

# Linguaggi di Programmazione

## Corso C

### Parte n.4

## Linguaggi: Gerarchia ed Operazioni

Nicola Fanizzi ([fanizzi@di.uniba.it](mailto:fanizzi@di.uniba.it))

**Dipartimento di Informatica**  
**Università degli Studi di Bari**

# Gerarchia di Chomsky

Sia  $G = (X, V, S, P)$  una grammatica. A seconda della forma delle regole di produzione  $P$  si distinguono le seguenti classi di grammatiche:

**Tipo 0** nessuna limitazione:  $\alpha \rightarrow \beta$  dove  $\alpha \in (X \cup V)^+$  contiene almeno un non terminale e  $\beta \in (X \cup V)^*$ ;

**Tipo 1 dipendente da contesto** (o context-sensitive) con produzioni

1.  $yAz \rightarrow ywz$  con  $A \in V$ ,  $y, z \in (X \cup V)^*$  e  $w \in (X \cup V)^+$ ;
2.  $S \rightarrow \lambda$  purché  $S$  non compaia a destra di alcuna produzione;

**Tipo 2 libera da contesto** (o context-free) con produzioni

$$v \rightarrow w, v \in V;$$

**Tipo 3 lineare destra** con produzioni

1.  $A \rightarrow bC$  con  $A, C \in V$ ,  $b \in X$
2.  $A \rightarrow d$  con  $A \in V$ ,  $d \in X \cup \{\lambda\}$ .

I linguaggi generati dalle grammatiche ne mutuano i tipi.

# Esempi

## 0. a forma di frase

- $(X, V \cup \{A, B\}, S, P \cup \{AB \rightarrow BA\})$

## 1. dipendenti da contesto

- linguaggi monotoni

## 2. liberi

- linguaggio delle parentesi ben formate, linguaggio dei numeri interi relativi, linguaggio delle stringhe con un numero uguale di 0 e 1
- linguaggi  $L_1 = \{a^n b^n \mid n > 0\}$  e  $L_2 = \{a^k b^n c^{2k} \mid n, k > 0\}$
- molti linguaggi di programmazione

## 3. lineari

- linguaggi finiti
- linguaggio delle stringhe binarie  $\{0, 1\}^*$
- linguaggio delle stringhe su  $\{0, 1\}$  con un numero pari di 1

# Teorema della Gerarchia

Dato un alfabeto  $X$

denotate le classi di linguaggi nel seguente modo

$$\mathcal{L}_i = \{L \subseteq X^* \mid L = L(G), G \text{ grammatica di tipo } i\}$$

risulta:

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$$

**Dim.**

$\mathcal{L}_3 \subset \mathcal{L}_2$  Si osservi che  $\mathcal{L}_3 \subseteq \mathcal{L}_2$  per definizione, occorre solo provare che l'inclusione sia stretta usando un linguaggio separatore per es.  $L = \{a^k b^k \mid k > 0\}$

$\mathcal{L}_2 \subset \mathcal{L}_1$  Si osservi che  $\mathcal{L}_2 \subseteq \mathcal{L}_1$  data la forma delle produzioni, tranne nel caso di  $\lambda$ -produzioni del tipo:  $A \longrightarrow \lambda$  con  $A \neq S$ .

In tal caso, se  $\lambda$  è nel linguaggio considerato,

per il lemma della stringa vuota esiste una grammatica equivalente che fa produrre  $\lambda$  direttamente da  $S$ .

Altrimenti basta eliminare le  $\lambda$ -produzioni.

Infine si può usare come linguaggio separatore  $L = \{a^k b^k c^k \mid k > 0\}$

$\mathcal{L}_1 \subset \mathcal{L}_0$  Per definizione  $\mathcal{L}_1 \subseteq \mathcal{L}_0$ .

Data una qualunque grammatica di tipo 1  $G = (X, V, S, P)$ , si consideri allora la grammatica  $(X, V \cup \{A, B\}, S, P \cup \{AB \longrightarrow BA\})$  che genera un linguaggio separatore di tipo 0.

