

## 3 - Linguaggi liberi e dipendenti da contesto

### Grammatica libera da contesto

Una grammatica  $G = (X, V, S, P)$  è **libera da contesto** (o **context-free - C.F.**) se, per ogni produzione,  $v \rightarrow w$ ,  $v$  è un non terminale.

Definito anche in:

$$G \text{ è libera da contesto} \iff \forall v \rightarrow w \in P : v \in V$$

### Linguaggio libero da contesto

Un linguaggio  $L$  su un alfabeto  $X$  è libero da contesto se può essere generato da una grammatica libera da contesto

Definito anche in

$$L \text{ libero da contesto} \iff \exists G \text{ libera da contesto tale che } L(G) = L$$

Se si ha una grammatica C.F. che genera  $L$ , non è detto che non esista un'altra grammatica che generi lo stesso linguaggio

La maggior parte dei linguaggi di programmazione ricade nella classe dei C.F., termine che nasce dal fatto che la sostituzione di un  $NT$  non è condizionata dal contesto (caratteri adiacenti) da cui compare;

Un simbolo  $NT$  in una forma di frase può essere sempre sostituito usando una produzione  $A \rightarrow \beta$ , questa sostituzione è sempre valida;

Viceversa, se  $L = L(G)$  (se il linguaggio è generato dalla grammatica) e  $G$  non è C.F., non possiamo concludere che  $L$  **non** è C.F., poiché non si può escludere che esista una grammatica C.F.  $G'$  per cui  $L = L(G')$

Quindi la **grammatica** può essere non context-free, ma il **linguaggio** potrebbe comunque essere context-free se esiste almeno una grammatica context-free che lo genera.

### Linguaggi dipendenti da contesto

Una grammatica  $G = (X, V, S, P)$  è **dipendente da contesto** se ogni produzione in  $P$  è in una delle seguenti forme:

- **Produzione contestuale:**

$$yAz \rightarrow ywz$$

con:

- $A$  è un simbolo non terminale ( $A \in V$ ),
- $y$  e  $z$  sono stringhe di simboli terminali e non terminali  $y, z \in (X \cup V)^*$ ,

- $w$  è una stringa non vuota di simboli terminali e non terminali  $w \in (X \cup V)^+$ .

Può essere letta anche come:

"Il simbolo non terminale  $A$  può essere sostituito con  $w$  solo se è circondato dal contesto sinistro  $y$  e dal contesto destro  $z$ ."

- **Produzione speciale per la stringa vuota:**

$$S \rightarrow \lambda$$

dove  $S$  è il simbolo iniziale della grammatica e  $\lambda$  rappresenta la stringa vuota.

**Questa produzione è ammessa solo se  $S$  non compare nella parte destra di alcuna produzione.**

Quindi un linguaggio  $L$  è dipendente da contesto se **può essere generato da una grammatica dipendente da contesto**

## Relazione tra C.F e C.S. con eccezione

La relazione tra i **linguaggi liberi da contesto (C.F.)** e i **linguaggi dipendenti da contesto (C.S.)** è basata sulla gerarchia dei linguaggi formali, che classifica i linguaggi in base alla complessità delle grammatiche che li generano. In particolare, i linguaggi liberi da contesto sono un **sottoinsieme** dei linguaggi dipendenti da contesto, ma c'è un'importante **eccezione** che riguarda le produzioni che generano la stringa vuota ( $\lambda$ ).

Ogni grammatica libera da contesto è un caso particolare di grammatica dipendente da contesto, questo perché le produzioni delle grammatiche C.F. possono essere viste come un caso speciale delle produzioni C.S. in cui il contesto è vuoto (cioè  $y = z = \lambda$ ).

In altre parole, una produzione C.F.  $A \rightarrow w$  può essere vista come una produzione C.S.  $\lambda A \lambda \rightarrow \lambda w \lambda$ , dove il contesto è vuoto, pertanto **tutti i linguaggi liberi da contesto sono anche linguaggi dipendenti da contesto.**

## Grammatica monotona

Una grammatica  $G = (X, V, S, P)$  si dice **monotona** se tutte le sue produzioni  $v \rightarrow w$  soddisfano la condizione:

$$|v| \leq |w|$$

Definita anche con:

$$\forall v \rightarrow \in P : |v| \leq |w|$$

La grammatica monotona comprende due caratteristiche principali:

1. **Non contrazione:** Le produzioni non possono accorciare la stringa derivata (non si può avere  $|v| > |w|$ ).
2. **Eccezione per la stringa vuota:**
  - L'unica produzione che può ridurre la lunghezza è  $S \rightarrow \lambda$ , dove  $S$  è il simbolo iniziale.
  - Se presente,  $S$  **non deve comparire nella parte destra** di altre produzioni.

## Linguaggio monotono

Un linguaggio  $L$  si dice **monotono** se esiste una grammatica monotona  $G$  che lo genera, cioè:

$$L = L(G)$$

**Proprietà:**

- **Equivalenza con i linguaggi dipendenti da contesto (C.S.):** La classe dei linguaggi monotoni **coincide** con quella dei linguaggi dipendenti da contesto, ogni grammatica monotona può essere convertita in una grammatica C.S. equivalente e viceversa.

