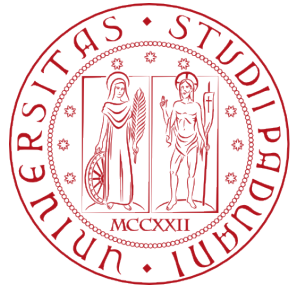
Automi e Linguaggi Formali

Parte 5 – Linguaggi Context-free



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Davide Bresolin
Ultimo aggiornamento: 21 marzo 2021

Sommario

Linguaggi Context-free

- Abbiamo visto che esistono linguaggi non regolari
  - Esempio: $\{0^n 1^n \mid n \geq 0\}$
- Consideriamo allora una classe più grande di linguaggi:
  - i Linguaggi Context-Free (CFL), usati nello studio dei linguaggi naturali dal 1950, e nello studio dei compilatori dal 1960
- Vedremo due metodi per descrivere CFL:
  - Grammatiche context-free (CFG)
  - Automi a pila (pushdown automata)

Una grammatica context-free

La grammatica G_1:

$$A \to 0A1$$
$$A \to B$$
$$B \to \#$$

- insieme di regole di sostituzione (o produzioni)
- variabili: A, B
- terminali (simboli dell'alfabeto): $0, 1, \#$
- variabile iniziale: A

Grammatiche e derivazioni

Una grammatica genera stringhe nel seguente modo:

1. Scrivi la variabile iniziale
2. Trova una variabile che è stata scritta e una regola che inizia con quella variabile. Sostituisci la variabile con il lato destro della regola
3. Ripeti 2 fino a quando non ci sono più variabili

Grammatiche e derivazioni

Una grammatica genera stringhe nel seguente modo:

1. Scrivi la variabile iniziale
2. Trova una variabile che è stata scritta e una regola che inizia con quella variabile. Sostituisci la variabile con il lato destro della regola
3. Ripeti 2 fino a quando non ci sono più variabili

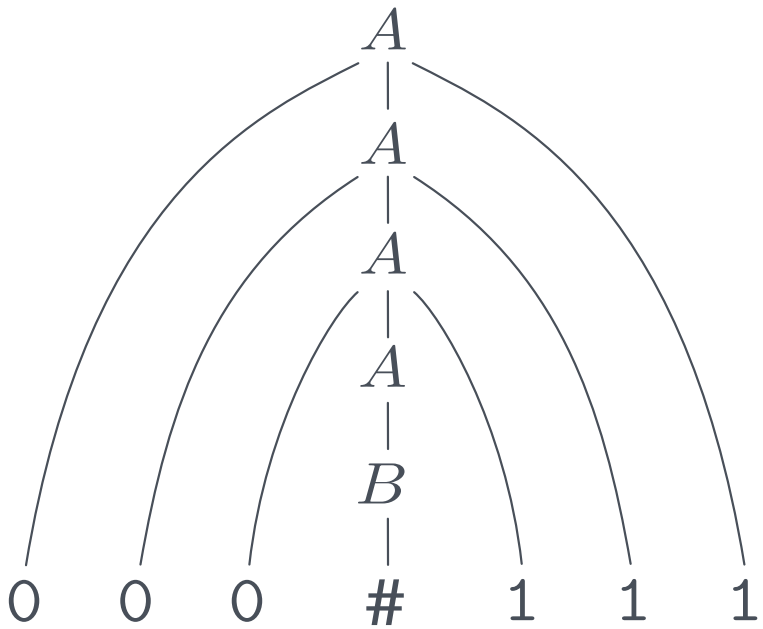
Esempio per G_1:

$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000\#111$$

La sequenza di sostituzioni si chiama derivazione di $000\#111$

Albero sintattico

Una derivazione definisce un albero sintattico (parse tree):



- la radice è la variabile iniziale
- i nodi interni sono variabili
- le foglie sono terminali

