Linguaggi di Programmazione
Docente: Cataldo Musto

Capitolo 3 – Linguaggi liberi da contesto e linguaggi dipendenti da contesto

Si ringraziano il Prof. Giovanni Semeraro, il Prof. Marco de Gemmis e il Prof. Pasquale Lops per la concessione del materiale didattico di base

Classificazione delle grammatiche secondo Chomsky

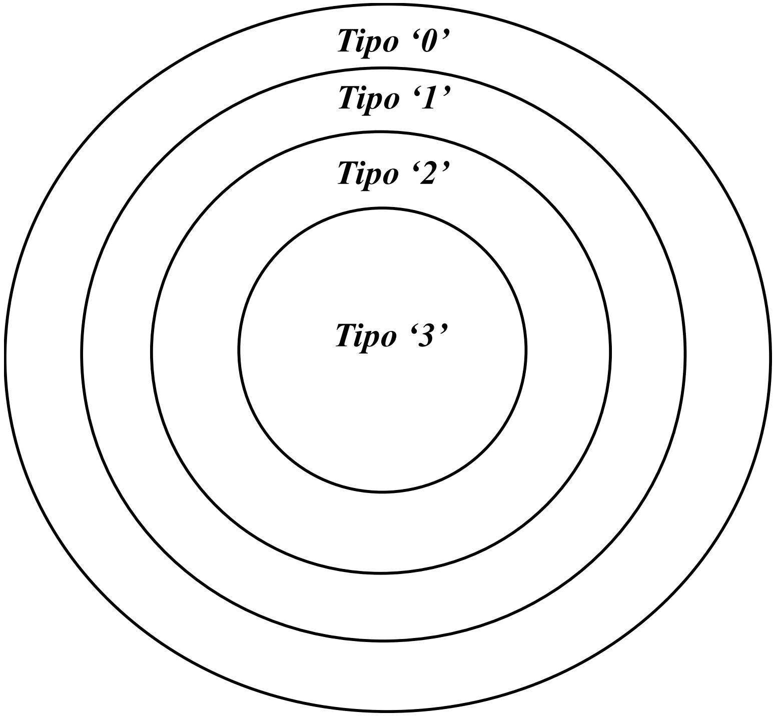
- Definizione: Gerarchia di Chomsky (1956-1959)
  - Sia $G = (X, V, S, P)$ una grammatica. Dalla definizione di grammatica, si ha:

$$P = \left\{ v \rightarrow w \mid v \in (X \cup V)^{+} \text{e } \ v \,\text{contiene almeno un}\, NT, \ \ w \in (X \cup V)^{*} \right\}$$

A seconda delle restrizioni imposte sulle regole di produzione, si distinguono le varie classi di grammatiche.

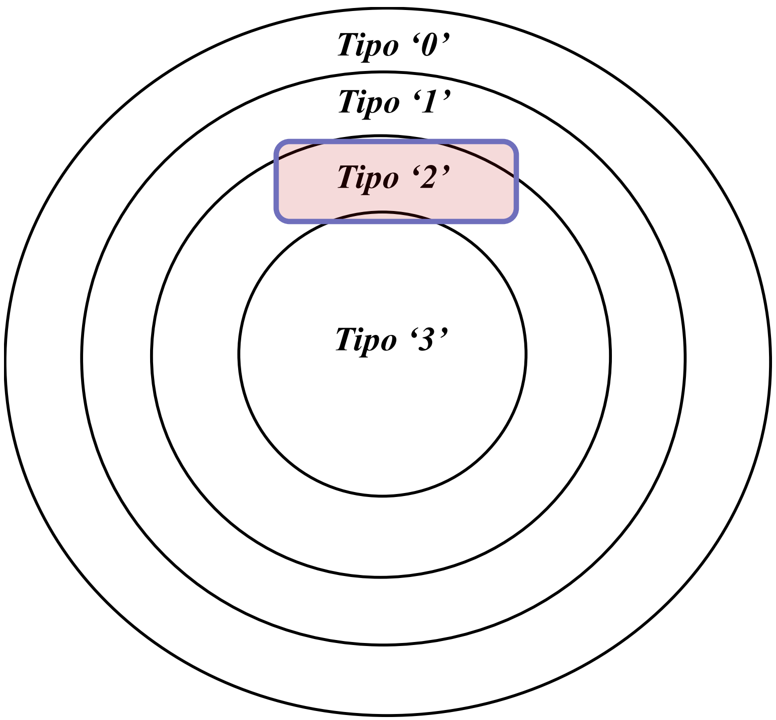
Classificazione delle grammatiche secondo Chomsky

