# Linguaggi di Programmazione
Docente: Cataldo Musto

Capitolo 5 – Grammatiche e macchine

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

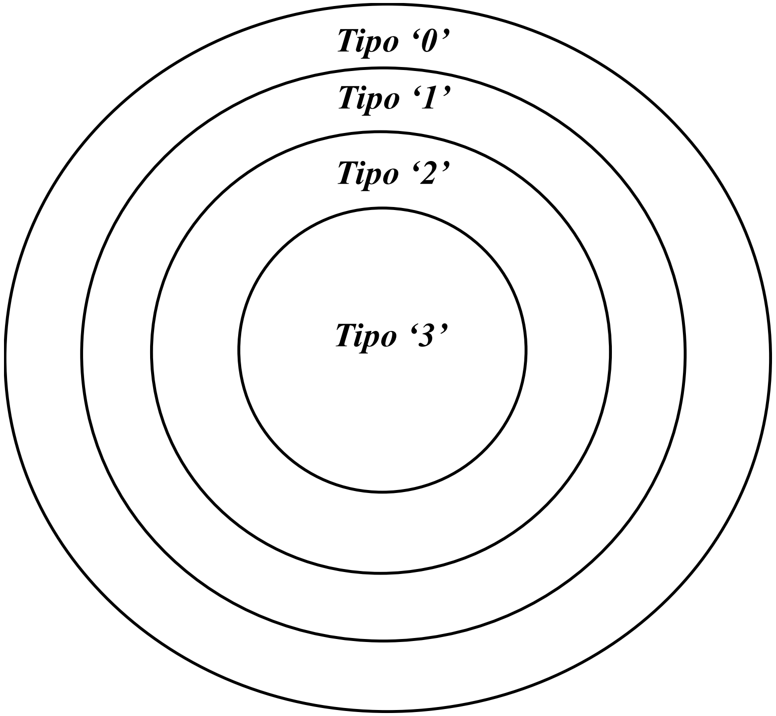
- Definizione: Gerarchia di Chomsky (1956-1959)
  - Sia $G = (X, V, S, P)$ una grammatica. Dalla definizione di grammatica, si ha:

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

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

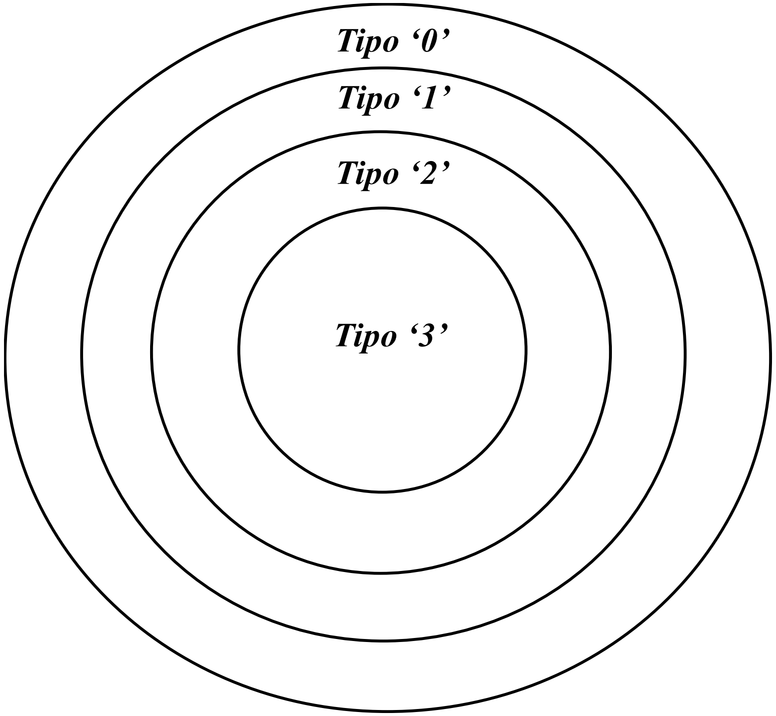
# Classificazione delle grammatiche secondo Chomsky

- Chomsky definisce quattro classi di linguaggi



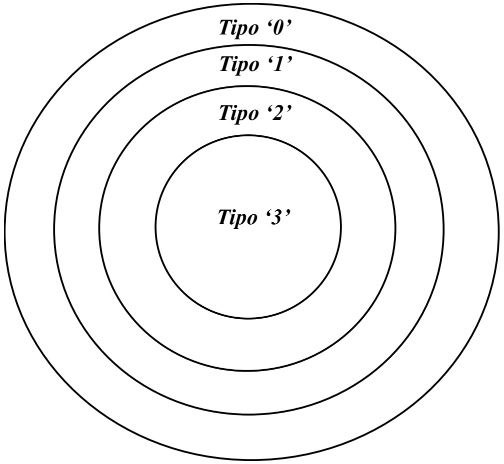
# Classificazione delle grammatiche secondo Chomsky

- Definizione: Gerarchia di Chomsky (1956-1959)
  - Tipo ‘0’ - Quando le stringhe che appaiono nella produzione $v \rightarrow w$ non sono soggette ad alcuna limitazione.



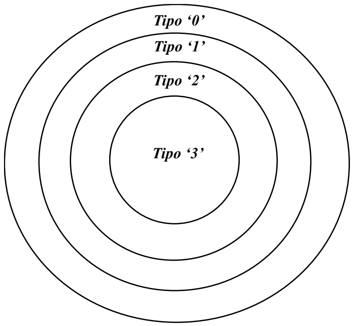
# Classificazione delle grammatiche secondo Chomsky

- Definizione: Gerarchia di Chomsky (1956-1959)
  - Tipo '0' - Quando le stringhe che appaiono nella produzione $v \rightarrow w$ non sono soggette ad alcuna limitazione.
  - Tipo '1' - Dipendente da contesto - quando le produzioni sono limitate alla forma:
    (1) $yAz \rightarrow ywz$ , con $A \in V,\ y, z \in (X \cup V)^*,\ w \in (X \cup V)^+$
    (2) $S \rightarrow \lambda$ , purché *S* non compaia nella parte destra di alcuna produzione.