Una grammatica per l'inglese

G_2: grammatica per un frammento della lingua inglese

$$
\begin{aligned}
\langle \mathit{SENTENCE} \rangle &\to \langle \mathit{NOUN\_PHRASE} \rangle \langle \mathit{VERB\_PHRASE} \rangle \\
\langle \mathit{NOUN\_PHRASE} \rangle &\to \langle \mathit{CMPLX\_NOUN} \rangle \\
\langle \mathit{NOUN\_PHRASE} \rangle &\to \langle \mathit{CMPLX\_NOUN} \rangle \langle \mathit{PREP\_PHRASE} \rangle \\
\langle \mathit{VERB\_PHRASE} \rangle &\to \langle \mathit{CMPLX\_VERB} \rangle \\
\langle \mathit{VERB\_PHRASE} \rangle &\to \langle \mathit{CMPLX\_VERB} \rangle \langle \mathit{PREP\_PHRASE} \rangle \\
\langle \mathit{PREP\_PHRASE} \rangle &\to \langle \mathit{PREP} \rangle \langle \mathit{CMPLX\_NOUN} \rangle \\
\langle \mathit{CMPLX\_NOUN} \rangle &\to \langle \mathit{ARTICLE} \rangle \langle \mathit{NOUN} \rangle \\
\langle \mathit{CMPLX\_VERB} \rangle &\to \langle \mathit{VERB} \rangle \mid \langle \mathit{VERB} \rangle \langle \mathit{NOUN\_PHRASE} \rangle \\
\langle \mathit{ARTICLE} \rangle &\to \texttt{a} \mid \texttt{the} \\
\langle \mathit{NOUN} \rangle &\to \texttt{boy} \mid \texttt{girl} \mid \texttt{flower} \\
\langle \mathit{VERB} \rangle &\to \texttt{touches} \mid \texttt{likes} \mid \texttt{sees} \\
\langle \mathit{PREP} \rangle &\to \texttt{with}
\end{aligned}
$$

Stringhe generate da G_2

Tra le stringhe nel linguaggio di G_2 ci sono:

```
a boy sees
the boy sees a flower
a girl with a flower likes the boy
```

Esempio di derivazione:

$$
\begin{aligned}
\langle SENTENCE \rangle &\Rightarrow \langle NOUN\_PHRASE \rangle \langle VERB\_PHRASE \rangle \\
&\Rightarrow \langle CMPLX\_NOUN \rangle \langle VERB\_PHRASE \rangle \\
&\Rightarrow \langle ARTICLE \rangle \langle NOUN \rangle \langle VERB\_PHRASE \rangle \\
&\Rightarrow \texttt{a}\ \langle NOUN \rangle \langle VERB\_PHRASE \rangle \\
&\Rightarrow \texttt{a boy}\ \langle VERB\_PHRASE \rangle \\
&\Rightarrow \texttt{a boy}\ \langle CMPLX\_VERB \rangle \\
&\Rightarrow \texttt{a boy}\ \langle VERB \rangle \\
&\Rightarrow \texttt{a boy sees}
\end{aligned}
$$

Definizione formale

Definition

Una grammatica context-free è una quadrupla (V, Σ, R, S), dove

- V è un insieme finito di variabili
- Σ è un insieme finito di terminali disgiunto da V
- R è un insieme di regole, dove ogni regola è una variabile e una stringa di variabili e terminali
- $S \in V$ è la variabile iniziale

Linguaggio di una grammatica

Se u, v, w sono stringhe di variabili e terminali e $A \to w$ è una regola:

- uAv produce uwv: $uAv \Rightarrow uwv$
- u deriva v: $u \Rightarrow^* v$ se:
  - $u = v$, oppure
  - esiste una sequenza $u_1, u_2, \ldots, u_k$ tale che $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k \Rightarrow v$
- il linguaggio della grammatica è $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$

