Azzolini Riccardo 2020-11-26

CFG — Inferenza ricorsiva e alberi sintattici

1 Generazione top-down e bottom-up

Il processo di derivazione visto finora è sostanzialmente un processo **top-down**, dall'alto verso il basso, poiché parte dal simbolo iniziale della grammatica e utilizza le regole di produzione per generare una stringa del linguaggio.

In seguito, verrà invece introdotto un processo **bottom-up**, nel quale le regole di produzione vengono applicate **backward**, dal corpo alla testa, partendo da una stringa del linguaggio e cercando di arrivare al simbolo iniziale.

2 Generazione per inferenza ricorsiva

Il processo di generazione per **inferenza ricorsiva** è sostanzialmente un processo bottom-up. Data una grammatica $G = \langle V, T, \Gamma, S \rangle$, esso funziona generando, per ogni simbolo non-terminale $A \in V$, il corrispondente linguaggio $L(A)$, a partire dai simboli non-terminali che consentono di generare stringhe fatte solo di terminali.

La generazione di $L(A)$ avviene come segue: per ogni regola di produzione con testa A,

$$A \to X_1 \dots X_n \quad \text{con } X_i \in V \cup T$$

appartengono a $L(A)$ tutte le stringhe $\alpha_1 \dots \alpha_n$ tali che:

- $\alpha_i = t_i$, se $X_i = t_i \in T$ è un simbolo terminale;
- $\alpha_i \in L(X_i)$, se X_i è un simbolo non-terminale.

Questo viene fatto per ogni non-terminale della grammatica, tra cui in particolare il simbolo iniziale S, che determina il linguaggio $L(S) = L(G)$ generato dalla grammatica.

Si potrebbe dimostrare che, per ogni grammatica, il linguaggio generato mediante la procedura di inferenza ricorsiva coincide con quello generato mediante derivazioni in zero o più passi. Allora, anche l'inferenza ricorsiva, come la derivazione in zero o più passi, consente di generare la classe dei linguaggi context-free a partire dalle grammatiche context-free.

Da un punto di vista applicativo, la procedura di derivazione e quella di inferenza ricorsiva corrispondono a due diversi meccanismi di riconoscimento per i linguaggi context-free.

2.1 Esempio

Si consideri la solita grammatica delle espressioni:

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

La stringa $a*(a+b00) \in L(G_{\text{Exp}})$ può ad esempio essere generata per inferenza ricorsiva nel modo seguente:

	Stringa di $L(X)$ ricavata	X	Produzione impiegata	Stringhe impiegate
1	a	I	$I \to a$	
2	b	I	$I \to b$	
3	$b0$	I	$I \to I0$	(2) $b \in L(I)$
4	$b00$	I	$I \to I0$	(3) $b0 \in L(I)$
5	a	E	$E \to I$	(1) $a \in L(I)$
6	$b00$	E	$E \to I$	(4) $b00 \in L(I)$
7	$a + b00$	E	$E \to E + E$	$(5, 6)$ $a, b00 \in L(E)$
8	$(a + b00)$	E	$E \to (E)$	(7) $a + b00 \in L(E)$
9	$a * (a + b00)$	E	$E \to E * E$	$(5, 8)$ $a, (a + b00) \in L(E)$

Per prima cosa, sono state generate le uniche stringhe che si possono ottenere senza ricorrere ad altre regole della grammatica: a e b. A partire da queste, si sono poi costruite altre stringhe, utilizzando regole che invece coinvolgono anche simboli non-terminali, fino a ricavare la stringa desiderata $a * (a + b00)$.

Si osservi che qui l'obiettivo non era produrre in modo esaustivo tutte le stringhe possibili, anche perché queste sono infinite, ma piuttosto ottenere una specifica stringa nel linguaggio.

3 Alberi sintattici

Un altro modo per definire il linguaggio associato a una grammatica sono gli *alberi sintattici*.

Un **albero sintattico** per una CFG $G = \langle V, T, \Gamma, S\rangle$ è un albero che soddisfa le seguenti condizioni:

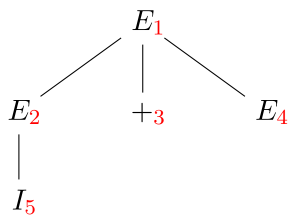
1. ogni *nodo interno* è etichettato da un simbolo non-terminale (un elemento di V);
2. ogni *foglia* è etichettata o da un simbolo non-terminale, o da un simbolo terminale (un elemento di T), oppure dalla stringa vuota ϵ;
3. se una foglia è etichettata ϵ, allora deve essere l'unico figlio del suo nodo padre;
4. se un nodo interno è etichettato $A \in V$ e i suoi figli, ordinati da sinistra verso destra, sono etichettati $X_1, X_2, \ldots, X_k$, allora $A \to X_1 X_2 \ldots X_k$ deve essere una regola di produzione in Γ.