# Classificazione delle grammatiche secondo Chomsky

- Definizione: Gerarchia di Chomsky (1956-1959)
  - Tipo ‘0’ - Quando le stringhe che appaiono nella produzione $v \rightarrow w$ non sono soggette ad alcuna limitazione.
  - Tipo ‘1’ - Dipendente da contesto - quando le produzioni sono limitate alla forma:
    (1) $yAz \rightarrow ywz$ , con $A \in V,\ y,z \in (X \cup V)^*,\ w \in (X \cup V)^+$
    (2) $S \rightarrow \lambda$ , purché *S* non compaia nella parte destra di alcuna produzione.
  - Tipo ‘2’ - Libera da contesto - quando le produzioni sono limitate alla forma: $v \rightarrow w$ con $v \in V$



# Classificazione delle grammatiche secondo Chomsky

- Definizione: Gerarchia di Chomsky (1956-1959)
  - Tipo '0' - Quando le stringhe che appaiono nella produzione $v \to w$ non sono soggette ad alcuna limitazione.
  - Tipo '1' - Dipendente da contesto - quando le produzioni sono limitate alla forma:
    (1) $yAz \to ywz$ , con $A \in V,\ y,z \in (X \cup V)^*,\ w \in (X \cup V)^+$
    (2) $S \to \lambda$ , purché *S* non compaia nella parte destra di alcuna produzione.
  - Tipo '2' - Libera da contesto - quando le produzioni sono limitate alla forma: $v \to w$ con $v \in V$
  - Tipo '3' - Lineare destra - quando le produzioni sono limitate alla forma:
    (1) $A \to bC$ con $A, C \in V$ e $b \in X$;
    (2) $A \to b$ con $A \in V$ e $b \in X \cup \{\lambda\}$.

# Classificazione delle grammatiche secondo Chomsky

- Una grammatica di tipo ‘3’ è detta lineare destra perché il *NT*, se c’è, compare a destra (nella parte destra della produzione).

- Un linguaggio generato da una tale grammatica è detto *di tipo ‘3’* o *lineare a destro*.

# Esempi di linguaggi di tipo ‘3’

- Consideriamo la grammatica $G_1$:

$$S \to \lambda \mid 0 \mid 0S \mid 1 \mid 1S$$

$G_1$ è lineare destra.

**Quale è il linguaggio generato da questa grammatica?**

# Esempi di linguaggi di tipo '3'

- Consideriamo la grammatica $G_1$:

$$S \to \lambda \mid 0 \mid 0S \mid 1 \mid 1S$$

$G_1$ è lineare destra.

$L(G_1) = \{0, 1\}^*$ (l'insieme di tutte le stringhe binarie).

# Esempi di linguaggi di tipo ‘3’

- Consideriamo la grammatica $G_1$:

$$S \rightarrow \lambda \mid 0 \mid 0S \mid 1 \mid 1S$$

$G_1$ è lineare destra. $L(G_1) = \{0,1\}^*$ (l’insieme di tutte le stringhe binarie).

- Consideriamo la grammatica $G_2$:

$$S \rightarrow \lambda \mid 0S \mid 1T$$

$$T \rightarrow 0T \mid 1S$$

$G_2$ è lineare destra.

# Esempi di linguaggi di tipo ‘3’

- Consideriamo la grammatica $G_1$:

$$S \rightarrow \lambda \mid 0 \mid 0S \mid 1 \mid 1S$$

  $G_1$ è lineare destra. $L(G_1) = \{0,1\}^*$(l’insieme di tutte le stringhe binarie).

- Consideriamo la grammatica $G_2$:

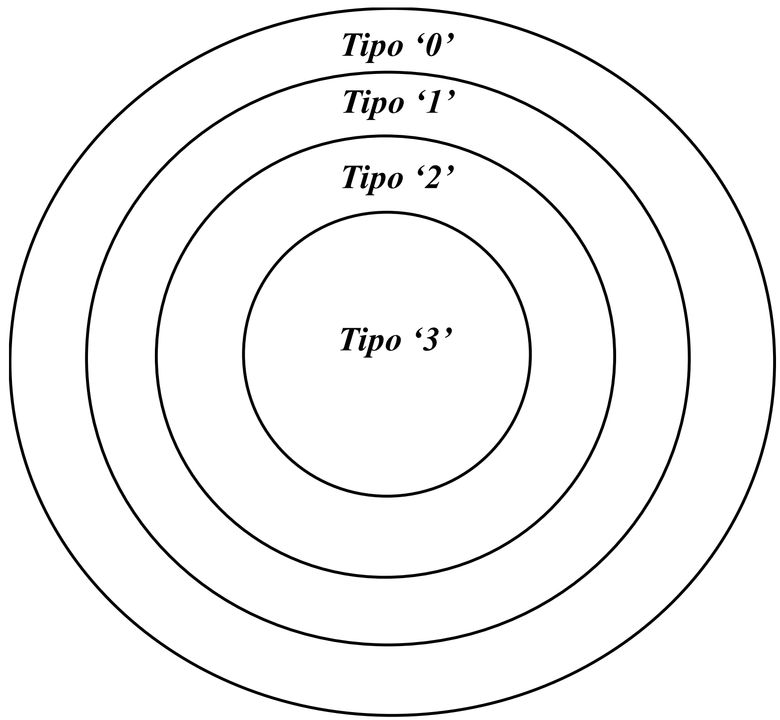
$$S \rightarrow \lambda \mid 0S \mid 1T$$
$$T \rightarrow 0T \mid 1S$$

  $G_2$ è lineare destra.

$$L(G_2) = \left\{ w \in \{0,1\}^* \mid w \text{ ha un numero pari di } 1 \right\}$$

# Teorema della gerarchia

- Dimostriamo formalmente che le quattro classi di linguaggi viste costituiscono una gerarchia



# Teorema della gerarchia

- Denotiamo con $\mathcal{L}_i$ il seguente insieme:

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

(classe dei linguaggi di tipo *i*).