Linguaggi context-free

linguaggi generati da grammatiche context-free

Esempi

$G_3 = (\{S\}, \{(,)\}, R, S)$

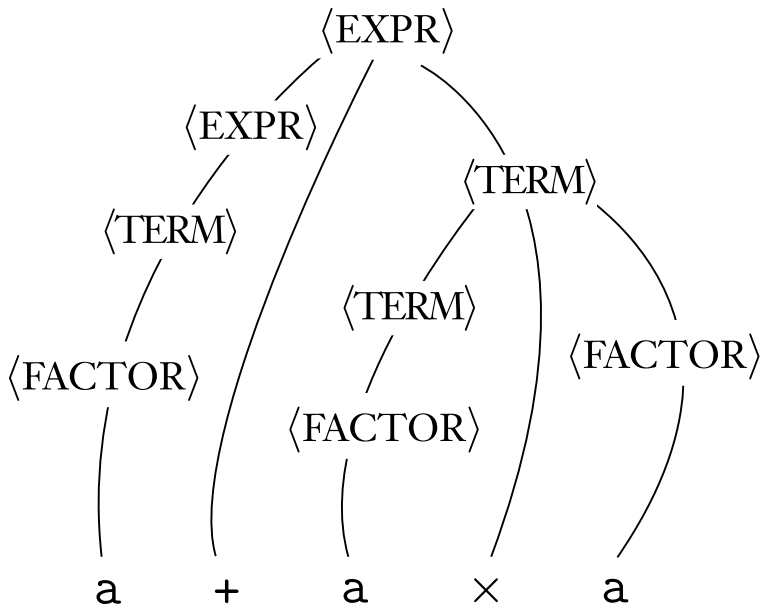
$$S \to (S) \mid SS \mid \varepsilon$$

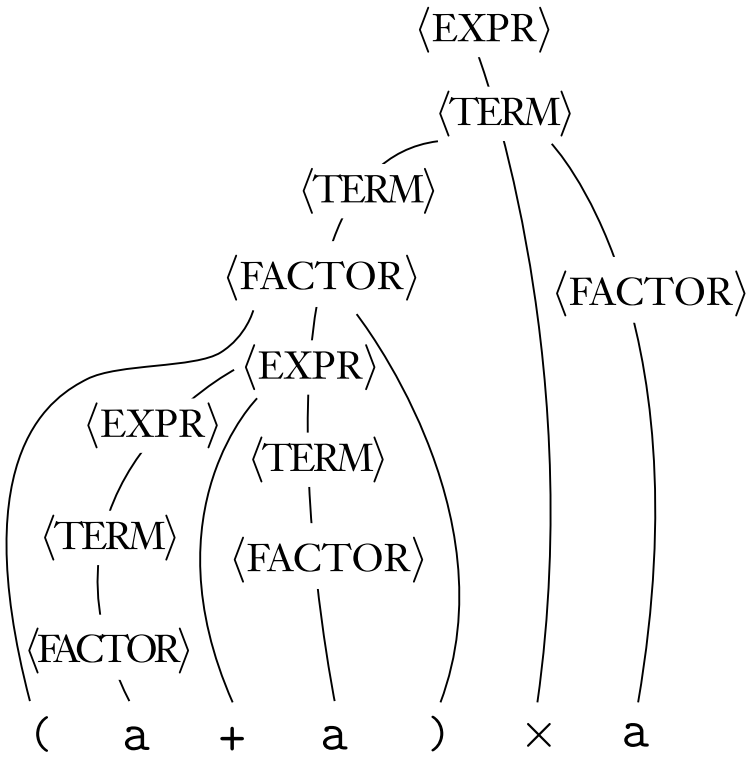
$G_4 = (\{\langle EXPR\rangle, \langle TERM\rangle, \langle FACTOR\rangle\}, \{a, +, \times, (,)\}, R, \langle EXPR\rangle)$

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

Quali sono i linguaggi delle grammatiche G_3 e G_4?

