sind – und zwar von hinten nach vorne.

Wir rekapitulieren: Jetzt gilt für jede Regel (u, v) in P entweder $v \in \Sigma$ oder $|v| \geq 2$.

Als nächstes werden Terminale, die in Regeln (u, v) mit $|v| > 1$ vorkommen, durch neue Variablen (eine Variable für jedes Terminal) ersetzt, z.B. a durch V_a, und dann die Regel $V_a \to a$ ergänzt.

Nun haben alle Regeln die Form (u, v), wobei entweder $v \in \Sigma$ gilt oder $v \in V^*$, aber dann $|v| > 1$.

Letzter Schritt

Die störenden Regeln, die wir jetzt noch haben, sind von der Form $A \to C_1 \dots C_k$ mit $k \geq 2$. Für $k = 2$ sind diese schon so, wie in der CNF vorgesehen. Aber was machen wir bei $k > 2$?

Wir wählen neue Variablen $D_2, \dots, D_{k-1}$ und ersetzen die eine Regel $A \to C_1 \dots C_k$ durch folgende $k - 1$ Regeln:

$$A \to C_1 D_2$$

$$D_2 \to C_2 D_3$$

usw. bis

$$D_{k-1} \to C_{k-1} C_k$$

Auch durch diese letzte Anpassung wird die erzeugte Sprache nicht geändert. Also erhalten wir den folgenden Satz...

Satz (Chomsky-Normalform)

Satz: Zu jeder kontextfreien Grammatik G mit $\varepsilon \notin L(G)$ gibt es eine Grammatik G' in Chomsky-Normalform, so dass

$$L(G) = L(G')$$

gilt.

Der Beweis für diesen Satz, den wir in dieser Einheit geführt haben, war konstruktiv – d.h. wir können G' sogar konstruieren!

Schema zur Herstellung einer CNF

Wir werden immer genau nach dem Schema vorgehen, das durch den Beweis der Existenz einer CNF vorgegeben ist:

1) Ringableitungen entfernen
   Nun bilden die Variablen mit Regeln der Form (A, B) einen azyklischen Graph.
2) Variablen anordnen
   Nun können die Regeln der Form (A, B) eliminiert werden.
3) Abkürzungen verwenden
   Jetzt gibt es keine Regeln der Form (A, B) mehr.
4) Pseudoterminale einführen
   Danach stören nur noch Regeln der Form (A, v) mit $|v| > 2$.
5) $A \to C_1 \dots C_k$ mit $k \geq 3$ jeweils durch $k - 1$ Regeln ersetzen.
   Die CNF ist fertig.

Erstes Beispiel zur CNF

Wir betrachten die Grammatik $S \to aSb \mid ab$, mit der die Sprache

$$L = \{a^n b^n \mid n \geq 1\}$$

erzeugt wird.

Es gibt keine Ringableitungen. Das Anordnen der Variablen und Verwenden von Abkürzungen entfällt auch, weil in der gegebenen Grammatik nur eine Variable vorkommt.

Wir führen Pseudoterminale V_a und V_b ein und erhalten:

$$S \to V_a S V_b \mid V_a V_b \qquad V_a \to a \qquad V_b \to b$$

Wir brauchen eine neue Variable C und erhalten als Endergebnis:

$$S \to V_a C \mid V_a V_b \qquad V_a \to a$$
$$C \to S V_b \qquad V_b \to b$$