La gerarchia di Chomsky è una gerarchia in senso stretto di classi di linguaggi:

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

Dimostrazione

# Teorema della gerarchia

- $\mathcal{L}_3 \underset{\neq}{\subset} \mathcal{L}_2$

  Per dimostrare che la classe di linguaggi $\mathcal{L}_3$ è *inclusa propriamente* nella classe di linguaggi $\mathcal{L}_2$ si deve dimostrare che ogni linguaggio di tipo ‘3’ può essere generato da una grammatica di tipo ‘2’ e che esiste almeno un linguaggio C.F. (di tipo ‘2’) che non può essere generato da una grammatica di tipo ‘3’.

# Teorema della gerarchia

- $\mathcal{L}_3 \underset{\neq}{\subset} \mathcal{L}_2$

  Per dimostrare che la classe di linguaggi $\mathcal{L}_3$ è *inclusa propriamente* nella classe di linguaggi $\mathcal{L}_2$ si deve dimostrare che ogni linguaggio di tipo ‘3’ può essere generato da una grammatica di tipo ‘2’ e che esiste almeno un linguaggio C.F. (di tipo ‘2’) che non può essere generato da una grammatica di tipo ‘3’.

  $\mathcal{L}_3 \subseteq \mathcal{L}_2$ discende dalle definizioni di linguaggio di tipo ‘3’ e di grammatica di tipo ‘2’. Infatti, si osserva facilmente che ogni grammatica di tipo ‘3’ è anche una grammatica di tipo ‘2’.

$$
\begin{array}{llll}
(1) & A \to bC & \text{con} & A, C \in V \ \text{ e } \ b \in X; \\
(2) & A \to b & \text{con} & A \in V \ \text{ e } \ b \in X \cup \{\lambda\}.
\end{array}
$$

# Teorema della gerarchia

- $\mathcal{L}_3 \underset{\neq}{\subset} \mathcal{L}_2$

  Per dimostrare che la classe di linguaggi $\mathcal{L}_3$ è *inclusa propriamente* nella classe di linguaggi $\mathcal{L}_2$ si deve dimostrare che ogni linguaggio di tipo '3' può essere generato da una grammatica di tipo '2' e che esiste almeno un linguaggio C.F. (di tipo '2') che non può essere generato da una grammatica di tipo '3'.

  $\mathcal{L}_3 \subseteq \mathcal{L}_2$ discende dalle definizioni di linguaggio di tipo '3' e di grammatica di tipo '2'. Infatti, si osserva facilmente che ogni grammatica di tipo '3' è anche una grammatica di tipo '2'.

  $\mathcal{L}_3 \neq \mathcal{L}_2$: posponiamo questa dimostrazione.
  Mostreremo che $L = \{a^k b^k \mid k > 0\}$ non è di tipo '3' (abbiamo già determinato una grammatica di tipo '2' che genera *L*).

# Teorema della gerarchia

- $\mathcal{L}_2 \underset{\neq}{\subset} \mathcal{L}_1$
  Abbiamo già osservato che le produzioni C.F. sono un caso particolare delle produzioni C.S.

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

- **Quando?**

# Teorema della gerarchia

- $\mathcal{L}_2 \underset{\neq}{\subset} \mathcal{L}_1$

  Abbiamo già osservato che le produzioni C.F. sono un caso particolare delle produzioni C.S.

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

  (2) $S \rightarrow \lambda$ , purché $S$ non compaia nella parte destra di alcuna produzione.

- **Quando?**
  - Quando il contesto sinistro e destro sono uguali alla parola vuota.
  - In più abbiamo un'unica eccezione. **Quale?**

# Teorema della gerarchia

- $\mathcal{L}_2 \underset{\neq}{\subset} \mathcal{L}_1$

  Abbiamo già osservato che le produzioni C.F. sono un caso particolare delle produzioni C.S.
- Con l'unica eccezione rappresentata dalle produzioni:

$$A \to \lambda, \qquad A \neq S$$

che sono C.F. ma ***non*** C.S. Dunque:

$$\forall L : L \in \mathcal{L}_2 \overset{def}{\Leftrightarrow} \exists G,\ G \text{ è C.F.} : L = L(G).$$

Se $A \to \lambda,\ A \in V \setminus \{S\}$ non è una produzione di G, allora G è anche C.S. (di tipo '1') e l'asserto è dimostrato.

- Il problema sorge se G ha almeno una $\lambda$-produzione. In tal caso, ci avvaliamo del seguente risultato:

# Lemma della stringa vuota

- Sia $G = (X, V, S, P)$ una grammatica C.F. con almeno una $\lambda$-produzione. Allora esiste una grammatica C.F. *G'* tale che:
  - i) *L(G)=L(G')* (*G'* è equivalente a *G*);
  - ii) se $\lambda \notin L(G)$ allora in *G'* non esistono produzioni del tipo $A \rightarrow \lambda$;
  - iii) se $\lambda \in L(G)$ allora in *G'* esiste un'unica produzione $S' \rightarrow \lambda$ , ove *S'* è il simbolo iniziale di *G'* ed *S'* non compare nella parte destra di alcuna produzione di *G'*.