# Operazioni sui Linguaggi

Dati due linguaggi  $L$  e  $L'$  definiti sullo stesso alfabeto  $X$ :

## unione

$$L \cup L' = \{w \in X^* \mid w \in L \vee w \in L'\}$$

## prodotto

$$L \cdot L' = \{w = w_1 \cdot w_2 \in X^* \mid w_1 \in L \wedge w_2 \in L'\}$$

## iterazione

$$L^* = \{w = w_1 w_2 \dots w_n \in X^* \mid \forall n \forall i : w_i \in L, 0 \leq i \leq n\}$$

## complemento

$$\bar{L} = \{w \in X^* \mid w \notin L\}$$

## intersezione

$$L \cap L' = \{w \in X^* \mid w \in L \wedge w \in L'\}$$

## potenza

$$L^k = \begin{cases} \{\lambda\} & k=0 \\ L^{k-1} \cdot L & k>0 \end{cases}$$

## chiusura transitiva

$$L^+ = \bigcup_{k>0} L^k \quad (\text{quindi } L^* = L^0 \cup L^+ = \{\lambda\} \cup L^+)$$

# Proprietá delle Operazioni sui Linguaggi

Dati i linguaggi  $L, L', L''$

definiti sullo stesso alfabeto  $X$ :

1.  $(L \cdot L') \cdot L'' = L \cdot (L' \cdot L'')$  proprietá associativa

2.  $L \cdot L' \neq L' \cdot L$  non commutativitá

3.  $L \cdot \{\lambda\} = \{\lambda\} \cdot L = L$  elemento neutro

Quindi anche  $(\wp(X^*), \cdot)$  è un monoide.

4.  $L \cdot \emptyset = \emptyset \cdot L = \emptyset$  elemento assorbente

5. Se  $\lambda \in L$ :  $L' \subseteq L \cdot L'$

$$L' \subseteq L' \cdot L$$

6. Se  $\lambda \in L'$ :  $L \subseteq L \cdot L'$

$$L \subseteq L' \cdot L$$

## Esempi

Dati i linguaggi  $L_1 = \{a^{2n} | n \geq 0\}$  e  $L_2 = \{b, cc\}$

- $L_1 \cdot L_2 = \{b, cc, aab, aacc, aaaab, aaaacc, \dots\}$
- $L_2 \cdot L_1 = \{b, cc, baa, ccaa, baaaa, ccaaaa, \dots\}$

è verificata quindi la proprietà 5)

mentre non vale la 6).

Inoltre:

- $L_1 \cup L_2 = L_2 \cup L_1 = \{\lambda, b, cc, aa, aaaa, aaaaaa, \dots\}$
- $L_1^* = \{\lambda, aa, aaaa, aaaaaa, aaaaaaaa, \dots\}$
- $L_2^* = \{\lambda, b, cc, bb, bcc, ccb, bbb, cccc, \dots\}$
- $L_2^0 = \{\lambda\}$   
 $L_2^1 = \{b, cc\}$   
 $L_2^2 = \{bb, bcc, ccb, cccc\} \dots$
- $L_2^+ = \{b, cc, bb, bcc, ccb, bbb, cccc, \dots\}$

# Chiusura rispetto alle Operazioni

Sia  $\mathcal{L}$  una classe di linguaggi su  $X$ .

- **operazione unaria  $\Delta$ :**

$$\wp(X^*) \longrightarrow \wp(X^*)$$

$$L \mapsto \Delta(L)$$

$\mathcal{L}$  è **chiusa** rispetto a  $\Delta$  sse

$$\forall L \in \mathcal{L} : \Delta(L) \in \mathcal{L}$$

- **operazione binaria  $\square$ :**

$$\wp(X^*) \times \wp(X^*) \longrightarrow \wp(X^*)$$

$$(L_1, L_2) \mapsto \square(L_1, L_2)$$

$\mathcal{L}$  è **chiusa** rispetto a  $\square$  sse

$$\forall L_1, L_2 \in \mathcal{L} : \square(L_1, L_2) \in \mathcal{L}$$

## Schema di dimostrazione

Dati i linguaggi  $L_1$  e  $L_2$  generati da

$$G_1 = (X, V_1, S_1, P_1) \text{ e } G_2 = (X, V_2, S_2, P_2)$$

assumiamo che:  $V_1 \cap V_2$  e  $S \notin V_1 \cap V_2$

allora lo schema di dimostrazione è il seguente:

- considerare una operazione  $\square$
- costruire  $G$  date  $G_1$  e  $G_2$ ;
- dimostrare che se  $G_1$  e  $G_2$  sono di tipo  $i$  allora anche  $G$  è di tipo  $i$ ;
- dimostrare che  $L(G) = \square(L_1, L_2)$  quindi  $\mathcal{L}$  è chiusa rispetto all'operazione  $\square$

Analogamente per le operazioni unarie:

- Considerata  $G$  costruire una grammatica  $G'$ ;
- Dimostrare che se  $G$  è di tipo  $i$  allora anche  $G'$  è di tipo  $i$ ;
- dimostrare che  $L(G') = \Delta(L(G))$  quindi  $\mathcal{L}$  è chiusa rispetto all'operazione  $\Delta$

# Teorema di Chiusura rispetto all'Unione

Le classi dei linguaggi  $\mathcal{L}_i$ ,  $i = 0, 1, 2, 3$  sono chiuse rispetto all'unione.

**Dim.**

Sia  $G = (X, V, S, P)$  ove:

$$S \notin V_1 \cup V_2$$

$$V_1 \cap V_2 = \emptyset$$

$$V = V_1 \cup V_2 \cup \{S\}$$

$$P = \{S \longrightarrow S_1 | S_2\} \cup P_1 \cup P_2$$

Questo vale per  $\mathcal{L}_i$ ,  $i = 0, 1, 2$ .

Nel caso di  $\mathcal{L}_3$

consideriamo la grammatica lineare

$G = (X, V, S, P)$  ove:

$$\begin{aligned} P = & \{S \longrightarrow w | S_1 \longrightarrow w \in P_1\} \cup \\ & \cup \{S \longrightarrow w | S_2 \longrightarrow w \in P_2\} \cup \\ & \cup P_1 \cup P_2 \end{aligned}$$

## Esempio

Date le grammatiche

$G_1$  con  $P_1 = \{S_1 \longrightarrow aA, A \longrightarrow b\}$  e

$G_2$  con  $P_2 = \{S_2 \longrightarrow cC, C \longrightarrow b\}$ .

Si osservi che  $V_1 \cap V_2 = \emptyset$ ,  $X = \{a, b, c\}$

Si consideri  $G = (X, V_1 \cup V_2 \cup \{S\}, S, P)$

$$P = \{S \longrightarrow aA, S \longrightarrow cC, \\ S_1 \longrightarrow aA, A \longrightarrow b, \\ S_2 \longrightarrow cC, C \longrightarrow b\}$$

E' facile vedere che

$$L(G) = L(G_1) \cup L(G_2) = \{ab, cb\}$$

Si osservi che le produzioni

$$S_1 \longrightarrow aA \text{ e } S_2 \longrightarrow cC$$

diventano inutili e possono essere eliminate:

$$G = (X, \\ V_1 \cup V_2 \cup \{S\}, \\ S, \\ \{S \longrightarrow aA \mid cC, A \longrightarrow b, C \longrightarrow b\})$$

# Teorema di Chiusura rispetto al Prodotto

Le classi dei linguaggi  $\mathcal{L}_i$ ,  $i = 0, 1, 2, 3$  sono chiuse rispetto al prodotto.

