Grammatiche libere da contesto

Grammatiche e Linguaggi Liberi da Contesto

- Abbiamo visto che molti linguaggi non sono regolari. Consideriamo allora classi più grandi di linguaggi.
- I **Linguaggi Liberi da Contesto** (CFL) sono stati usati nello studio dei linguaggi naturali dal 1950, e nello studio dei compilatori dal 1960.
- Le **grammatiche libere da contesto** (CFG) sono la base della sintassi BNF (Backus-Naur-Form).
- Oggi i CFL sono importanti per XML.
- Studieremo: CFG, i linguaggi che generano e gli alberi sintattici.

Esempio informale di CFG

- Consideriamo $L_{pal} = \{w \in \Sigma^* : w = w^R\}$
- Per esempio: `otto` $\in L_{pal}$, `madamimadam` $\in L_{pal}$.
- Sia $\Sigma = \{0, 1\}$ e supponiamo che L_{pal} sia regolare.
- Sia n dato dal pumping lemma. Allora $0^n10^n \in L_{pal}$. Nel leggere 0^n il FA deve passare per un loop. Se omettiamo il loop, contraddizione.
- Definiamo L_{pal} induttivamente:
  - **Base:** $\epsilon, 0$, e 1 sono palindromi.
  - **Induzione:** Se w è palindroma, anche $0w0$ e $1w1$ lo sono.
  - Nessun altra stringa è palindroma.

Le CFG sono un modo formale per definire linguaggi come L_{pal}.

1. $P \to \epsilon$
2. $P \to 0$
3. $P \to 1$
4. $P \to 0P0$
5. $P \to 1P1$

- 0 e 1 sono *terminali*
- P è una *variabile* (o *nonterminale*, o *categoria sintattica*)
- P è in questa gramatica anche il *simbolo iniziale*.
- 1–5 sono *produzioni* (o *regole*)

Definizione formale di CFG

Una *grammatica libera da contesto* è una quadrupla

$$G = (V, T, P, S)$$

dove

- V è un insieme finito di *variabili* o *[simboli] non terminali* o *categorie sintattiche*.
- T è un insieme finito di *[simboli] terminali*.
- P è un insieme finito di *produzioni* della forma $A \to \alpha$, dove A è una variabile, la *testa della produzione*, e $\alpha \in (V \cup T)^*$ è il *corpo della produzione*
- S è una variabile distinta chiamata il *simbolo iniziale*.

Esempi

- $G_{pal} = (\{P\}, \{0, 1\}, A, P)$, dove $A = \{P \to \epsilon, P \to 0, P \to 1, P \to 0P0, P \to 1P1\}$.
- A volte raggruppiamo le produzioni con la stessa testa: $A = \{P \to \epsilon | 0 | 1 | 0P0 | 1P1\}$.
- Le espressioni regolari su $\{0, 1\}$ possono essere definite dalla grammatica

  $$G_{regex} = (\{E\}, \{0, 1\}, A, E)$$

  dove A corrisponde a

  $$\{E \to \mathbf{0}, E \to \mathbf{1}, E \to E.E, E \to E + E, E \to E^\star, E \to (E)\}$$

Esempio

Espressioni (semplici) in un tipico linguaggio di programmazione.
Gli operatori sono + e *, e gli operandi sono identificatori, cioè stringhe in $L((\mathbf{a}+\mathbf{b})(\mathbf{a}+\mathbf{b}+\mathbf{0}+\mathbf{1})^*)$
Usiamo la grammatica $G = (\{E, I\}, T, P, E)$ dove
$T = \{+, *, (,), a, b, 0, 1\}$ e P è il seguente insieme di produzioni:

1. $E \to I$
2. $E \to E + E$
3. $E \to E * E$
4. $E \to (E)$
5. $I \to a$
6. $I \to b$
7. $I \to Ia$
8. $I \to Ib$
9. $I \to I0$
10. $I \to I1$

Derivazioni usando le grammatiche

- *Inferenza ricorsiva*, usando le produzioni dal corpo alla testa
- *Derivazioni*, usando le produzioni dalla testa al corpo