  - Sinteticamente, se $\lambda \in L(G)$ lo facciamo produrre direttamente da S, altrimenti eliminiamo direttamente la produzione.

# Teorema della gerarchia

- Riprendiamo la dimostrazione di $\mathcal{L}_2 \underset{\neq}{\subset} \mathcal{L}_1$

$$\forall L : L \in \mathcal{L}_2 \overset{def}{\Leftrightarrow} \exists G,\ G \text{ è C.F.} : L = L(G).$$

Se G ha almeno una $\lambda$-produzione, utilizziamo il Lemma della stringa vuota per determinare una grammatica C.F. *G'* equivalente a G, ma priva di $\lambda$-produzioni (al più, in *G'* compare la produzione $S' \rightarrow \lambda$, ed *S'* non compare nella parte destra di alcuna produzione di *G'*). *G'* è di tipo '1'.

Questo dimostra che $\mathcal{L}_2 \subseteq \mathcal{L}_1$.

# Teorema della gerarchia

- $\mathcal{L}_2 \neq \mathcal{L}_1$

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

è di tipo ‘1’ ma non di tipo ‘2’. Si osservi che, per asserire che $L$ è di tipo ‘1’, ci siamo avvalsi del teorema che stabilisce l’equivalenza delle classi di linguaggi contestuali e monotoni.

# Teorema della gerarchia

- $\mathcal{L}_2 \neq \mathcal{L}_1$

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

è di tipo ‘1’ ma non di tipo ‘2’. Si osservi che, per asserire che *L* è di tipo ‘1’, ci siamo avvalsi del teorema che stabilisce l’equivalenza delle classi di linguaggi contestuali e monotoni.

- $\mathcal{L}_1 \underset{\neq}{\subset} \mathcal{L}_0$

Non lo dimostriamo formalmente. La dimostrazione comporta la conoscenza degli automi limitati lineari e delle macchine di Turing (che riconoscono linguaggi di tipo ‘0’ o ricorsivamente enumerabili).

- Ci limitiamo ad osservare che $\mathcal{L}_1 \subseteq \mathcal{L}_0$ discende direttamente dalle definizioni di linguaggio di tipo ‘1’ e di grammatica di tipo ‘0’. ***c.v.d.***

# Operazioni sui linguaggi

# Operazioni sui linguaggi

- Siano $L_1$ ed $L_2$ due linguaggi definiti su uno stesso alfabeto $X$ ($L_1, L_2 \subseteq X^*$).
  - *L'unione di $L_1$ ed $L_2$ è:*

$$L_1 \cup L_2 = \{w \mid w \in L_1 \vee w \in L_2\}$$

# Operazioni sui linguaggi

- Siano $L_1$ ed $L_2$ due linguaggi definiti su uno stesso alfabeto $X$ ($L_1, L_2 \subseteq X^*$).
  - *L'unione di $L_1$ ed $L_2$ è:*

$$L_1 \cup L_2 = \{w \mid w \in L_1 \vee w \in L_2\}$$

  - *La concatenazione di $L_1$ ed $L_2$ (anche detta il prodotto di $L_1$ ed $L_2$ o "$L_1$ punto $L_2$") è:*

$$L_1 \cdot L_2 = \{w \mid w = w_1 w_2,\ \ w_1 \in L_1,\ \ w_2 \in L_2\}$$

# Operazioni sui linguaggi

- Siano $L_1$ ed $L_2$ due linguaggi definiti su uno stesso alfabeto $X$ ($L_1, L_2 \subseteq X^*$).
  - *L'unione di $L_1$ ed $L_2$ è:*

$$L_1 \cup L_2 = \{w \mid w \in L_1 \vee w \in L_2\}$$

  - *La concatenazione di $L_1$ ed $L_2$ (anche detta il prodotto di $L_1$ ed $L_2$ o "$L_1$ punto $L_2$") è:*

$$L_1 \cdot L_2 = \{w \mid w = w_1 w_2,\ \ w_1 \in L_1,\ \ w_2 \in L_2\}$$

  - L'*iterazione* di $L_1$ (o *chiusura riflessiva e transitiva* di $L_1$ rispetto all'operazione di concatenazione, anche detta *stellatura* di $L_1$ o "*$L_1$ star*" o *chiusura di Kleene*) è:

$$L_1^* = \{w \mid w = w_1 w_2 \ldots w_n,\ \ n \geq 0 \text{ e } \forall i : w_i \in L_1\}$$

# Operazioni sui linguaggi

- Siano $L_1$ ed $L_2$ due linguaggi definiti su uno stesso alfabeto $X$ ( $L_1, L_2 \subseteq X^*$).
  - Il *complemento* di $L_1$ è:

$$\overline{L_1} = X^* - L_1$$

  - L'*intersezione* di $L_1$ ed $L_2$ è:

$$L_1 \cap L_2 = \{w \mid w \in L_1 \wedge w \in L_2\}$$

# Proprietà

- Dati $L_1, L_2, L_3 \subseteq X^*$ $(\equiv L_1, L_2, L_3 \in 2^{X^*})$, risulta:
  - $(L_1 \cdot L_2) \cdot L_3 = L_1 \cdot (L_2 \cdot L_3)$ (proprietà associativa)
  - $L_1 \cdot L_2 \neq L_2 \cdot L_1$
  - $L_1 \cdot \{\lambda\} = \{\lambda\} \cdot L_1 = L_1$ ($\{\lambda\}$ è l'elemento neutro rispetto all'operazione di concatenazione di linguaggi)

# Definizioni

- Fissato $\mathbf{X^*}$
  - 
- Possiamo definire $\mathbf{2^{X^*}}$
  - 

# Definizioni

- Fissato $\mathbf{X^*}$
  - Insieme di tutte le possibili stringhe che si possono costruire a partire da X
- Possiamo definire $\mathbf{2^{X^*}}$
  - Insieme di tutti i possibili sottoinsiemi di stringhe (insieme di tutti i possibili linguaggi!) che si possono costruire a partire da X