- Chomsky definisce quattro classi di linguaggi



Classificazione delle grammatiche secondo Chomsky

- Definizione: Gerarchia di Chomsky (1956-1959)
  - I linguaggi studiati finora sono tutti linguaggi di tipo ‘2’ (anche detti **liberi da contesto**)



Definizione di 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 nonterminale.

$$G \text{ è libera da contesto } \overset{def}{\Leftrightarrow} \forall\, v \rightarrow w \in P : v \in V$$

Definizione di grammatica libera da contesto

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

$$G \text{ è libera da contesto } \overset{def}{\Leftrightarrow} \forall\, v \to w \in P : v \in V$$

- Esempi di grammatiche libere da contesto

$$S \to \overset{(1)}{A} \,|\, \overset{(2)}{B}, \quad A \to \overset{(3)}{aA} \,|\, \overset{(4)}{a}, \quad B \to \overset{(5)}{bB} \,|\, \overset{(6)}{b}$$

$$X = \{a, b\}, \quad V = \{S\}, \quad P = \left\{ S \overset{(1)}{\to} aSb, S \overset{(2)}{\to} ab \right\}$$

$$S \to +I \,|\, -I$$

$$I \to D \,|\, I\,D$$

$$D \to 0 \,|\, 1 \,|\, 2 \,|\, 3 \,|\, 4 \,|\, 5 \,|\, 6 \,|\, 7 \,|\, 8 \,|\, 9$$

Definizione di 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.

  L libero da contesto $\overset{def}{\Leftrightarrow} \exists G$ libera da contesto tale che *L*(*G*) = *L*.

- Il termine C.F. nasce dal fatto **che la sostituzione di un *NT* non è condizionata dal contesto** - ossia dai caratteri adiacenti - in cui compare.
- Un *NT* *A* in una forma di frase può sempre essere sostituito usando una produzione del tipo $A \to \beta$. La sostituzione è sempre valida.

Esempio

- Se si ha una grammatica C.F. che genera *L*, **non è detto che non esista** un'altra grammatica che generi lo stesso linguaggio.
  - Un linguaggio può essere generato da più grammatiche

- Esempio: linguaggio delle parentesi ben formate
- G_1

  S → ( )

  S → (S)

  S → SS

  G_2

  S → (A

  S → (S A

  S → SS

  A →)

Linguaggi liberi da contesto

- Un linguaggio può essere generato da più grammatiche **dunque**

- Se $L = L(G)$ e G non è C.F., non possiamo affermare che L non è C.F. perché **non possiamo escludere** che esista una grammatica C.F. G' per cui $L=L(G')$.

Esempi di linguaggi C.F.

- Il linguaggio delle parentesi ben formate
- Il linguaggio dei numeri interi relativi
- Il linguaggio $L = \{a^n b^n \mid n > 0\}$
- Il linguaggio delle stringhe con ugual numero di 0 e di 1.
- Il linguaggio $L = \{a^n b^{2n} \mid n > 0\}$