Esempio di inferenza ricorsiva:

	Stringa	Ling.	Prod.	Stringhe usate
(i)	a	I	$5. I \to a$	-
(ii)	b	I	$6. I \to b$	-
(iii)	$b0$	I	$9. I \to I0$	*(ii)*
(iv)	$b00$	I	$9. I \to I0$	*(iii)*
(v)	a	E	$1. E \to I$	*(i)*
(vi)	$b00$	E	$1. E \to I$	*(iv)*
(vii)	$a + b00$	E	$2. E \to E + E$	*(v), (vi)*
(viii)	$(a + b00)$	E	$4. E \to (E)$	*(vii)*
(ix)	$a * (a + b00)$	E	$3. E \to E * E$	*(v), (viii)*

Derivazioni

- Sia $G = (V, T, P, S)$ una CFG, $A \in V$, $\{\alpha, \beta\} \subset (V \cup T)^*$, e $A \to \gamma \in P$.
- Allora scriviamo

$$\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$$

  o, se è ovvia la G,

$$\alpha A \beta \Rightarrow \alpha \gamma \beta$$

  e diciamo che da $\alpha A \beta$ **si deriva** $\alpha \gamma \beta$.
- Definiamo $\overset{*}{\Rightarrow}$ la chiusura riflessiva e transitiva di $\Rightarrow$, cioe':
  - **Base:** Sia $\alpha \in (V \cup T)^*$. Allora $\alpha \overset{*}{\Rightarrow} \alpha$.
  - **Induzione:** Se $\alpha \overset{*}{\Rightarrow} \beta$, e $\beta \Rightarrow \gamma$, allora $\alpha \overset{*}{\Rightarrow} \gamma$.

Esempio

Derivazione di $a * (a + b00)$ da E nella grammatica delle espressioni:

$$E \Rightarrow E*E \Rightarrow I*E \Rightarrow a*E \Rightarrow a*(E) \Rightarrow$$
$$a*(E+E) \Rightarrow a*(I+E) \Rightarrow a*(a+E) \Rightarrow a*(a+I) \Rightarrow$$
$$a*(a+I0) \Rightarrow a*(a+I00) \Rightarrow a*(a+b00)$$

- Ad ogni passo potremmo avere varie regole tra cui scegliere, ad esempio
  $I*E \Rightarrow a*E \Rightarrow a*(E)$, oppure
  $I*E \Rightarrow I*(E) \Rightarrow a*(E)$.
- Non tutte le scelte portano a derivazioni di una particolare stringa, per esempio

$$E \Rightarrow E+E$$

  non ci fa derivare $a*(a+b00)$.

Derivazioni a sinistra e a destra

- **Derivazione a sinistra** $\underset{lm}{\Rightarrow}$: rimpiazza sempre la variabile più e a sinistra con il corpo di una delle sue regole.
- **Derivazione a destra** $\underset{rm}{\Rightarrow}$: rimpiazza sempre la variabile più a destra con il corpo di una delle sue regole.
- Der. a sinistra: quella del lucido precedente.
- A destra:

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow}$$

$$E * (E) \underset{rm}{\Rightarrow} E * (E + E) \underset{rm}{\Rightarrow} E * (E + I) \underset{rm}{\Rightarrow} E * (E + I0)$$

$$\underset{rm}{\Rightarrow} E * (E + I00) \underset{rm}{\Rightarrow} E * (E + b00) \underset{rm}{\Rightarrow} E * (I + b00)$$

$$\underset{rm}{\Rightarrow} E * (a + b00) \underset{rm}{\Rightarrow} I * (a + b00) \underset{rm}{\Rightarrow} a * (a + b00)$$

Possiamo concludere che $E \overset{*}{\underset{rm}{\Rightarrow}} a * (a + b00)$

Il linguaggio di una grammatica

- Se $G(V, T, P, S)$ è una CFG, allora il **linguaggio di** G è

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

cioè l'insieme delle stringhe su T^* derivabili dal simbolo iniziale.