3.1 Esempio

Considerando ancora

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

un esempio di albero sintattico è:



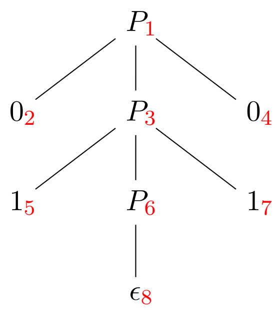
(qui i nodi sono stati numerati, in rosso, per potervi far riferimento più facilmente). Si verifica immediatamente che esso soddisfa le condizioni elencate prima:

1. i nodi interni (1, 2) sono etichettati da simboli non-terminali (in particolare da E);
2. le foglie sono etichettate da simboli non-terminali (4, 5) e terminali (3);
3. non ci sono foglie etichettate ϵ, quindi la terza condizione è vuotamente verificata;
4. tutte le relazioni padre-figlio corrispondono a regole di produzione della grammatica:
   - il nodo interno 1 (la radice), etichettato da E, ha figli etichettati $E, +, E$, ed è vero che $E \to E + E$ è una regola di produzione della grammatica;
   - il nodo interno 2 è etichettato da E e ha un unico figlio etichettato da I, il che corrisponde alla regola di produzione $E \to I$.

Data invece la grammatica

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

un possibile albero sintattico è:

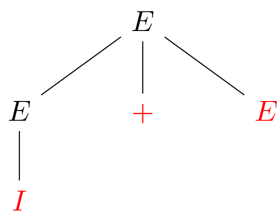


Anche qui sono verificate le quattro condizioni sugli alberi sintattici:

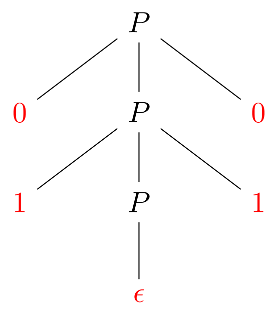
1. i nodi interni (1, 3, 6) sono etichettati da non-terminali;
2. le foglie (2, 4, 5, 7, 8) sono etichettate o da simboli terminali o da ϵ (in questo caso non ci sono foglie etichettate da non-terminali);
3. la foglia 8, etichettata ϵ, è l'unico figlio del suo nodo padre 6;
4. i figli del nodo 1 corrispondono alla regola di produzione $P \to 0P0$, quelli del nodo 3 corrispondono a $P \to 1P1$, e quelli del nodo 6 corrispondono a $P \to \epsilon$.

3.2 Prodotto di un albero sintattico

Il **prodotto di un albero sintattico** è la stringa ottenuta concatenando da sinistra verso destra le foglie dell'albero. Ad esempio, il prodotto dell'albero



è $I + E$, mentre il prodotto di



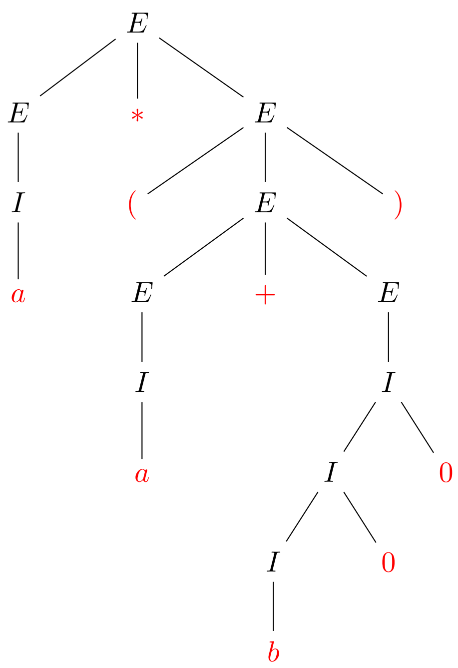
è $01\epsilon10 = 0110$.

Sono particolarmente importanti gli alberi sintattici in cui:

- la radice è etichettata dal simbolo iniziale;
- il prodotto è una **stringa terminale**, cioè composta esclusivamente da simboli terminali, il che avviene quando ogni foglia è etichettata o da un simbolo terminale o da ϵ (e non da un simbolo non-terminale).

Infatti, i prodotti degli alberi di questo tipo su una grammatica G sono le stringhe appartenenti al linguaggio $L(G)$.

Un esempio di albero su G_{Exp} che soddisfa queste ultime condizioni è



il cui prodotto è la stringa terminale $a * (a + b00) \in L(G_{\text{Exp}})$.

4 Definizioni di linguaggio generato da una grammatica

Date una CFG $G = \langle V, T, \Gamma, S\rangle$ e una stringa terminale $w \in T^*$, i seguenti fatti sono **equivalenti**:

- $S \overset{*}{\Rightarrow} w$;
- $S \underset{lm}{\overset{*}{\Rightarrow}} w$;
- $S \underset{rm}{\overset{*}{\Rightarrow}} w$;
- w è generabile da S per inferenza ricorsiva;
- w è il prodotto di un albero sintattico con radice S.

Tali equivalenze, che non verranno dimostrate, implicano che le nozioni di derivabilità, derivabilità leftmost, derivabilità rightmost, inferenza ricorsiva e prodotto di un albero sintattico danno sempre luogo allo stesso linguaggio.