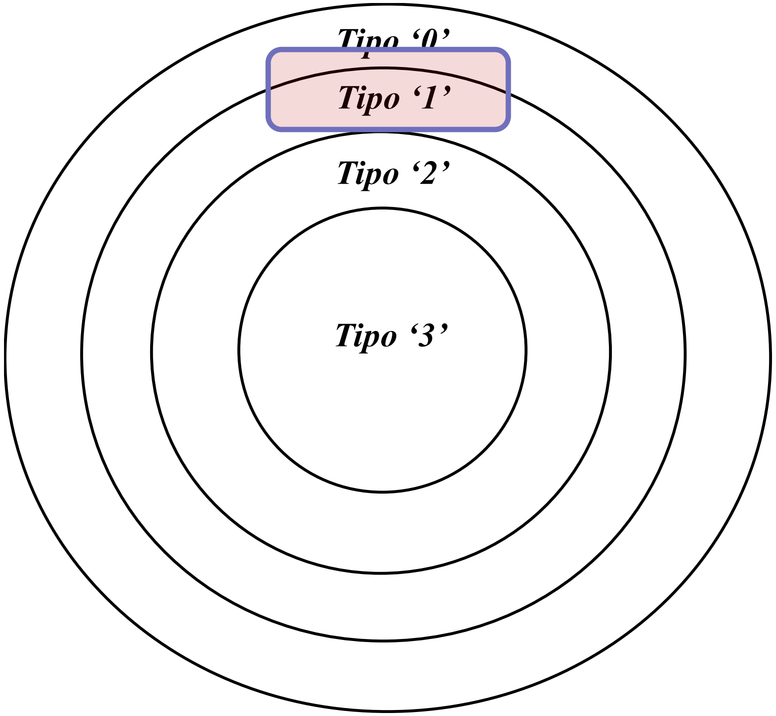
Esempi di linguaggi C.F.

- Il linguaggio delle parentesi ben formate
- Il linguaggio dei numeri interi relativi
- Il linguaggio $L = \{a^n b^n \mid n > 0\}$
- Il linguaggio delle stringhe con ugual numero di 0 e di 1.
- Il linguaggio $L = \{a^n b^{2n} \mid n > 0\}$

- **La maggior parte dei linguaggi di programmazione ricade nella classe dei linguaggi C.F.**
  - http://www.lysator.liu.se/c/ANSI-C-grammar-y.html
  - https://docs.python.org/3/reference/grammar.html

Classificazione delle grammatiche secondo Chomsky

- Definizione: Gerarchia di Chomsky (1956-1959)
  - La classe dei linguaggi di tipo 2 è inclusa nei c.d. linguaggi di tipo ‘1’ (linguaggi **dipendenti da contesto**)



Definizione di grammatica dipendente da contesto

- Una grammatica $G = (X, V, S, P)$ è *dipendente da contesto* (o *context-sensitive* - C.S.) se ogni produzione è in una delle seguenti forme:
  - (1) $yAz \to ywz$ con $A \in V,\ \ y, z \in (X \cup V)^*,\ \ w \in (X \cup V)^+$
    che si legge: "*A* può essere sostituita con *w* nel contesto *y-z* " (contesto sinistro *y* e contesto destro *z*).
  - (2) $S \to \lambda$ purché *S* non compaia nella parte destra di alcuna produzione.

Definizione di grammatica dipendente da contesto

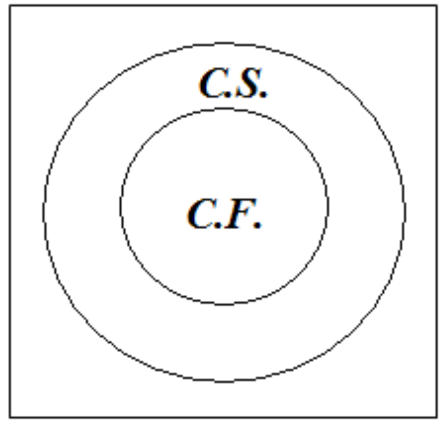
- Una grammatica $G = (X, V, S, P)$ è *dipendente da contesto* (o *context-sensitive* - C.S.) se ogni produzione è in una delle seguenti forme:
  - (1) $yAz \to ywz$ con $A \in V,\ \ y,z \in (X \cup V)^*,\ \ w \in (X \cup V)^+$
    che si legge: "*A* può essere sostituita con *w* nel contesto *y-z* " (contesto sinistro *y* e contesto destro *z*).
  - (2) $S \to \lambda$ purché *S* non compaia nella parte destra di alcuna produzione.