- Se G è una CFG, chiameremo $L(G)$ un **linguaggio libero da contesto**.
- Esempio: $L(G_{pal})$ è un linguaggio libero da contesto.
- Il linguaggio è visto come l'insieme delle stringhe generate dalla grammatica (approccio *generativo-sintetico*), mentre finora come l'abbiamo visto come l'insieme delle stringhe riconosciute o accettate dagli automi (approccio *riconoscitivo-analitico*).

Teorema 5.7:

$$L(G_{pal}) = \{w \in \{0,1\}^* : w = w^R\}$$

Dimostrazione: (direzione $\supseteq$) Supponiamo $w = w^R$. Mostriamo per induzione su $|w|$ che $w \in L(G_{pal})$.

- **Base:** $|w| = 0$, or $|w| = 1$. Allora w è $\epsilon, 0$, or 1. Dato che $P \to \epsilon$, $P \to 0$, and $P \to 1$ sono produzioni, concludiamo che $P \overset{*}{\underset{G}{\Rightarrow}} w$ in tutti i casi base.
- **Induzione:** Supponiamo $|w| \geq 2$. Dato che $w = w^R$, abbiamo $w = 0x0$, o $w = 1x1$, e $x = x^R$.
  Se $w = 0x0$ sappiamo che per l'ipotesi induttiva $P \overset{*}{\Rightarrow} x$.
  Allora
  $$P \Rightarrow 0P0 \overset{*}{\Rightarrow} 0x0 = w$$
  Quindi $w \in L(G_{pal})$.
  Il caso di $w = 1x1$ è simile.

(direzione $\subseteq$) Supponiamo che $w \in L(G_{pal})$ e dobbiamo mostrare che $w = w^R$.
Dato che $w \in L(G_{pal})$, abbiamo $P \overset{*}{\Rightarrow} w$.
Faremo un'induzione sulla lunghezza di $\overset{*}{\Rightarrow}$.

- **Base:** La derivazione $P \overset{*}{\Rightarrow} w$ ha 1 passo.
  Allora w deve essere $\epsilon, 0$, o 1, tutte palindromi.
- **Induzione:** Sia $n \geq 1$, e supponiamo che la derivazione abbia $n + 1$ passi. Allora dobbiamo avere
  $$w = 0x0 \overset{*}{\Leftarrow} 0P0 \Leftarrow P$$
  o
  $$w = 1x1 \overset{*}{\Leftarrow} 1P1 \Leftarrow P$$
  dove la seconda derivazione ha n passi.
  Per l'ipotesi induttiva, x è palindroma.

Forme sentenziali

- Sia $G = (V, T, P, S)$ una CFG, e $\alpha \in (V \cup T)^*$.
- Se $S \overset{*}{\Rightarrow} \alpha$ diciamo che α è una **forma sentenziale**.
- Se $S \underset{lm}{\Rightarrow} \alpha$ diciamo che α è una **forma sentenziale sinistra**,
- Se $S \underset{rm}{\Rightarrow} \alpha$ diciamo che α è una **forma sentenziale destra**
- Nota: $L(G)$ contiene le forme sentenziali che sono in T^*.

Esempi

- Prendiamo la G delle espressioni. Allora $E * (I + E)$ è una forma sentenziale perché

$$E \Rightarrow E * E \Rightarrow E * (E) \Rightarrow E * (E + E) \Rightarrow E * (I + E)$$

  Questa derivazione non è né a sinistra né a destra
- $a * E$ è una forma sentenziale sinistra, perché

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} I * E \underset{lm}{\Rightarrow} a * E$$

- $E * (E + E)$ è una forma sentenziale destra, perché

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E)$$

Alberi sintattici

- Se $w \in L(G)$, per una CFG, allora w ha un **albero sintattico**, che ci dice la struttura (sintattica) di w
- w potrebbe essere un programma, una query SQL, un documento XML, ...
- Gli alberi sintattici sono una rappresentazione alternativa alle derivazioni e alle inferenze ricorsive.
- Ci possono essere diversi alberi sintattici per la stessa stringa
- Idealmente ci dovrebbe essere solo un albero sintattico (la "vera" struttura), cioè il linguaggio dovrebbe essere non ambiguo.
- Sfortunatamente, non sempre possiamo rimuovere l'ambiguità.

