Automi e Linguaggi Formali

Parte 6 – Proprietà delle grammatiche context-free

Davide Bresolin
Ultimo aggiornamento: 27 marzo 2024

Università degli Studi di Padova

Sommario

Alberi sintattici: definizione

Università degli Studi di Padova

Definition

Data una grammatica $G = (V, \Sigma, R, S)$, un albero sintattico è un albero che soddisfa le seguenti condizioni:

1. i nodi interni sono variabili di V
2. le foglie sono, variabili, simboli terminali o ε
3. Se un nodo interno è etichettato con A e i suoi figli sono, da sinistra a destra

   $$X_1, X_2, \ldots, X_k$$

   allora $A \to X_1X_2 \ldots X_k$ è una regola di G

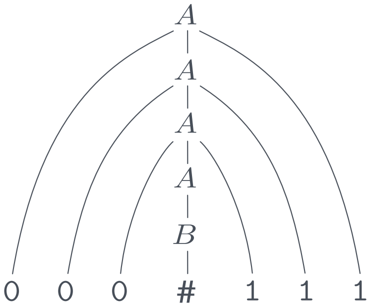
Costruzione di un albero sintattico

Possiamo generare gli alberi sintattici per le stringhe che stanno nel linguaggio di G nel seguente modo:

1. Usa la variabile iniziale come radice dell'albero
2. Trova una foglia etichettata con una variabile A e una regola $A \rightarrow X_1 X_2 \dots X_k$. Aggiungi alla foglia i figli $X_1, \dots, X_k$ da sinistra a destra
3. Ripeti 2 fino a quando tutte le foglie sono etichettate con terminali o ε

Albero sintattico

Esempio di albero sintattico per la grammatica G_1:

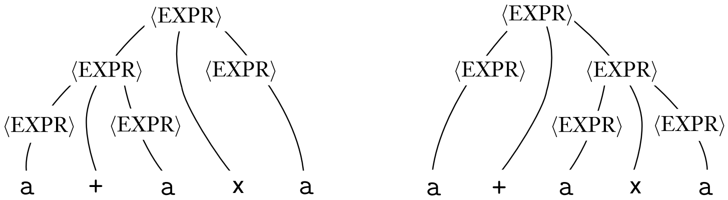


Una nuova grammatica per le espressioni

G_5

$$\langle EXPR\rangle \to \langle EXPR\rangle + \langle EXPR\rangle \mid \langle EXPR\rangle \times \langle EXPR\rangle \mid (\langle EXPR\rangle) \mid a$$

Questa grammatica genera la stringa $\mathtt{a} + \mathtt{a} \times \mathtt{a}$ in due modi diversi!



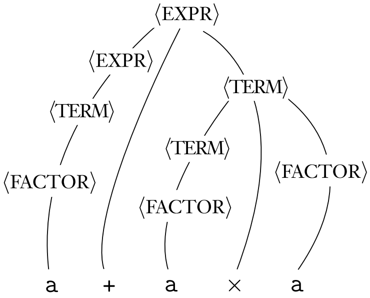
Una grammatica non ambigua

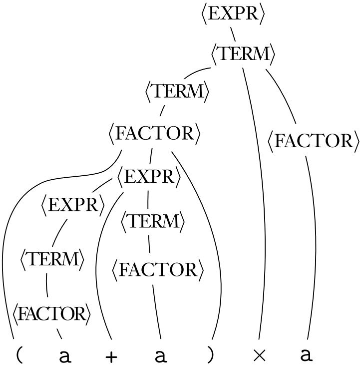
$G_4 = (\{\langle EXPR\rangle, \langle TERM\rangle, \langle FACTOR\rangle\}, \{a, +, \times, (,)\}, R, \langle EXPR\rangle)$

$$\begin{aligned}
\langle EXPR\rangle &\rightarrow \langle EXPR\rangle + \langle TERM\rangle \mid \langle TERM\rangle \\
\langle TERM\rangle &\rightarrow \langle TERM\rangle \times \langle FACTOR\rangle \mid \langle FACTOR\rangle \\
\langle FACTOR\rangle &\rightarrow (\langle EXPR\rangle) \mid a
\end{aligned}$$

Questa grammatica genera alberi sintattici che corrispondono alle regole di valutazione delle espressioni aritmetiche.

Alberti sintattici per G_4





Grammatiche ambigue (1)

- Una grammatica genera ambiguamente una stringa se esistono due alberi sintattici diversi per quella stringa
- Attenzione! Derivazioni diverse possono portare allo stesso albero sintattico!
- Definiamo un ordine per le derivazioni:
  - Derivazione a sinistra (leftmost derivation): ad ogni passo, sostituisco la variabile che si trova più a sinistra.

Grammatiche ambigue (2)

Definition

- Una stringa w è derivata ambiguamente dalla grammatica G se esistono due o più alberi sintattici che la generano
- Equivalente: Una stringa w è derivata ambiguamente dalla grammatica G se esistono due o più derivazioni a sinistra che la generano
- Una grammatica è ambigua se genera almeno una stringa ambiguamente

Linguaggi inerentemente ambigui

- In alcuni casi, possiamo trovare una grammatica non ambigua per il linguaggio
- Esistono linguaggi context-free che sono generati solamente da grammatiche ambigue:
  - li chiameremo linguaggi inerentemente ambigui

- Esempio: il linguaggio $\{a^i b^j c^k \mid i = j \text{ oppure } j = k\}$

Esercizio

Dimostrare che $\{a^i b^j c^k \mid i = j \text{ oppure } j = k\}$ è inerentemente ambiguo.

Sommario

- È spesso conveniente avere le grammatiche in una forma semplificata
- Una delle forme più semplici e utili è la Forma Normale Chomsky

Definition

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

Trasformazione di grammatiche

Theorem

Ogni linguaggio context-free è generato da una grammatica in forma normale di Chomsky

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

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

Esempio

Trasformiamo la grammatica G_6 in forma normale di Chomsky:

$$
\begin{aligned}
S &\to ASA \mid aB \\
A &\to B \mid S \\
B &\to b \mid \varepsilon
\end{aligned}
$$

Algoritmo di trasformazione (1)

Per trasformare $G = (V, \Sigma, R, S)$ in Forma Normale di Chomsky

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

Algoritmo di trasformazione (2)

Per trasformare $G = (V, \Sigma, R, S)$ in Forma Normale di Chomsky

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

Algoritmo di trasformazione (3)

Per trasformare $G = (V, \Sigma, R, S)$ in Forma Normale di Chomsky

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

Algoritmo di trasformazione (4)

Per trasformare $G = (V, \Sigma, R, S)$ in Forma Normale di Chomsky

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

Per ogni linguaggio L, sia

$$\mathrm{suffix}(L) = \{v \mid uv \in L \text{ per qualche stringa } u\}.$$

Dimostra che se L è un linguaggio context-free, allora anche $\mathrm{suffix}(L)$ è un linguaggio context-free.