  - Esempi di produzioni dipendenti da contesto
    - abAa → abca
    - aAb → accb
    - aA -> aBc

Definizione di linguaggio dipendente da contesto

- Un linguaggio *L* è *dipendente da contesto* se può essere generato da una grammatica dipendente da contesto.

Relazione tra linguaggi C.F. e C.S.

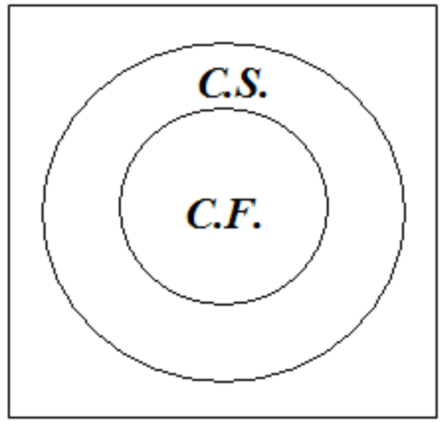


- Tale relazione sussiste perché le regole di produzione C.S. sono una generalizzazione di quelle C.F.
- Le produzioni C.F. sono un caso particolare delle produzioni di tipo (1) delle grammatiche C.S. **Verifichiamolo.**

$$v \rightarrow w \in P : v \in V$$

C.F.

Relazione tra linguaggi C.F. e C.S.



- Tale relazione sussiste perché le regole di produzione C.S. sono una generalizzazione di quelle C.F.
- Le produzioni C.F. sono un caso particolare delle produzioni di tipo (1) delle grammatiche C.S. **Verifichiamolo.**

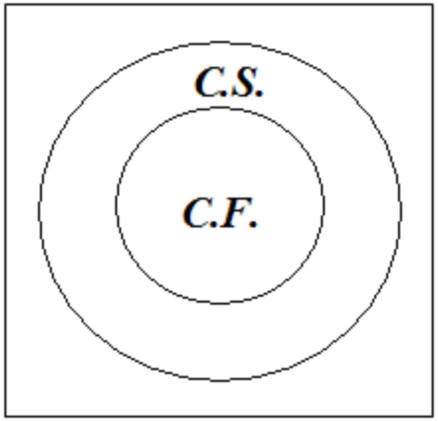
$$v \rightarrow w \in P : v \in V$$

C.F.

$$yAz \rightarrow ywz$$

C.S.

Relazione tra linguaggi C.F. e C.S.



- Tale relazione sussiste perché le regole di produzione C.S. sono una generalizzazione di quelle C.F.
- Le produzioni C.F. sono un caso particolare delle produzioni di tipo (1) delle grammatiche C.S., che si verifica quando:

  $y = z = \lambda$ contesto destro e sinistro equivalenti alla parola vuota (c'è una eccezione).

Eccezione

- Le produzioni C.F. sono un caso particolare delle produzioni di tipo (1) delle grammatiche C.S., che si verifica quando contesto destro e sinistro sono equivalenti alla parola vuota.
  - **Osserviamo però con attenzione la definizione di grammatica C.F.**

$$G \text{ è libera da contesto } \overset{def}{\Leftrightarrow} \forall\, v \to w \in P : v \in V$$
$$w \in (X \cup V)^*$$

Eccezione

- Le produzioni C.F. sono un caso particolare delle produzioni di tipo (1) delle grammatiche C.S., che si verifica quando contesto destro e sinistro sono equivalenti alla parola vuota.
  - **Osserviamo però con attenzione la definizione di grammatica C.F.**

$$G \text{ è libera da contesto } \overset{def}{\Leftrightarrow} \forall\, v \to w \in P : \begin{array}{l} v \in V \\ w \in (X \cup V)^* \end{array}$$