I linguaggi C.F. sono un sottoinsieme dei linguaggi monotoni/C.S.

**Esempio:**  $a^n b^n \mid n \geq 0$  è C.F. e monotono, mentre  $a^n b^n c^n \mid n \geq 1$  è monotono ma **non** C.F.

**Esempi di produzioni monotone a produzioni contestuali:**

1. Produzione monotona:

$$AB \rightarrow CDEF$$

$AB \rightarrow CDEF$  può essere sostituita con produzioni contestuali come

$$AB \rightarrow AG, AG \rightarrow CG, CG \rightarrow CDEF$$

2. Produzione monotona

$$CB \rightarrow BC$$

Può essere sostituita con  $CB \rightarrow XB, XB \rightarrow XC, XC \rightarrow BC$

## Proposizione

La classe dei linguaggi dipendenti da contesto (C.S.) coincide con la classe dei linguaggi monotoni.

Da questa proposizione possiamo derivare il teorema:

## Teorema

Per ogni grammatica monotona  $G$ , esiste una grammatica C.S.  $G'$  equivalente, cioè tale che  $L(G) = L(G')$ .

Definita anche come:

$$L \text{ è C.S. } \iff \exists G \text{ C.S. : } L = L(G)$$

Da questo teorema possiamo determinare una riformulazione di questo, riscritta come

Un linguaggio  $L$  è dipendente da contesto se e solo se esiste una grammatica  $G$  con produzioni  $u \rightarrow v$  tali che:

$$0 < |u| \leq |v|,$$

## Dimostrazioni

## Parte 1 (Diretta)

Se  $L$  è C.S., per definizione esiste una grammatica  $G$  C.S. con produzioni di due tipi:

1. **Contestuali:**  $yAz \rightarrow ywz$  con  $A \in V$ ,  $y, z \in (X \cup V)^*$ ,  $w \in (X \cup V)^+$ .

Queste soddisfano  $0 < |yAz| \leq |ywz|$  perché  $|w| \geq 1$ .

2. **Lambda-produzione:**  $S \rightarrow \lambda$ , con  $S$  non a destra in altre produzioni.

Ricade nell'eccezione.

## Parte 2 (Inversa)

Data una grammatica  $G$  con produzioni  $u \rightarrow v$  e  $0 < |u| \leq |v|$ , possiamo trasformarla in una grammatica C.S. equivalente  $G'$ .

### Ipotesi

Sia  $G$  una grammatica monotona con produzioni nella forma  $u \rightarrow v$ , dove  $0 < |u| \leq |v|$ , eccetto eventualmente  $S \rightarrow \lambda$  (con  $S$  non a destra in altre produzioni).

### Tesi

Esiste una grammatica  $G'$  contestuale (C.S.) equivalente a  $G$ , cioè tale che  $L(G) = L(G')$ .

Ogni produzione monotona:

$$A_1 A_2 \dots A_m \rightarrow B_1 B_2 \dots B_n$$

(con  $m \leq n$ ) può essere sostituita da una sequenza di produzioni contestuali.

Introduciamo  $m$  nuovi non terminali  $C_1, C_2, \dots, C_m$ , non presenti in  $G$ .

La produzione originale viene "scomposta" in  $2m$  passi contestuali (o produzioni):

### 1. Fase di marcatura

Sostituiamo progressivamente i simboli  $A_i$  con i nuovi  $C_i$ :

$$\begin{cases} A_1 A_2 \dots A_m \rightarrow C_1 A_2 \dots A_m \\ C_1 A_2 \dots A_m \rightarrow C_1 C_2 A_3 \dots A_m \\ \vdots \\ C_1 C_2 \dots C_{m-1} A_m \rightarrow C_1 C_2 \dots C_m B_{m+1} \dots B_n \end{cases}$$

(Ogni passo rispetta il contesto  $y_z$ , dove  $y$  e  $z$  sono prefissi/suffissi fissati.)

### 2. Fase di sostituzione

Sostituiamo i  $C_i$  con i simboli  $B_i$  desiderati:

$$\begin{cases} C_1 C_2 \dots C_m B_{m+1} \dots B_n \rightarrow C_1 \dots C_{m-1} B_m B_{m+1} \dots B_n \\ \vdots \\ C_1 B_2 \dots B_n \rightarrow B_1 B_2 \dots B_n \end{cases}$$

### Esempio di Trasformazione:

Per una produzione monotona  $ABC \rightarrow DEFGH$  ( $m = 3$ ,  $n = 5$ ), si introducono  $m$  nuovi non terminali  $C_1, C_2, C_3$  e si creano 6 produzioni contestuali:

1.  $ABC \rightarrow C_1 BC$
2.  $C_1 BC \rightarrow C_1 C_2 C$
3.  $C_1 C_2 C \rightarrow C_1 C_2 C_3 GH$

$$4. C_1C_2C_3GH \rightarrow C_1C_2FGH$$

$$5. C_1C_2FGH \rightarrow C_1EFGH$$

$$6. C_1EFGH \rightarrow DEFGH$$

Ogni passo sostituisce un simbolo alla volta, mantenendo la condizione  $|u| \leq |v|$ . La grammatica risultante  $G'$  è C.S. e genera lo stesso linguaggio di  $G$ .