# Proprietà

- Dati $L_1, L_2, L_3 \subseteq X^*$ $(\equiv L_1, L_2, L_3 \in 2^{X^*}$ ), risulta:
  - $(L_1 \cdot L_2) \cdot L_3 = L_1 \cdot (L_2 \cdot L_3)$ (proprietà associativa)
  - $L_1 \cdot L_2 \neq L_2 \cdot L_1$
  - $L_1 \cdot \{\lambda\} = \{\lambda\} \cdot L_1 = L_1$ ($\{\lambda\}$ è l'elemento neutro rispetto all'operazione di concatenazione di linguaggi)

  **Dunque anche $\left(2^{X^*}, \cdot\right)$ è un monoide;**

  - $L_1 \cdot \varnothing = \varnothing \cdot L_1 = \varnothing$ ($\varnothing$ è l'elemento assorbente);
  - Se $\lambda \in L_1$ $L_2 \subseteq L_1 \cdot L_2$
    $L_2 \subseteq L_2 \cdot L_1$
  - Se $\lambda \in L_2$ $L_1 \subseteq L_1 \cdot L_2$
    $L_1 \subseteq L_2 \cdot L_1$

**Esempio 5.1**

Si considerino i linguaggi:

$$L_1 = \{a^{2n} \mid n \geq 0\} \qquad L_2 = \{b, cc\}$$

$$L_1 \cdot L_2 = \{b, cc, a^2b, a^2cc, a^4b, a^4cc, ...\}$$

$$L_2 \cdot L_1 = \{b, cc, ba^2, cca^2, ba^4, cca^4, ...\}$$

Dunque:

$$L_2 \subset L_1 \cdot L_2 \quad \text{e} \quad L_2 \subset L_2 \cdot L_1$$

mentre:

$$L_1 \not\subseteq L_1 \cdot L_2 \quad \text{e} \quad L_1 \not\subseteq L_2 \cdot L_1$$

## Esempio 5.1

Si considerino i linguaggi:

$$L_1 = \{a^{2n} \mid n \geq 0\} \qquad L_2 = \{b, cc\}$$

$$L_1 \cup L_2 = L_2 \cup L_1 = \{\lambda, b, cc, a^2, a^4, a^6, ...\}$$

$$L_2^* = \{\lambda, b, cc, bb, bcc, ccb, bbb, cccc, ...\}$$

$$L_1^* = \{\lambda, a^2, a^4, a^6, ...\} = L_1$$

# Definizione di potenza di un linguaggio

- Sia $L$ un linguaggio definito su un alfabeto $X$. Dicesi *potenza n-esima* di $L$, e si denota con $L^n$, $n \geq 0$, il seguente linguaggio:

$$L^n = \begin{cases} \{\lambda\} & \text{se } \; n = 0 \\ L^{n-1} \cdot L & \text{altrimenti} \end{cases}$$

Posto:

$$L^+ = \bigcup_{i \geq 1} L^i$$

si ha:

$$L^* = \{\lambda\} \cup L^+ = L^0 \cup L^+ = \bigcup_{i \geq 0} L^i$$

# Definizione di potenza di un linguaggio

- $L^+$ è detta chiusura transitiva rispetto alla operazione di concatenazione.
  Dunque si ha:

$$L^0 = \{\lambda\}$$

$$L^1 = L^0 \cdot L = L$$

$$L^2 = L^1 \cdot L = (L^0 \cdot L) \cdot L$$

$$L^3 = L^2 \cdot L = (L^1 \cdot L) \cdot L = \left((L^0 \cdot L) \cdot L\right) \cdot L$$

# Proposizione

- Sia *L* un linguaggio definito su un alfabeto *X*. Si ha:

$$L^* = \bigcup_{i \geq 0} L^i = \{\lambda\} \cup L \cup L^2 \cup L^3 \cup \ldots$$

# Proposizione

- Sia *L* un linguaggio definito su un alfabeto *X*. Si ha:

$$L^* = \bigcup_{i \geq 0} L^i = \{\lambda\} \cup L \cup L^2 \cup L^3 \cup \ldots$$

**<u>Esempio 5.2</u>**

Si consideri nuovamente il linguaggio $L_2 = \{b, cc\}$. Si ha:

$$L_2^{\ 0} = \{\lambda\},$$

$$L_2^{\ 1} = L_2,$$

$$L_2^{\ 2} = \{bb, bcc, ccb, cccc\},$$

$$L_2^{\ *} = L_2^{\ 0} \cup L_2^{\ 1} \cup L_2^{\ 2} \cup \ldots = \{\lambda, b, cc, bb, bcc, ccb, cccc, bbb, \ldots\}.$$

# Proprietà di chiusura delle classi di linguaggi

- Definizioni
  - L *linguaggio* definito su $X$ $\overset{def}{\Leftrightarrow} L \subseteq X^* \Leftrightarrow L \in 2^{X^*}$
  - $\mathcal{L}$ *classe di linguaggi* su $X$ $\overset{def}{\Leftrightarrow} \mathcal{L} \subseteq 2^{X^*} \Leftrightarrow \mathcal{L} \in 2^{2^{X^*}}$
    - *Esempi di classe di linguaggi, sono ad esempio «tutti i linguaggi di tipo 2» oppure «tutti i linguaggi di tipo 1»*

# Proprietà di chiusura delle classi di linguaggi

- Definizione di chiusura
  - Sia $\mathcal{L}$ una classe di linguaggi su *X*.

