Azzolini Riccardo 2020-11-24

CFG — Derivazioni e linguaggio generato

1 Derivazione in un passo

Data una CFG $G = \langle V, T, \Gamma, S \rangle$, si definisce la relazione di **derivazione in un passo**, $\Rightarrow_G$, tale che $\alpha A \beta \Rightarrow_G \alpha\gamma\beta$ se e solo se

- $\alpha, \beta, \gamma \in (V \cup T)^*$ sono generiche stringhe sull'insieme dei terminali e non-terminali della grammatica;
- $A \in V$ è un simbolo non-terminale della grammatica;
- γ è il corpo di una regola produzione della grammatica che ha A come testa, cioè l'insieme delle regole di produzione Γ contiene una regola $A \to \gamma$.

Sostanzialmente, questa relazione indica come trasformare stringhe che contengono un simbolo non-terminale in stringhe nelle quali tale simbolo è stato sostituito con il corpo di un'opportuna regola di produzione.

Osservazione: È possibile che γ contenga a sua volta il simbolo non-terminale A.

1.1 Esempi

Si consideri la grammatica

$$G_{pal} = \langle \{P\}, \{0, 1\}, \Gamma, P \rangle \qquad P \to \epsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$$

introdotta in precedenza.

Alcuni esempi di derivazioni in un passo corrette su G_{pal} sono:

- $P \Rightarrow_{G_{pal}} 0$, ottenuta ponendo $\alpha = \beta = \epsilon$ e applicando la regola di produzione $P \to 0$;
- $00P00 \Rightarrow_{G_{pal}} 001P100$, in cui $\alpha = \beta = 00$ e la regola applicata è $P \to 1P1$;
- $00P11 \Rightarrow_{G_{pal}} 0011$, ottenuta con $\alpha = 00$, $\beta = 11$ e $P \to \epsilon$.

Come già anticipato, si dimostrerà che la grammatica G_{pal} genera il linguaggio L_{pal} su $\{0, 1\}$. Si osserva però che, nella derivazione in un passo

$$00P11 \Rightarrow_{G_{pal}} 0011$$

si ottiene una stringa non palindroma, 0011, ma l'applicazione della regola è comunque corretta (tutto è coerente rispetto alla definizione di derivazione in un passo): si vedrà in seguito che 0011 non fa parte del linguaggio generato perché, a partire dal simbolo iniziale della grammatica, non è possibile ottenere la stringa di sinistra della derivazione ($00P11$).

2 Derivazione in zero o più passi

Data una CFG $G = \langle V, T, \Gamma, S \rangle$, la relazione di **derivazione in zero o più passi**, $\overset{*}{\Rightarrow}_G$, è la relazione ottenuta considerando zero o più passi di derivazione. Formalmente, questa relazione può essere definita equivalentemente in modi diversi:

- $\overset{*}{\Rightarrow}_G$ è la chiusura riflessiva e transitiva di $\Rightarrow_G$, cioè la più piccola relazione tale che:
  1. per ogni $\alpha \in (V \cup T)^*$, $\alpha \overset{*}{\Rightarrow}_G \alpha$ (questo è il caso in cui si effettuano zero passi);
  2. se $\alpha \overset{*}{\Rightarrow}_G \beta$ e $\beta \Rightarrow_G \gamma$, allora $\alpha \overset{*}{\Rightarrow}_G \gamma$.
- Si ha $\alpha \overset{*}{\Rightarrow}_G \beta$ se e solo se
  - $\alpha = \beta$
  - oppure
  $$\alpha = \gamma_0 \Rightarrow_G \gamma_1 \Rightarrow_G \cdots \Rightarrow_G \gamma_n = \beta$$
  cioè esistono $\gamma_0, \gamma_1, \dots, \gamma_n \in (V \cup T)^*$, con $n \geq 1$, tali che:
    1. $\gamma_0 = \alpha$;
    2. $\gamma_n = \beta$;
    3. $\forall i = 0, 1, \dots, n-1, \;\; \gamma_i \Rightarrow_G \gamma_{i+1}$.

Osservazione: Definire anche le derivazioni in zero passi, che lasciano la stringa di partenza inalterata, serve ad avere un caso base "comodo" nelle dimostrazioni per induzione di alcune proprietà delle derivazioni.

Notazione: Se la grammatica G è chiara dal contesto, si possono scrivere semplicemente $\Rightarrow$ e $\overset{*}{\Rightarrow}$ invece di $\Rightarrow_G$ e $\overset{*}{\Rightarrow}_G$.

3 Linguaggio generato da una CFG

Data una CFG $G = \langle V, T, \Gamma, S \rangle$, il **linguaggio generato** da G è

$$L(G) = \{w \in T^* \mid S \overset{*}{\Rightarrow} w\}$$

cioè l'insieme delle stringhe composte *solo* da simboli terminali che sono derivabili in zero o più passi a partire dal simboli iniziale della grammatica.