Costruzione di un albero sintattico

Sia $G = (V, T, P, S)$ una CFG. Un albero è un **albero sintattico** per G se:

1. Ogni nodo interno è etichettato con una variabile in V.
2. Ogni foglia è etichettata con un simbolo in $V \cup T \cup \{\epsilon\}$. Ogni foglia etichettata con ϵ deve essere l'unico figlio del suo genitore.
3. Se un nodo interno è etichettato A, e i suoi figli (da sinistra a destra) sono etichettati

$$X_1 X_2 \ldots X_k$$

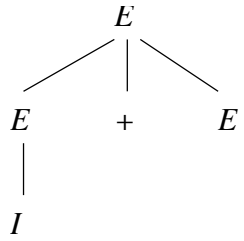
allora $A \to X_1 X_2 \ldots X_k \in P$.

Esempio

Nella grammatica

1. $E \to I$
2. $E \to E + E$
3. $E \to E * E$
4. $E \to (E)$

il seguente è un albero sintattico:



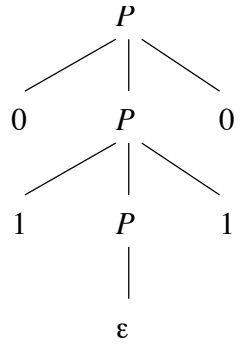
Questo albero sintattico mostra la derivazione $E \overset{*}{\Rightarrow} I + E$

Esempio

Nella grammatica

1. $P \to \epsilon$
2. $P \to 0$
3. $P \to 1$
4. $P \to 0P0$
5. $P \to 1P1$

il seguente è un albero sintattico:

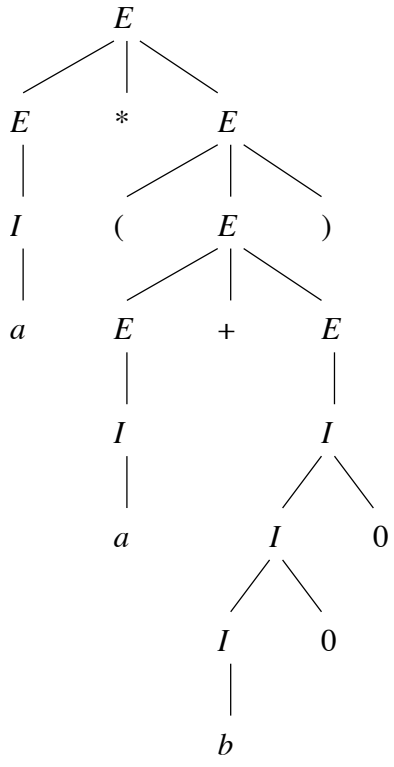


Mostra la derivazione $P \overset{*}{\Rightarrow} 0110$.

Il prodotto di un albero sintattico

- Il **prodotto** di un albero sintattico è la stringa di foglie da sinistra a destra.
- Sono importanti quegli alberi sintattici dove:
  1. Il prodotto è una stringa terminale.
  2. La radice è etichettata dal simbolo iniziale.
- L'insieme dei prodotti di questi alberi sintattici è il linguaggio della grammatica.