**Dim.** Sia  $G = (X, V, S, P)$  ove:

$$P = \{S \longrightarrow S_1 \cdot S_2\} \cup P_1 \cup P_2.$$

Se  $G_1, G_2$  sono di tipo  $i = 0, 1, 2$  anche  $G$  è di tipo  $i$

Tuttavia questo funziona solo per  $\mathcal{L}_2$  (linguaggi liberi)  
per i problemi di interferenza dei contesti:

---

**Es.**

$G_1$  con  $P_1 = \{S_1 \longrightarrow b\}$  e

$G_2$  con  $P_1 = \{bS_2 \longrightarrow bb\}$

per cui

$L(G_1) = \{b\}$  e  $L(G_2) = \emptyset$  e quindi

$$L(G_1 \cdot G_2) = L(G_1) \cdot L(G_2) = \emptyset$$

Se fosse

$$\begin{aligned} P &= \{S \longrightarrow S_1 \cdot S_2\} \cup P_1 \cup P_2 = \\ &= \{S \longrightarrow S_1 \cdot S_2, S_1 \longrightarrow b, bS_2 \longrightarrow bb\} \end{aligned}$$

allora  $S \Longrightarrow S_1 S_2 \Longrightarrow bS_2 \Longrightarrow bb$

quindi  $S \xrightarrow{*} bb$  cioè  $bb \in L(G) \neq \emptyset$

---

## Teorema di Chiusura rispetto al Prodotto (II)

Per tenere distinte le derivazioni considero nuovi non terminali disgiunti:

$$X' = \{x' | x \in X\} \text{ e}$$

$$X'' = \{x'' | x \in X\} \text{ tali che:}$$

- $V_1 \cap X' = \emptyset$
- $V_2 \cap X'' = \emptyset$
- $X' \cap X'' = \emptyset$

Quindi si trasformano le produzioni in  $P_1$  e  $P_2$ , sostituendo ogni terminale  $x$  con un nuovo non terminale  $x'$ , risp.  $x''$ ; infine si aggiungono le produzioni dei terminali  $x_i$  dai nuovi non terminali  $x'_i$ , risp.  $x''_i$ :

$$\begin{aligned} P = & \{S \longrightarrow S_1 S_2\} \cup \\ & \cup P_1[x'/x] \cup P_2[x''/x] \cup \\ & \cup \{x' \longrightarrow x | x \in X\} \cup \{x'' \longrightarrow x | x \in X\} \end{aligned}$$

---

### Es. precedente

$$\begin{aligned} P = & \{S \longrightarrow S_1 \cdot S_2, S_1 \longrightarrow b', b'' S_2 \longrightarrow b'' b'', \\ & b' \longrightarrow b, b'' \longrightarrow b\} \end{aligned}$$

Allora  $S \not\stackrel{*}{\Rightarrow} bb$  essendo:  $S \Rightarrow S_1 S_2 \Rightarrow b' S_2$ .

Nessuna derivazione di stringhe di terminali è possibile.  
Quindi ora  $L(G) = \emptyset$ .

---

# Teorema di Chiusura rispetto al Prodotto (III)

Nel caso della classe  $\mathcal{L}_3$ ,

non potendo usare produzioni  $S \longrightarrow S_1S_2$ ,

si sfrutta il fatto che l'unico (al piú) simbolo non terminale compare come simbolo piú a destra in una produzione

Perció si sostituisce ogni produzione  $A \longrightarrow b$  in  $G_1$  in modo da far cominciare la derivazione da  $S_2$ :

$G = (X, V \setminus \{S\}, S_1, P)$  con

$$P = \{A \longrightarrow bB \mid A \longrightarrow bB \in P_1\} \cup$$

$$\cup \{A \longrightarrow bS_2 \mid A \longrightarrow b \in P_1, b \neq \lambda\} \cup$$

$$\cup \{A \longrightarrow bS_2 \mid A \longrightarrow bB \in P_1, B \longrightarrow \lambda \in P_1\} \cup$$

$$\cup \{S_1 \longrightarrow w \mid S_2 \longrightarrow w \in P_2, S_1 \longrightarrow \lambda \in P_1\} \cup$$

$$\cup P_2$$

# Teorema di Chiusura rispetto all'Iterazione

Le classi dei linguaggi  $\mathcal{L}_i$ ,  $i = 0, 1, 2, 3$  sono chiuse rispetto all'iterazione.