    Sia $\alpha$ un'operazione binaria sui linguaggi di $\mathcal{L}$:

    $$\alpha : 2^{X^*} \times 2^{X^*} \to 2^{X^*}, \qquad (L_1, L_2) \mapsto \alpha(L_1, L_2)$$

    Sia $\beta$ un'operazione unaria sui linguaggi di $\mathcal{L}$:

    $$\beta : 2^{X^*} \to 2^{X^*}, \qquad L \mapsto \beta(L)$$

# Proprietà di chiusura delle classi di linguaggi

- Definizione di chiusura
  - Sia $\mathcal{L}$ una classe di linguaggi su *X*.

    Sia $\alpha$ un'operazione binaria sui linguaggi di $\mathcal{L}$:

    $$\alpha : 2^{X^*} \times 2^{X^*} \rightarrow 2^{X^*}, \qquad (L_1, L_2) \mapsto \alpha(L_1, L_2)$$

    Sia $\beta$ un'operazione unaria sui linguaggi di $\mathcal{L}$:

    $$\beta : 2^{X^*} \rightarrow 2^{X^*}, \qquad L \mapsto \beta(L)$$

Esempi di operazioni binarie sui linguaggi?

Esempi di operazioni unarie?

# Proprietà di chiusura delle classi di linguaggi

- Definizione di chiusura
  - Sia $\mathcal{L}$ una classe di linguaggi su *X*.

    Sia $\alpha$ un'operazione binaria sui linguaggi di $\mathcal{L}$:

    $$\alpha : 2^{X^*} \times 2^{X^*} \to 2^{X^*}, \qquad (L_1, L_2) \mapsto \alpha(L_1, L_2)$$

    Sia $\beta$ un'operazione unaria sui linguaggi di $\mathcal{L}$:

    $$\beta : 2^{X^*} \to 2^{X^*}, \qquad L \mapsto \beta(L)$$

Esempi di operazioni binarie sui linguaggi?

Concatenazione

Esempi di operazioni unarie?

Iterazione

# Proprietà di chiusura delle classi di linguaggi

- Definizione di chiusura
  - Sia $\mathcal{L}$ una classe di linguaggi su *X*.

    Sia $\alpha$ un'operazione binaria sui linguaggi di $\mathcal{L}$:

    $$\alpha : 2^{X^*} \times 2^{X^*} \to 2^{X^*}, \qquad (L_1, L_2) \mapsto \alpha(L_1, L_2)$$

    Sia $\beta$ un'operazione unaria sui linguaggi di $\mathcal{L}$:

    $$\beta : 2^{X^*} \to 2^{X^*}, \qquad L \mapsto \beta(L)$$

    $\mathcal{L}$ è *chiusa* rispetto ad $\alpha$ $\overset{def}{\Leftrightarrow}$ $\forall\, L_1, L_2 \in \mathcal{L} : \alpha(L_1, L_2) \in \mathcal{L}$

    $\mathcal{L}$ è *chiusa* rispetto a $\beta$ $\overset{def}{\Leftrightarrow}$ $\forall\, L_1 \in \mathcal{L} : \beta(L_1) \in \mathcal{L}$
- Esempio
  - $\mathcal{L}$ è chiusa rispetto all'iterazione se $\forall\, L_1 \in \mathrm{L} : L_1^* \in \mathrm{L}$

# Teorema di chiusura

- La classe dei linguaggi di tipo *i*, *i* = 0, 1, 2, 3, è chiusa rispetto alle operazioni di *unione, concatenazione* ed *iterazione*.

# Teorema di chiusura

- La classe dei linguaggi di tipo *i*, *i* = 0, 1, 2, 3, è chiusa rispetto alle operazioni di *unione, concatenazione* ed *iterazione*.

  Dimostrazione

  La dimostrazione di questo teorema è *costruttiva*.

  Siano $L_1$ ed $L_2$ due linguaggi:

  $L_1 = L(G_1)$ $\quad G_1 = (X, V_1, S_1, P_1)$

  $L_2 = L(G_2)$ $\quad G_2 = (X, V_2, S_2, P_2)$

  Assumiamo che: $V_1 \cap V_2 = \emptyset \qquad S \notin V_1 \cup V_2$

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

Nel caso in cui tale assunzione non sia vera, ridenominiamo i nonterminali in comune.

# Dimostrazione Teorema di chiusura

- Lo schema generale della dimostrazione è il seguente:
  - consideriamo un'operazione alla volta (denotata con $\alpha$);
    - Esempio: unione
  - date $G_1$ e $G_2$, costruiamo $G$;
  - si dimostra che, se $G_1$ e $G_2$ sono di tipo $i$, allora $G$ è di tipo $i$;
  - si dimostra che $L(G) = \alpha(L_1, L_2)$ e dunque la classe di linguaggi $\mathcal{L}_i$ è chiusa rispetto alla operazione $\alpha$.

# Dimostrazione Teorema di chiusura

- UNIONE

  Fissate $G_1$ e $G_2$

  Costruiamo la grammatica: $G_3 = (X, V, S, P_3)$ e fissiamo:

$$P_3 = \{S \to S_1,\ S \to S_2\} \cup P_1 \cup P_2$$

# Dimostrazione Teorema di chiusura

- UNIONE

  Costruiamo la grammatica: $G_3 = (X, V, S, P_3)$ e fissiamo:

$$P_3 = \{S \to S_1,\ S \to S_2\} \cup P_1 \cup P_2$$

Osserviamo che, se $G_1$ e $G_2$ sono entrambe di tipo $i$, $i = 0, 1, 2$, lo è anche $G_3$. In ciascuno di questi casi, si ha:

$$L(G_3) = L_1 \cup L_2$$

Infatti una derivazione da $S$ in $G_3$ deve necessariamente iniziare o con:

$S \Rightarrow S_1$ ed in tal caso può generare unicamente parole di $L(G_1)$ oppure con

$S \Rightarrow S_2$ ed in tal caso genera una parola di $L(G_2)$.