  - **Osserviamo ora la definizione di grammatica C.S. Cosa notiamo?**

$$yAz \to ywz \quad A \in V,\ \ y, z \in (X \cup V)^*,\ \ w \in (X \cup V)^+$$

Eccezione

- Le produzioni C.F. sono un caso particolare delle produzioni di tipo (1) delle grammatiche C.S., che si verifica quando contesto destro e sinistro sono equivalenti alla parola vuota.
  - **Osserviamo però con attenzione la definizione di grammatica C.F.**

$$G \text{ è libera da contesto } \overset{def}{\Leftrightarrow} \forall\, v \to w \in P : v \in V,\ w \in (X \cup V)^*$$

  - **Osserviamo ora la definizione di grammatica C.S. Cosa notiamo?**

$$yAz \to ywz \quad A \in V,\ \ y, z \in (X \cup V)^*, \boxed{w \in (X \cup V)^+}$$

Eccezione

- Le produzioni C.F. sono un caso particolare delle produzioni di tipo (1) delle grammatiche C.S., che si verifica quando contesto destro e sinistro sono equivalenti alla parola vuota.
  - Osservando con attenzione la definizione di grammatica C.F. si nota che, $w \in (X \cup V)^*$ mentre nella definizione di grammatica C.S. $w \in (X \cup V)^+$.
  - Dunque le grammatiche C.F. ammettono produzioni del tipo, $A \to \lambda$ con *A* che può anche non essere il simbolo iniziale, **mentre le grammatiche C.S. non ammettono tali produzioni.**
  - Chiameremo tutte le produzioni del tipo *λ-produzioni* o *λ-regole*.

Esempi

- Esempi di produzioni contestuali
  - $bC \rightarrow bc$
  - $baACbA \rightarrow baAabA$
- Esempio di grammatica contestuale
  - $S \rightarrow \lambda \mid bC$

    $bC \rightarrow bc$

$S \rightarrow \lambda$ è una produzione C.S. ed *S* non compare a destra di un'altra produzione.

- Di che produzione si tratta?
  - $CB \rightarrow BC$

Esempi

- Esempi di produzioni contestuali
  - $bC \to bc$
  - $baACbA \to baAabA$
- Esempio di grammatica contestuale
  - $S \to \lambda \mid bC$

    $bC \to bc$

$S \to \lambda$ è una produzione C.S. ed *S* non compare a destra di un'altra produzione.

- **Di che produzione si tratta?**
  - $CB \to BC$

non è né C.S. né C.F. È una produzione *monotona* perché del tipo $v \to w$ con $|v| \leq |w|$

Definizione di grammatica monotona

- Una grammatica $G = (X, V, S, P)$ è *monotona* se ogni sua produzione è monotona, cioè se

$$\forall v \to w \in P : |v| \le |w|$$

- Esempi

$$AB \to CDEF$$
$$CB \to BC$$

Definizione di linguaggio monotono

- Un linguaggio *L* è *monotono* se può essere generato da una grammatica monotona.

Esempio

- Produzioni monotone
  - $AB \rightarrow CDEF$
  - $CB \rightarrow BC$
- Una produzione monotona può essere **sostituita da una sequenza di produzioni contestuali** senza alterare il linguaggio generato.
  - $AB \rightarrow CDEF$ può essere sostituita dalle seguenti produzioni contestuali:
    - $AB \rightarrow AG$ B → G con A come **contesto sinistro**

Esempio

- Produzioni monotone
  - $AB \rightarrow CDEF$
  - $CB \rightarrow BC$
- Una produzione monotona può essere **sostituita da una sequenza di produzioni contestuali** senza alterare il linguaggio generato.
  - $AB \rightarrow CDEF$ può essere sostituita dalle seguenti produzioni contestuali:
    - $AB \rightarrow AG$ — B → G con A come **contesto sinistro**
    - $AG \rightarrow CG$ — A → C con G come **contesto destro**