**Dim.**

Considerata  $G_1$  sia  $G = (X, V_1 \cup \{S\}, S, P)$  dove:

$$P = \{S \longrightarrow \lambda, S \longrightarrow S_1 S\} \cup P_1$$

Si osservi che questo metodo vale per la classe  $\mathcal{L}_2$

1. per le classi  $\mathcal{L}_0$  e  $\mathcal{L}_1$   
si ripresenta il problema dell'interferenza del contesto
2. le nuove produzioni non possono essere usate per la classe  $\mathcal{L}_3$

# Teorema di Chiusura rispetto all'iterazione (I)

Per risolvere il caso 1.:

- Si eliminano le produzioni  $S_1 \longrightarrow \lambda$   
(per il Lemma della stringa vuota)

- Si considerano gli insiemi

$$X' = \{x' | x \in X_1\} \text{ e } X'' = \{x'' | x \in X_1\}$$

$$\text{con } V_1 \cap X' = \emptyset, V_1 \cap X'' = \emptyset \text{ e } X' \cap X'' = \emptyset$$

- costruendo due copie di  $G_1$

$$G'_1 = (X, V_1 \cup X' \cup \{S_1\}, S_1, P'_1) \text{ e}$$

$$G''_1 = (X, V_1 \cup X'' \cup \{S_2\}, S_2, P''_1)$$

con ins. di produzioni

$$P'_1 = P_1[x'/x] \text{ e } P''_1 = P_1[x''/x]$$

- Si ottiene

$$G = \{X, V_1 \cup X' \cup X'' \cup \{S_1, S'_1, S_2, S'_2, S\}, S, P\}$$

$$\text{dove } P = \{S \longrightarrow \lambda | S'_1 | S'_2,$$

$$S'_1 \longrightarrow S_1 | S_1 S'_2,$$

$$S'_2 \longrightarrow S_2 | S_2 S'_1\} \cup$$

$$\cup P'_1 \cup P''_1 \cup$$

$$\cup \{x' \longrightarrow x | x \in X\} \cup \{x'' \longrightarrow x | x \in X\}$$

# Teorema di Chiusura rispetto all'Iterazione (II)

Per risolvere il caso 2. si costruisce  $G = \{X, V_1 \cup \{S\}, S, P\}$  dove:

- si introduce la produzione  $S \longrightarrow \lambda$  eliminando l'eventuale  $S_1 \longrightarrow \lambda \in P_1$
- a  $P$  si aggiunge  $S \longrightarrow w$  per ogni  $S_1 \longrightarrow w \in P_1$
- se nell'insieme in  $P$  costruzione, esiste una produzione  $A \longrightarrow b$  o una  $A \longrightarrow bB$  (quando  $B \longrightarrow \lambda \in P$ ), allora si aggiunge a  $P$  anche  $A \longrightarrow bS$
- ottenendo (in forma ricorsiva):

$$\begin{aligned} P = & \{S \longrightarrow \lambda\} \cup (P_1 \setminus \{S_1 \longrightarrow \lambda\}) \cup \\ & \cup \{S \longrightarrow w \mid S_1 \longrightarrow w \in P_1\} \cup \\ & \cup \{A \longrightarrow bS \mid A \longrightarrow b \in P \wedge b \neq \lambda\} \cup \\ & \cup \{A \longrightarrow bS \mid A \longrightarrow bB \in P \wedge B \longrightarrow \lambda \in P\} \end{aligned}$$

## Altri Teoremi di Chiusura

1. La classe dei linguaggi lineari destri  $\mathcal{L}_3$  è chiusa rispetto al complemento e all'intersezione.
2. La classe dei linguaggi non contestuali  $\mathcal{L}_2$  non è chiusa rispetto al complemento e all'intersezione.
3. La classe dei linguaggi contestuali  $\mathcal{L}_1$  è chiusa rispetto al complemento e all'intersezione.
4. La classe dei linguaggi  $\mathcal{L}_0$  non è chiusa rispetto al complemento.