Esempio

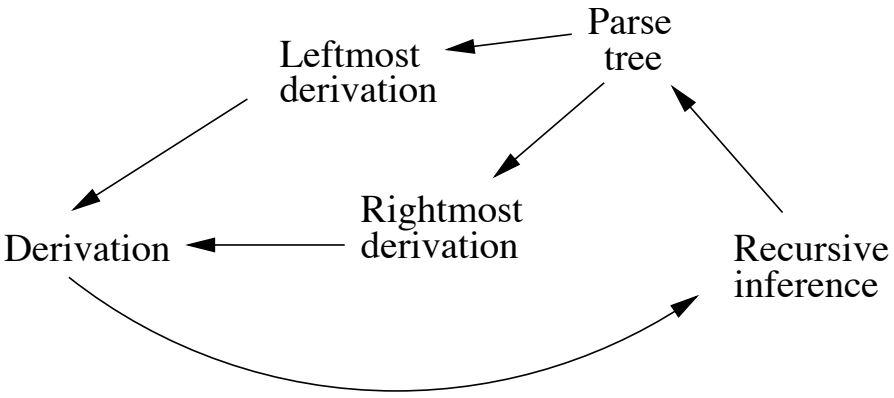


Il prodotto è $a * (a + b00)$.

Sia $G = (V, T, P, S)$ una CFG, e $A \in V$. I seguenti sono equivalenti:

1. Possiamo determinare per inferenza ricorsiva che w è nel linguaggio di A
2. $A \stackrel{*}{\Rightarrow} w$
3. $A \underset{lm}{\stackrel{*}{\Rightarrow}} w$, e $A \underset{rm}{\stackrel{*}{\Rightarrow}} w$
4. C'è un albero sintattico di G con radice A e prodotto w.

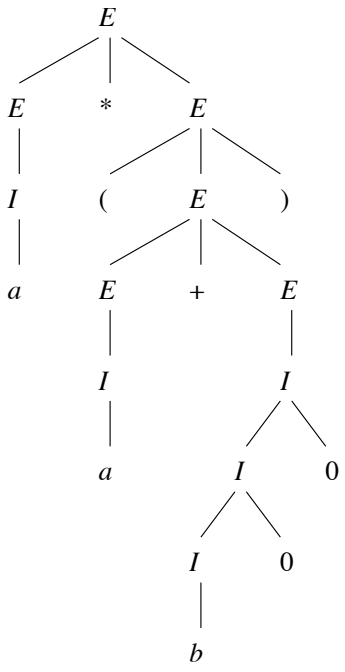
Per provare l'equivalenza, usiamo il seguente piano.



- Dalle inferenze ricorsive agli alberi: partiamo dall'ultimo passo e costruiamo sottoalberi, poi passiamo ai passi precedenti fino al primo.
- Dagli alberi alle derivazioni a sinistra: seguo l'albero da sinistra a destra.
- Dalle derivazioni alle inferenze ricorsive: vado all'indietro nella derivazione.

Esempio

Costruiamo la derivazione a sinistra per l'albero



Supponiamo di aver induttivamente costruito la derivazione a sinistra

$$E \underset{lm}{\Rightarrow} I \underset{lm}{\Rightarrow} a$$

corrispondente al sottoalbero più a sinistra, e la derivazione a sinistra

$$E \underset{lm}{\Rightarrow} (E) \underset{lm}{\Rightarrow} (E+E) \underset{lm}{\Rightarrow} (I+E) \underset{lm}{\Rightarrow} (a+E) \underset{lm}{\Rightarrow}$$

$$(a+I) \underset{lm}{\Rightarrow} (a+I0) \underset{lm}{\Rightarrow} (a+I00) \underset{lm}{\Rightarrow} (a+b00)$$

corrispondente al sottoalbero più a destra.

Per la derivazione corrispondente all'intero albero, iniziamo con $E \underset{lm}{\Rightarrow} E*E$ e espandiamo la prima E con la prima derivazione e la seconda E con la seconda derivazione:

$$E \underset{lm}{\Rightarrow} E*E \underset{lm}{\Rightarrow}$$
$$I*E \underset{lm}{\Rightarrow} a*E \underset{lm}{\Rightarrow}$$
$$a*(E) \underset{lm}{\Rightarrow} a*(E+E) \underset{lm}{\Rightarrow}$$
$$a*(I+E) \underset{lm}{\Rightarrow} a*(a+E) \underset{lm}{\Rightarrow}$$
$$a*(a+I) \underset{lm}{\Rightarrow} a*(a+I0) \underset{lm}{\Rightarrow}$$
$$a*(a+I00) \underset{lm}{\Rightarrow} a*(a+b00)$$