Un linguaggio $L \subseteq \Sigma^*$ è un **linguaggio libero dal contesto** (**Context-Free Language**, **CFL**) se esiste una CFG $G = \langle V, \Sigma, \Gamma, S \rangle$ che genera L, cioè tale che $L(G) = L$.

4 Esempio di derivazione

Si consideri la grammatica semplificata delle espressioni introdotta in precedenza:

$$\begin{gathered} G_{\text{Exp}} = \langle \{E, I\}, \{+, *, (,), a, b, 0, 1\}, \Gamma, E \rangle \\ E \to I \mid E + E \mid E * E \mid (E) \\ I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{gathered}$$

Si vuole generare una stringa appartenente al linguaggio $L(G_{\text{Exp}})$. Per definizione, bisogna partire dal simbolo iniziale della grammatica e applicare regole di produzione fino a ottenere una stringa composta solo da simboli terminali. Scegliendo in modo sostanzialmente casuale quali regole di produzione applicare (perché lo scopo di questo esempio è solo mostrare come funziona il processo), una possibile derivazione è:

$$\begin{aligned}
\boldsymbol{E} &\Rightarrow \boldsymbol{E} * E && \text{Regola } E \to E * E \\
&\Rightarrow \boldsymbol{I} * E && E \to I \\
&\Rightarrow a * \boldsymbol{E} && I \to a \\
&\Rightarrow a * (\boldsymbol{E}) && E \to (E) \\
&\Rightarrow a * (\boldsymbol{E} + E) && E \to E + E \\
&\Rightarrow a * (\boldsymbol{I} + E) && E \to I \\
&\Rightarrow a * (a + \boldsymbol{E}) && I \to a \\
&\Rightarrow a * (a + \boldsymbol{I}) && E \to I \\
&\Rightarrow a * (a + \boldsymbol{I}0) && I \to I0 \\
&\Rightarrow a * (a + \boldsymbol{I}00) && I \to I0 \\
&\Rightarrow a * (a + b00) && I \to b
\end{aligned}$$

(dove ciascun simbolo non-terminale evidenziato in grassetto è quello a cui viene applicata una regola nel passo successivo). La stringa ottenuta all'ultimo passo, $a * (a + b00)$, è formata interamente da simboli terminali: anche volendo, non ci sono più regole da applicare, quindi la derivazione si deve per forza interrompere. Tutta la derivazione viene infine riassunta dalla relazione $E \overset{*}{\Rightarrow} a * (a + b00)$, e si ha che $a * (a + b00) \in L(G_{\text{Exp}})$.

Si può osservare che qui, a ogni passo di derivazione, si è scelto di applicare una qualche regola di produzione al simbolo non-terminale *più a sinistra* nella stringa. Il modo di scegliere su quale simbolo non-terminale lavorare prende il nome di **strategia di derivazione**. In particolare, la scelta del non-terminale più a sinistra è chiamata derivazione **leftmost**, indicata con $\underset{lm}{\Rightarrow}$ e $\underset{lm}{\overset{*}{\Rightarrow}}$: $E \underset{lm}{\overset{*}{\Rightarrow}} a * (a + b00)$. Un'altra è la strategia di derivazione **rightmost** ($\underset{rm}{\Rightarrow}$ e $\underset{rm}{\overset{*}{\Rightarrow}}$), nella quale si applicano le regole di produzione sempre al simbolo non-terminale più a destra. La stringa $a * (a + b00)$ può essere derivata anche con la strategia rightmost, $E \underset{rm}{\overset{*}{\Rightarrow}} a * (a + b00)$:

$$
\begin{aligned}
\boldsymbol{E} &\Rightarrow E * \boldsymbol{E} && \text{Regola } E \to E * E \\
&\Rightarrow E * (\boldsymbol{E}) && E \to (E) \\
&\Rightarrow E * (E + \boldsymbol{E}) && E \to E + E \\
&\Rightarrow E * (E + \boldsymbol{I}) && E \to I \\
&\Rightarrow E * (E + \boldsymbol{I}0) && I \to I0 \\
&\Rightarrow E * (E + \boldsymbol{I}00) && I \to I0 \\
&\Rightarrow E * (\boldsymbol{E} + b00) && I \to b \\
&\Rightarrow E * (\boldsymbol{I} + b00) && E \to I \\
&\Rightarrow \boldsymbol{E} * (a + b00) && I \to a \\
&\Rightarrow \boldsymbol{I} * (a + b00) && E \to I \\
&\Rightarrow a * (a + b00) && I \to a
\end{aligned}
$$

Si vedrà più avanti che, in generale, la scelta della strategia di derivazione non influisce sul linguaggio generato da una CFG.