Esempio

- Produzioni monotone
  - $AB \rightarrow CDEF$
  - $CB \rightarrow BC$
- Una produzione monotona può essere **sostituita da una sequenza di produzioni contestuali** senza alterare il linguaggio generato.
  - $AB \rightarrow CDEF$ può essere sostituita dalle seguenti produzioni contestuali:
    - $AB \rightarrow AG$ B → G con A come **contesto sinistro**
    - $AG \rightarrow CG$ A → C con G come **contesto destro**
    - $CG \rightarrow CDEF$ G → DEF con C come **contesto sinistro**

Esempio

- Produzioni monotone
  - $CB \rightarrow BC$ può essere sostituita dalle seguenti produzioni contestuali:
    - $CB \rightarrow XB$
    - $XB \rightarrow XC$
    - $XC \rightarrow BC$

    oppure
    - $CB \rightarrow X_1B$
    - $X_1B \rightarrow X_1X_2$
    - $X_1X_2 \rightarrow X_1C$
    - $X_1C \rightarrow BC$

Esempio

- Produzioni monotone
  - $CB \rightarrow BC$ può essere sostituita dalle seguenti produzioni contestuali:
    - $CB \rightarrow XB$
    - $XB \rightarrow XC$
    - $XC \rightarrow BC$

    oppure
    - $CB \rightarrow X_1B$
    - $X_1B \rightarrow X_1X_2$
    - $X_1X_2 \rightarrow X_1C$
    - $X_1C \rightarrow BC$

Ad ogni passaggio tengo «bloccato» un contesto e applico la produzione sulla parte restante della stringa.

I passaggi richiedono l'introduzione di nuovi non terminali.

Proposizione

- Cosa possiamo intuire da questa relazione?
  - La classe dei linguaggi contestuali coincide con la classe dei linguaggi monotoni.
- Tale proposizione deriva immediatamente dal teorema che segue
  - *Dimostrabile con la 'doppia inclusione'*

Teorema

- **Sia *G* una grammatica monotona**, cioè tale che ogni produzione di *G* è della forma $v \to w$ con $|v| \le |w|$, eccetto che ci può essere un'unica λ-produzione $S \to \lambda$ se *S* non appare alla destra di una produzione.
- Esiste allora una **grammatica C.S. *G'* equivalente a *G*, cioè tale che *L(G)=L(G')*.**
- Il teorema precedente può essere enunciato anche nella seguente forma:

Teorema (seconda formulazione)

- **Un linguaggio *L* è dipendente da contesto se e solo se** esiste una grammatica *G* tale che *L* = *L*(*G*) ed ogni produzione di *G* nella forma $u \to v$ ha la proprietà che: $0 < |u| \leq |v|$, con una sola eccezione: se $\lambda \in L(G)$ allora $S \to \lambda$ è una produzione di *G* ed in tal caso *S* non può comparire nella parte destra di altre produzioni.

  Dimostrazione (doppia inclusione)

  $\Rightarrow$ Ogni linguaggio contestuale è monotono

  $\Leftarrow$ Ogni linguaggio monotono è contestuale

Dimostrazione

- $\Rightarrow$) Ogni linguaggio contestuale è monotono. Banale.

  Se *L* è dipendente da contesto allora, per definizione, esiste *G* dipendente da contesto tale che *L* = *L*(*G*).

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

Allora ogni produzione di G è in una delle due forme:

- (1) $yAz \to ywz$ con $A \in V,\ y, z \in (X \cup V)^*,\ w \in (X \cup V)^+$
- (2) $S \to \lambda$ con *S* che non compare nella parte destra di alcuna produzione.

Dimostrazione

- $\Rightarrow$) Ogni linguaggio contestuale è monotono. Banale.

  Se L è dipendente da contesto allora, per definizione, esiste G dipendente da contesto tale che $L = L(G)$.