Dunque, risulta dimostrato che <u>$\mathcal{L}_0$, $\mathcal{L}_1$ e $\mathcal{L}_2$, sono chiuse rispetto all'unione</u>. Perché questo non vale per $\mathcal{L}_3$?

# Dimostrazione Teorema di chiusura

- UNIONE

  Però, se $G_1$ e $G_2$ sono di tipo ‘3’, $G_3$ non è lineare destra, perché le produzioni $S \rightarrow S_1$ ed $S \rightarrow S_2$ non sono ammesse.

# Dimostrazione Teorema di chiusura

- UNIONE

  Però, se $G_1$ e $G_2$ sono di tipo '3', $G_3$ non è lineare destra, perché le produzioni $S \rightarrow S_1$ ed $S \rightarrow S_2$ non sono ammesse.
  Per avere ancora produzioni lineari destre che simulino il passo iniziale di una derivazione in $G_1$ ed anche il passo iniziale di una derivazione in $G_2$, costruiamo pertanto la grammatica: $G_4 = (X, V, S, P_4)$ ove:

$$P_4 = \{S \rightarrow w \mid S_1 \rightarrow w \in P_1\} \cup \{S \rightarrow w \mid S_2 \rightarrow w \in P_2\} \cup P_1 \cup P_2$$

  $G_4$ è lineare destra se $G_1$ e $G_2$ lo sono e inoltre:

$$L(G_4) = L_1 \cup L_2$$

  ed <u>$\mathcal{L}_3$ è chiusa rispetto all'unione</u>.

# Dimostrazione Teorema di chiusura

**Esempio 5.4**

$$G_1 \text{ con } P_1 = \{S_1 \to aA,\ A \to b\}$$
$$G_2 \text{ con } P_2 = \{S_2 \to cC,\ C \to b\}$$
$$V_1 \cap V_2 = \varnothing \quad \text{metre} \quad X_1 = \{a, b\} \text{ ed } X_2 = \{b, c\}$$

Consideriamo

$$X = \{a, b, c\} = X_1 \cup X_2$$

# Dimostrazione Teorema di chiusura

**Esempio 5.4**

$$G_1 \text{ con } P_1 = \{S_1 \to aA,\ A \to b\}$$
$$G_2 \text{ con } P_2 = \{S_2 \to cC,\ C \to b\}$$
$$V_1 \cap V_2 = \varnothing \quad \text{metre} \quad X_1 = \{a,b\} \text{ ed } X_2 = \{b,c\}$$

Consideriamo

$$X = \{a,b,c\} = X_1 \cup X_2$$

e

$$G_4 \text{ con } P_4 = \{S \to aA,\ S \to cC,\ S_1 \to aA,\ A \to b,\ S_2 \to cC,\ C \to b\}$$
$$L(G_1) = \{ab\} \qquad L(G_2) = \{cb\} \qquad L(G_4) = \{ab, cb\}$$

dunque

$$L(G_4) = L(G_1) \cup L(G_2)$$

e

$$G_4 = \left(X = \{a,b,c\},\ V = V_1 \cup V_2 \cup \{S\} = \{S, S_1, S_2, A, C\},\ S,\ P_4\right).$$

# Dimostrazione Teorema di chiusura

**<u>Esempio 5.4</u>**

$$G_1 \text{ con } P_1 = \{S_1 \to aA,\ A \to b\}$$

$$G_2 \text{ con } P_2 = \{S_2 \to cC,\ C \to b\}$$

$$V_1 \cap V_2 = \varnothing \quad \text{metre} \quad X_1 = \{a,b\} \text{ ed } X_2 = \{b,c\}$$

Consideriamo

$$X = \{a,b,c\} = X_1 \cup X_2$$

e

$$G_4 \text{ con } P_4 = \boxed{\{S \to aA,\ S \to cC,}\ S_1 \to aA,\ A \to b,\ S_2 \to cC,\ C \to b\}$$

$$L(G_1) = \{ab\} \qquad L(G_2) = \{cb\} \qquad L(G_4) = \{ab, cb\}$$

dunque

$$L(G_4) = L(G_1) \cup L(G_2)$$

e

$$G_4 = \left(X = \{a,b,c\},\ V = V_1 \cup V_2 \cup \{S\} = \{S, S_1, S_2, A, C\},\ S,\ P_4\right).$$

# Dimostrazione Teorema di chiusura

Osserviamo che $S_1$, $S_2$ e le produzioni $S_1 \to aA$, $S_2 \to cC$ non sono di alcuna utilità, in quanto non possono essere utilizzate in alcuna derivazione da $S$ in $G_4$. $S_1$ ed $S_2$ sono detti *nonterminali inutili* (Definizione 8.12) perché non compariranno mai all'interno di una forma di frase generata da $S$ in $G_4$.

La grammatica $G_4$, per effetto della eliminazione dei nonterminali inutili, diventa:

$$G_4' = \left(X' = \{a,b,c\},\ V' = \{S, A, C\},\ S,\ P_4' = \{S \to aA \mid cC,\ A \to b,\ C \to b\}\right)$$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Costruiamo la grammatica: $G_5 = (X, V, S, P_5)$ ove:

  $$P_5 = \{S \to S_1 S_2\} \cup P_1 \cup P_2$$

  Osserviamo che, se $G_1$ e $G_2$ sono entrambe di tipo $i$, $i = 0, 1, 2$, lo è anche $G_5$.

Il caso più semplice è rappresentato dalle grammatiche di tipo 2.

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Costruiamo la grammatica: $G_5 = (X, V, S, P_5)$ ove:

  $$P_5 = \{S \rightarrow S_1 S_2\} \cup P_1 \cup P_2$$

  Osserviamo che, se $G_1$ e $G_2$ sono entrambe di tipo *i*, *i* = 0, 1, 2, lo è anche $G_5$.

