Forma normale di Chomsky

Di solito si usa scrivere la CFG in forma semplificata *(*utile nelle dimostrazioni pratiche in cui si chiede*, dato il linguaggio X che è context-free, dimostra che y è context-free; bisogna infatti trasformare la grammatica di quell'esercizio in Chomsky),* detta *forma normale di Chomsky:*

Una grammatica context-free è in forma normale di Chomsky se ogni regola è della forma

$$A \to BC$$
$$A \to a$$

dove a è un terminale, B, C non possono essere la variabile iniziale. Inoltre, ci può essere la regola $S \to \varepsilon$ per la variabile iniziale S

in cui si segue questo ordine di regole:

Idea: possiamo trasformare una grammatica G in forma normale di Chomsky:

1. aggiungiamo una nuova variabile iniziale
2. eliminiamo le ε-regole $A \to \varepsilon$
3. eliminiamo le regole unitarie $A \to B$
4. trasformiamo le regole rimaste nella forma corretta

(dove per il punto 5 sotto si intende, rimpiazza ogni variabile terminale sul lato destro di una regola con una nuova variabile e regola non terminale, tale che abbiamo a destra almeno 2 regole non terminali per il punto 4)

The conversion to Chomsky Normal Form has four main steps:

1. Get rid of all ε productions.
2. Get rid of all productions where RHS is one variable.
3. Replace every production that is too long by shorter productions.
4. Move all terminals to productions where RHS is one terminal.

Quindi in generale:

1)

If the Start Symbol S occurs on some right side, create a new Start Symbol S' and a new Production S'→S.

2)

Removal of Null Productions

In a CFG, a Non-Terminal Symbol 'A' is a nullable variable if there is a production A→∈ or there is a derivation that starts at 'A' and leads to ∈. (Like A→................→∈)

Procedure for Removal:

Step 1: To remove A→∈, look for all productions whose right side contains A

Step2: Replace each occurences of 'A' in each of these productions with ∈

Step 3: Add the resultant productions to the Grammar

3)

Any Production Rule of the form $A \to B$ where $A, B \in$ Non Terminals is called Unit Production

Procedure for Removal

Step 1: To remove $A \to B$, add production $A \to x$ to the grammar rule whenever $B \to x$ occurs in the grammar. [$x \in$ Terminal, x can be Null]

Step 2: Delete $A \to B$ from the grammar.

Step 3: Repeat from Step 1 until all Unit Productions are removed.

4)

Replace each Production $A \to B_1 \ldots\ldots B_n$ where $n > 2$, with $A \to B_1 C$ where $C \to B_2 \ldots\ldots B_n$

Repeat this step for all Productions having two or more Symbols on the right side.

5)

If the right side of any Production is in the form $A \to aB$ where 'a' is a terminal and A and B are non-terminals, then the Production is replaced by $A \to XB$ and $X \to a$.

Repeat this step for every Production which is of the form $A \to aB$

Esempio completo step by step Chomsky:

Trasformiamo la grammatica G_6 in forma normale di Chomsky:

$$S \to ASA \mid aB$$
$$A \to B \mid S$$
$$B \to b \mid \varepsilon$$

In questo caso si vede che ci sono regole unitarie, ε-simboli e altro.

Per cominciare a trasformarla consideriamo (il testo barrato è quello eliminato):

1 aggiungiamo una nuova variabile iniziale $S_0 \notin V$ e la regola

$$S_0 \to S$$

In questo modo garantiamo che la variabile iniziale non compare mai sul lato destro di una regola

$G' = (V', \Sigma, R', S_0)$

dove si nota che S appare a destra e si introduce un nuovo stato iniziale.

S’ → S

S → ASA |aB

A → B|S|ε

B→ b|ε

dove metto ε anche in A perché la regola successiva va in B che va a sua volta in ε.