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

Allora ogni produzione di G è in una delle due forme:

  - (1) $yAz \to ywz$ con $A \in V,\ y,z \in (X \cup V)^*,\ w \in (X \cup V)^+$
  - (2) $S \to \lambda$ con S che non compare nella parte destra di alcuna produzione.

  Dunque, ogni produzione di G verifica la condizione $u \to v$, con $0 < |u| \le |v|$, se è del tipo (1), mentre se è del tipo (2) con S che non compare a destra di alcuna produzione, ricade nell'eccezione. Pertanto G è la grammatica cercata.

Dimostrazione

- $\Leftarrow$) Ogni linguaggio monotono è contestuale
  Sia *G* una grammatica in cui ogni produzione è nella forma $u \rightarrow v$, con $0 < |u| \leq |v|$. Senza ledere la generalità della dimostrazione, possiamo supporre che una generica produzione di *G* abbia il formato:
  $A_1 A_2 ... A_m \rightarrow B_1 B_2 ... B_n \quad m \leq n \qquad A_i \in V, \; i = 1, 2, ..., m$

  Denotiamo con C_1, C_2, …, C_m *m* simboli nonterminali non presenti in *G*.

Dimostrazione

- Utilizziamo le C_k, k = 1, 2, …, m per costruire nuove regole contestuali che riscrivono la stringa $A_1A_2...A_m$ con $B_1B_2...B_n$.

$$\left.\begin{array}{l}
A_1A_2...A_m \rightarrow C_1A_2...A_m \\
C_1A_2...A_m \rightarrow C_1C_2A_3...A_m \\
... \\
C_1C_2...C_{m-1}A_m \rightarrow C_1C_2...C_{m-1}C_mB_{m+1}...B_n \\
C_1C_2...C_{m-1}C_mB_{m+1}...B_n \rightarrow C_1...C_{m-1}B_mB_{m+1}...B_n \\
... \\
C_1B_2...B_n \rightarrow B_1B_2...B_n
\end{array}\right\} \begin{array}{c} 2m \\ \text{produzioni} \end{array}$$

La nuova grammatica che incorpora queste produzioni è contestuale e si può dimostrare che *L(G)=L(G')*.
Lasciamo per esercizio tale dimostrazione. *c.v.d.*

Esempio

$$\underbrace{ABC}_{m=3} \rightarrow \underbrace{DEFGH}_{n=5}$$

6 produzioni contestuali

$$ABC \rightarrow C_1BC$$

$$C_1BC \rightarrow C_1C_2C$$

$$C_1C_2C \rightarrow C_1C_2C_3GH$$

$$C_1C_2C_3GH \rightarrow C_1C_2FGH$$

$$C_1C_2FGH \rightarrow C_1EFGH$$

$$C_1EFGH \rightarrow DEFGH$$

Esercizio

- Considieriamo il linguaggio:

$$L = \{a^n b^n c^n \mid n > 0\}$$

Determiniamo una grammatica che genera tale linguaggio.

Esercizio

Esempio 3.3

Consideriamo il linguaggio:

$$L = \{a^n b^n c^n \mid n > 0\}$$

Determiniamo una grammatica che genera tale linguaggio.

Il linguaggio somiglia ad un linguaggio già visto, $\{a^n b^n \mid n > 0\}$, generato dalla grammatica $S \to aSb \mid ab$.

Dunque le produzioni saranno del tipo:

$$S \to a\overset{(1)}{S}BC \mid a\overset{(2)}{B}C$$

il NT B per generare le b

il NT C per generare le c

Esercizio

Se applichiamo una volta la (1) e poi la (2), abbiamo però:

$$S \underset{(1)}{\Rightarrow} aSBC \underset{(2)}{\Rightarrow} aaBCBC$$

che non è nella forma desiderata in quanto le b e le c risulterebbero alternate.

