Forme normali di una grammatica

Esistono più modi per restringere la forma delle produzioni di una grammatica. I principali sono la forma normale di Chomsky e la forma normale di Greibach. Per arrivarci è necessaro introdurre alcune proprietà.

Eliminazione dei simboli inutili

Sia $G =< V, T, P, S >$ una grammatica context free. Un simbolo è utile se esiste una derivazione del tipo

$$S \to^* \alpha x \beta \to^* x \in T^*$$

dove $\alpha, \beta \in (V \cup T)^*$. I simboli inutili possono essere eliminati in due passaggi (due lemmi). Il primo passo consiste nell'eliminare tutte le variabili che non conducono in nessun modo ad una stringa di terminali:

Lemma 1 : Sia $G =< V, T, P, S >$ una grammatica context free con $\mathfrak{L}(G) \neq 0$, allora esiste una grammatica equivalente $G' =< V', T, P', S >$ tale che per ogni $A \in V'$ esiste $x \in T^*$ dove

$$A \to^* x' \quad e \quad \mathfrak{L}(G) = \mathfrak{L}(G)$$

Dimostrazione

Dimostriamo per costruzione che il lemma funziona, ossia produciamo un algoritmo in grado di generare tale grammatica.
Definiamo la funzione $\Gamma : \mathcal{P}(V) \to \mathcal{P}(V)$ nel seguente modo:

$$\Gamma(W) = \{A \in V : \exists \alpha \in (T \cup W)^*. A \to^* \alpha \in P\}$$

La funzione prende in input un insieme di simboli non terminale e restituisce in output un insieme di simboli non terminali dai quali è possibile raggiungere un simbolo terminale o un simbolo presente nell'insieme di input.
Applicando la funzione all'insieme vuoto avremo nell'insieme i soli simboli che direttamente portano ad un simbolo terminale, applicando nuovamente la funzione a questo insieme aggiungeremmo altri simboli, e così via, fino a che non arriveremmo al punto fisso, in cui avremmo aggiunto tutti i simboli ed ogni successiva applicazione non ne aggiungerà di nuovi. Questo può essere verificato per induzione.
Definiamo quindi per casi la funzione $\Gamma^{(n)}(W)$ che ci consentirà di applicare Γ n volte:

$$\begin{cases} \Gamma^0(W) & = W \\ \Gamma^{n+1}(W) & = \Gamma(\Gamma^{n+1}(W)) \end{cases}$$

Notiamo che nel caso pessimo ogni iterazione andrà ad aggiungere un solo simbolo all'insieme per cui dovremmo applicare Γ al più un numero di volte pari alla cardinalità dell'insieme dei simboli non terminali per essere certi di aver incluso tutti i simboli.
Per cui, possiamo definire i nuovi insiemi V' e P' nel seguente modo:

$$V' = \Gamma^{|V|}(\emptyset)$$
$$P' = \{A \to \alpha \in P : A \in V'\}$$

Il secondo passo consiste nell'eliminare tutte le variabili che non sono mai raggiunte da S:

Lemma 2 : Sia $G =< V, T, P, S >$ una grammatica context free, allora esiste una grammatica equivalente $G' =< V', T', P', S' >$ tale che per ogni $x \in (V' \cup T')$ esistono α e β in $(V' \cup T')$ dove

$$S \to^* \alpha x \beta \quad e \quad \mathfrak{L}(G) = \mathfrak{L}(G)$$

Dimostrazione

$\Gamma : P(V \cup T) \rightarrow P(V \cup T)$
$\Gamma(w) = \{x \in V \cup T \cup \{\epsilon\} : \exists A \in w.A \rightarrow \alpha x \beta \in P\} \cup S$

La soluzione per la dimostrazione è la raggiungibilità da parte di S, miriamo a ottenere tutti i simboli raggiungibili da S (che coincidono con l'insieme w).

$\Gamma(\emptyset) = S, \Gamma^2(\emptyset) = A, a, B, S, \Gamma(\emptyset)^3 = A, a, B, S, C$

Si può aggiungere un simbolo alla volta, nel caso peggiore si termina in un numero di passi uguale al numero di simboli.

Teorema Ogni linguaggio context free non vuoto è generato da una grammatica context free priva di simboli inutili.

Dimostrazione

Applichiamo alla grammatica G in sequenza il lemma 1 e il lemma 2.

$$G \xrightarrow{lemma1} G_1 \xrightarrow{lemma2} G_2$$