**Dim.**

Si sfrutta la legge di De Morgan:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Per la 2. si considerino i linguaggi

$$L_1 = \{a^n b^n c^m \mid n, m > 0\} \text{ e } L_2 = \{a^n b^m c^m \mid n, m > 0\}$$

la cui intersezione è

$$L_1 \cap L_2 = \{a^n b^n c^n \mid n > 0\}$$

# Operatore di Riflessione

- Data una parola  $w = x_1x_2 \cdots x_n$ ,  
con  $x_i \in X \quad \forall i = 1, \dots, n$   
dicesi **stringa riflessa** (o *riflessione*) di  $w$   
la stringa  $w^R = x_nx_{n-1} \cdots x_1$
- Questo definisce un'operazione unaria

$$(\cdot)^R : X^* \longrightarrow X^*$$

- Un palindromo è una parola  $w \in X^*$  tale che:

$$w = w^R$$

## Teorema.

Sia  $w$  una stringa su  $X$ .

Allora  $w$  è palindroma sse  $w = \alpha x \alpha^R$

con  $x \in X \cup \{\lambda\}$

## Teorema.

La classe dei linguaggi  $\mathcal{L}_2$

è chiusa rispetto a riflessione.

# Esercizi

(da risolvere tramite le proprietà di chiusura)

1. Dimostrare che  $L = \{a^n b^n c^m \mid n, m > 0\}$  è non contestuale (libero)
2. Dati  $L_1 = \{a^n b^n \mid n \geq 0\}$  e  $L_2 = \{a\}^* \cdot \{bb\}^*$ , dimostrare che  $L = L_1 \cap L_2$  è non contestuale
3. Utilizzare la proprietà di chiusura di  $\mathcal{L}_2$  rispetto a  $\cup$  per dimostrare che i seguenti linguaggi sono liberi:
  - (a)  $L = \{a^i b^j \mid i \neq j, i, j \geq 0\}$
  - (b)  $L = \{w \in \{a, b\}^* \mid w = w^R\}$
  - (c)  $L = \{a, b\}^* \setminus \{a^i b^i \mid i \geq 0\}$
4. Dimostrare che  $L = \{a^n b^m \mid n \neq m, n, m > 0\}$  non è lineare destro

**Esercizio** Dimostrare che  $L = \{a^n b^n c^m \mid n, m > 0\}$  è libero e trovare una grammatica che lo generi

---

$L$  può essere scritto come prodotto di linguaggi:

$L_1 = \{a^n b^n \mid n > 0\}$  è un linguaggio libero e

$L_2 = \{c^m \mid m > 0\} = \{c\}^+ = \{c\}^* \setminus \{\lambda\}$ , linguaggio lineare (e quindi anche libero  $\mathcal{L}_3 \subsetneq \mathcal{L}_2$ )

$L = L_1 \cdot L_2$  deve essere libero per la chiusura

$G_1 = (X_1, V_1, S_1, P_1)$  con

- $X_1 = \{a, b\}$        $V_1 = \{S_1\}$
- $P_1 = \{S_1 \longrightarrow aS_1b, S_1 \longrightarrow ab\}$

$G_2 = (X_2, V_2, S_2, P_2)$  con

- $X_2 = \{c\}$        $V_2 = \{S_2\}$
- $P_2 = \{S_2 \longrightarrow cS_2, S_2 \longrightarrow c\}$

Quindi:  $G = (X, V, S, P)$

$X = X_1 \cup X_2 = \{a, b, c\}$        $V = V_1 \cup V_2 \cup \{S\} = \{S, S_1, S_2\}$

$$\begin{aligned} P &= \{S \longrightarrow S_1 S_2\} \cup P_1 \cup P_2 = \\ &= \{S \longrightarrow S_1 S_2\} \cup \\ &\cup \{S_1 \longrightarrow aS_1b, S_1 \longrightarrow ab\} \cup \\ &\cup \{S_2 \longrightarrow cS_2, S_2 \longrightarrow c\} \cup \end{aligned}$$