Generalizzando, se applichiamo n-1 volte la (1) e poi la (2), si ha:

$$S \overset{n-1}{\underset{(1)}{\Rightarrow}} a^{n-1} S \underbrace{BCBC...BC}_{n-1} \underset{(2)}{\Rightarrow} a^n \underbrace{BCBC...BC}_{n} = a^n (BC)^n$$

Esercizio

Abbiamo dunque bisogno di una produzione che riporti le B e le C in posizione corretta:

$$CB \overset{(3)}{\rightarrow} BC$$

con cui:

$$S \underset{(1)}{\Rightarrow} aSBC \underset{(2)}{\Rightarrow} aaBCBC \underset{(3)}{\Rightarrow} aaBBCC = a^2B^2C^2$$

e

$$S \overset{n-1}{\underset{(1)}{\Rightarrow}} a^{n-1}S(BC)^{n-1} \underset{(2)}{\Rightarrow} a^n \underbrace{BCBC...BC}_{n} \underset{(3)}{\Rightarrow} a^n BBCC \underbrace{BC...BC}_{n-2} \underset{(3)}{\Rightarrow}$$

$$\underset{(3)}{\Rightarrow} a^n BBCBCC...BC \underset{(3)}{\Rightarrow} a^n BBBCCC \underbrace{BC...BC}_{n-3} \overset{*}{\underset{(3)}{\Rightarrow}} a^n B^n C^n$$

.

Esercizio

Ora abbiamo bisogno delle produzioni che generano i terminali b e c.

Consideriamo la produzione $B \rightarrow b$. Non va bene. Perché? Perché altrimenti potremmo applicarla all'inizio di una derivazione, ottenendo stringhe del tipo:

$$a^n bCbC...bC$$

che impediscono di applicare la (3) (ed hanno bisogno di ulteriori produzioni per scambiare le b con le C) e quindi di trasformare le B in b solo dopo che le B hanno raggiunto la posizione corretta.

Esercizio

Ora abbiamo bisogno delle produzioni che generano i terminali b e c.

Consideriamo la produzione $B \to b$. Non va bene. Perché? Perché altrimenti potremmo applicarla all'inizio di una derivazione, ottenendo stringhe del tipo:

$$a^n bCbC...bC$$

che impediscono di applicare la (3) (ed hanno bisogno di ulteriori produzioni per scambiare le b con le C) e quindi di trasformare le B in b solo dopo che le B hanno raggiunto la posizione corretta.

Dunque dobbiamo considerare produzioni contestuali che trasformino le B in b solo dopo che hanno raggiunto la posizione corretta:

$aB \overset{(4)}{\to} ab$ per la prima occorrenza delle B

$bB \overset{(5)}{\to} bb$ per le restanti occorrenze delle B

Analogamente per le C:

$bC \overset{(6)}{\to} bc$ per la prima occorrenza delle C

$cC \overset{(7)}{\to} cc$ per le restanti occorrenze delle C

Esercizio

Quindi, la grammatica G che genera $L = \{a^n b^n c^n \mid n > 0\}$ è:

$$S \overset{(1)}{\rightarrow} aSBC \mid \overset{(2)}{aBC}$$

$$CB \overset{(3)}{\rightarrow} BC$$

$$aB \overset{(4)}{\rightarrow} ab$$

$$bB \overset{(5)}{\rightarrow} bb$$

$$bC \overset{(6)}{\rightarrow} bc$$

$$cC \overset{(7)}{\rightarrow} cc$$

La grammatica è monotona, ma è facilmente determinabile una grammatica C.S. equivalente.

$$CB \overset{(3)}{\rightarrow} BC \qquad \begin{aligned} CB &\overset{(3.a)}{\rightarrow} XB \\ XB &\overset{(3.b)}{\rightarrow} XC \\ XC &\overset{(3.c)}{\rightarrow} BC \end{aligned}$$

Domande?


-It gets easier.
-Huh?
Every day, it gets a little easier.
-Yeah?
-But you gotta do it every day.
That's the hard part.
But it does get easier.