Successivamente:

2 Eliminiamo le ε-regole $A \to \varepsilon$:

- se $A \to \varepsilon$ è una regola dove A non è la variabile iniziale
- per ogni regola del tipo $R \to uAv$, aggiungiamo la regola

$$R \to uv$$

- attenzione: nel caso di più occorrenze di A, consideriamo tutti i casi: per le regole come $R \to uAvAw$, aggiungiamo

$$R \to uvAw \mid uAvw \mid uvw$$

- nel caso di regole $R \to A$ aggiungiamo $R \to \varepsilon$ solo se non abbiamo già eliminato $R \to \varepsilon$
- Ripeti finché non hai eliminato tutte le ε-regole

rimuovo le ε-regole, quindi B → ε ed A → ε:
Rimuovendo B → ε (quindi vuol dire che considero tutte le stringhe dove B è nullo)

S' → S
S → ASA|aB|a
A → B|S|ε
B→b

e poi rimuovo A → ε (tutti i casi con ASA dove a è nullo e tolgo la ε):

S' → S
S → ASA|aB|a|AS|SA|S
A →B|S
B→ b

applicando poi la terza parte:

3 Eliminiamo le regole unitarie $A \to B$:

- se $A \to B$ è una regola unitaria
- per ogni regola del tipo $B \to u$, aggiungiamo la regola

$$A \to u$$

a meno che $A \to u$ non sia una regola unitaria eliminata in precedenza
- Ripeti finché non hai eliminato tutte le regole unitarie

avendo come regole unitarie (regola che va verso un'altra regola non terminale), quindi:
S→S S'→S A→B A→S

Partiamo rimuovendo S → S, che non fa nulla in pratica:
S' → S
S → ASA|aB|a|AS|SA
A →B|S
B→b

quindi eliminiamo S' → S (quindi sostituisco S con ASA|aB|a|AS|SA)
S' → ASA|aB|a|AS|SA
S → ASA|aB|a|AS|SA
A →B|S
B→b

poi eliminiamo A → B (quindi sostituisco B con b):
S’ → ASA|aB|a|AS|SA
S → ASA|aB|a|AS|SA
A →b|S
B→b

ed infine eliminiamo A → S (quindi sostituisco S con ASA|aB|a|AS|SA):
S’ → ASA|aB|a|AS|SA
S → ASA|aB|a|AS|SA
A →b| ASA|aB|a|AS|SA
B→b

4 Trasformiamo le regole rimaste nella forma corretta:

- se $A \rightarrow u_1 u_2 \dots u_k$ è una regola tale che:
  - ogni u_i è una variabile o un terminale
  - $k \geq 3$
- sostituisci la regola con la catena di regole

$$A \rightarrow u_1 A_1, \quad A_1 \rightarrow u_2 A_2, \quad A_2 \rightarrow u_3 A_3, \quad \dots \quad A_{k-2} \rightarrow u_{k-1} u_k$$

- rimpiazza ogni terminale u_i sul lato destro di una regola con una nuova variabile U_i, e aggiungi la regola

$$U_i \rightarrow u_i$$

- ripeti per ogni regola non corretta

Ora dobbiamo trovare le produzioni che hanno più di 2 variabili a destra:
S’ → ASA   S → ASA   A → ASA
In pratica, vedendo che tutte hanno AS oppure SA come variabile rimpiazzabile, posso creare una nuova regola che le sostituisce, ottenendo:

S’ → AX|aB|a|AS|SA
S → AX|aB|a|AS|SA
A → b|AX|aB|a|AS|SA
B→ b
X → SA

Adesso “a” è stato terminale e quindi dobbiamo cambiare tutte le produzioni che contengono “a” con una nuova regola (le espressioni sono S’ → aB, S → ab, A → aB), aggiungendo come regola y → a:

S’ → AX|YB|a|AS|SA
S → AX|YB|a|AS|SA
A → b|AX|YB|a|AS|SA
B→ b
X → SA
Y → a