Alberti sintattici per G_4





Sommario

Progettare grammatiche context-free

- Progettare una grammatica è un processo creativo
- È un processo che non può essere ridotto a un processo meccanico
- Esistono però delle tecniche utili che si possono usare

(1) Unione di linguaggi più semplici

- Molti linguaggi sono unione di linguaggi più semplici
- Idea:
  - costruisci grammatiche separate per ogni componente
  - unisci le grammatiche con una nuova regola iniziale

$$S \to S_1 \mid S_2 \mid \cdots \mid S_k$$

dove $S_1, S_2, \ldots, S_k$ sono le regole iniziali delle componenti

(1) Unione di linguaggi più semplici

- Molti linguaggi sono unione di linguaggi più semplici
- Idea:
  - costruisci grammatiche separate per ogni componente
  - unisci le grammatiche con una nuova regola iniziale

$$S \to S_1 \mid S_2 \mid \cdots \mid S_k$$

dove $S_1, S_2, \ldots, S_k$ sono le regole iniziali delle componenti

Esempio: grammatica per $\{0^n 1^n \mid n \geq 0\} \cup \{1^n 0^n \mid n \geq 0\}$

- Grammatica per $0^n 1^n$: $S_1 \to 0S_1 1 \mid \varepsilon$
- Grammatica per $1^n 0^n$: $S_2 \to 1S_2 0 \mid \varepsilon$
- Unione: $S \to S_1 \mid S_2$

(2) Trasformare un DFA in CFG

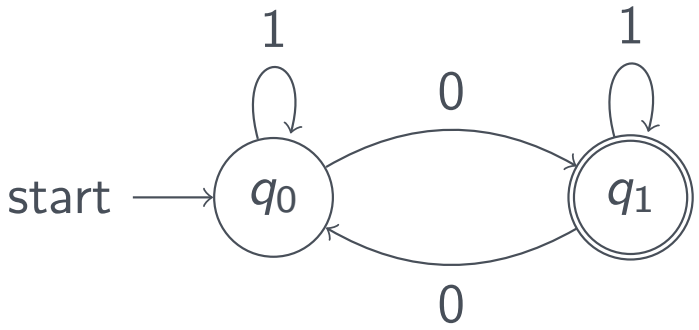
- Se il linguaggio è regolare, possiamo trovare un DFA che lo riconosce
- Idea: trasformiamo il DFA in grammatica:
  - una variabile R_i per ogni stato q_i
  - una regola $R_i \to aR_j$ per ogni transizione $\delta(q_i, a) = q_j$
  - una regola $R_i \to \varepsilon$ per ogni stato finale q_i
  - R_0 variabile iniziale, se q_0 è lo stato iniziale

Esempio

$\{w \in \{0,1\}^* \mid w \text{ contiene un numero dispari di } 0\}$

(2) Trasformare un DFA in CFG

- DFA per $\{w \in \{0,1\}^* \mid w \text{ contiene un numero dispari di } 0\}$:



- Grammatica context-free:

$$R_0 \to 0R_1 \mid 1R_0$$
$$R_1 \to 0R_0 \mid 1R_1 \mid \varepsilon$$

(3) Sottostringhe collegate

- Le parole del linguaggio possono essere formate da due sottostringhe collegate tra di loro:
  - una macchina per il linguaggio dovrebbe memorizzare dell'informazione su una sottostringa e poi verificare la corrispondenza con l'altra sottostringa
- Esempio: per riconoscere $\{0^n 1^n \mid n \geq 0\}$ bisogna contare gli 0 e verificare che siano in numero uguale agli 1
- Idea: regole nella forma $R \to uRv$ generano stringhe dove u corrisponde a v
- La grammatica $S \to 0S1 \mid \varepsilon$ garantisce che il numero di 0 sia uguale al numero di 1

Strutture Ricorsive

- Le stringhe possono contenere strutture che compaiono ricorsivamente come parte di altre strutture
- Idea: porre nelle regole la variabile che genera la struttura nei punti dove la struttura può comparire ricorsivamente
- Esempio: nella grammatica per le espressioni aritmetiche, possiamo sostituire a con una intera espressione aritmetica parentesizzata
  - La regola $\langle FACTOR \rangle \rightarrow (\langle EXPR \rangle) \mid a$ ci dice che possiamo fare questa sostituzione.