  **Se $G_1$ e $G_2$ sono C.F. (tipo '2'), allora si ha:**

  $$L(G_5) = L_1 \cdot L_2$$

  in quanto ogni derivazione da *S* in $G_5$ ha la seguente "struttura":

  $$S \Rightarrow S_1 S_2 \overset{*}{\underset{G_1}{\Rightarrow}} w_1 S_2 \overset{*}{\underset{G_2}{\Rightarrow}} w_1 w_2$$

  ove, evidentemente, si ha:

  $$S_1 \overset{*}{\Rightarrow} w_1 \qquad\qquad S_2 \overset{*}{\Rightarrow} w_2$$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Ad esempio il linguaggio $L = \{a^n b^{n+m}\}$ con $n \geq 0$ e $m \geq 0$

  può essere visto come $L = \{a^n b^n b^m\}$ e dunque immaginato come concatenazione di $L1 = \{a^n b^n\}$ e $L2 = \{b^m\}$

Dunque $G1 = \{S_1 \rightarrow aS_1b\ ,\ S_1 \rightarrow ab\}$ e $G2 = \{S_2 \rightarrow bS_2,\ S_2 \rightarrow b\}$

Il linguaggio concatenazione diviene $S \rightarrow S_1S_2$

Da cui poi si innescano le derivazioni dei due sotto-linguaggi

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Ad esempio il linguaggio $L = \{a^n b^{n+m}\}$ con $n \geq 0$ e $m \geq 0$

  può essere visto come $L = \{a^n b^n b^m\}$ e dunque immaginato come concatenazione di $L1 = \{a^n b^n\}$ e $L2 = \{b^m\}$

Importante!

Questo ragionamento non può essere applicato a linguaggi come $L = \{a^n b^n a^n\}$, sebbene sia $a^n b^n$ c che $a^n$ possano derivare da grammatiche di tipo 2. **Perché?**

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Ad esempio il linguaggio $L = \{a^n b^{n+m}\}$ con $n \geq 0$ e $m \geq 0$ può essere visto come $L = \{a^n b^n b^m\}$ e dunque immaginato come concatenazione di $L1 = \{a^n b^n\}$ e $L2 = \{b^m\}$

Importante!

Questo ragionamento non può essere applicato a linguaggi come $L = \{a^n b^n a^n\}$, sebbene sia $a^n b^n$ c che $a^n$ possano derivare da grammatiche di tipo 2. **Perché?**

La divisione di un linguaggio in due sotto-linguaggi può avvenire SOLO se il numero delle lettere è indipendente nei due sotto-gruppi (es., N e M). Altrimenti, la presenza di un vincolo, rende impossibile la divisione.

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Il seguente controesempio mostra che la condizione che $G_1$ e $G_2$ siano di tipo ‘2’ è fondamentale per la validità di

$$L(G_5) = L_1 \cdot L_2$$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Il seguente controesempio mostra che la condizione che $G_1$ e $G_2$ siano di tipo '2' è fondamentale per la validità di

$$L(G_5) = L_1 \cdot L_2$$

  Controesempio: Consideriamo

$G_1$ con $P_1 = \{S_1 \to b\}$

$G_2$ con $P_2 = \{bS_2 \to bb\}$.

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Il seguente controesempio mostra che la condizione che $G_1$ e $G_2$ siano di tipo '2' è fondamentale per la validità di

$$L(G_5) = L_1 \cdot L_2$$

  Controesempio: Consideriamo

$G_1$ con $P_1 = \{S_1 \rightarrow b\}$

$G_2$ con $P_2 = \{bS_2 \rightarrow bb\}$.

Da cui $L_1 = L(G_1) = \{b\}$ ed $L_2 = L(G_2) = \varnothing$

  Dunque $L_1 \cdot L_2 = \varnothing$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE
  Se costruiamo $G_5$, si ha: $P_5 = \{S \rightarrow S_1 S_2,\ S_1 \rightarrow b,\ bS_2 \rightarrow bb\}$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Se costruiamo $G_5$, si ha: $P_5 = \{S \to S_1 S_2,\ S_1 \to b,\ bS_2 \to bb\}$

  ed $L(G_5) \neq \emptyset$ in quanto $S \overset{*}{\underset{G_5}{\Rightarrow}} bb$ attraverso la seguente derivazione: $S \Rightarrow S_1 S_2 \Rightarrow bS_2 \Rightarrow bb$

Quale problema riscontriamo qui?

Le derivazioni che partono da $S_2$ **sono dipendenti da contesto**, e causano ‘interferenza’ nelle derivazioni che partono da $S_1$, che invece sono C.F.

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Se costruiamo $G_5$, si ha: $P_5 = \{S \to S_1 S_2,\ S_1 \to b,\ bS_2 \to bb\}$

  ed $L(G_5) \neq \varnothing$ in quanto $S \overset{*}{\underset{G_5}{\Rightarrow}} bb$ attraverso la seguente derivazione: $S \Rightarrow S_1 S_2 \Rightarrow bS_2 \Rightarrow bb$

Dunque la $G_5$ va bene solo per grammatiche di tipo '2' e risulta dimostrato che <u>$\mathcal{L}_2$ è chiusa rispetto alla concatenazione</u>.

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  La dimostrazione fatta non va bene per grammatiche di tipo ‘0’ e di tipo ‘1’, in quanto entrambe queste classi di grammatiche presentano produzioni dipendenti da contesto.

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  La dimostrazione fatta non va bene per grammatiche di tipo ‘0’ e di tipo ‘1’, in quanto entrambe queste classi di grammatiche presentano produzioni dipendenti da contesto.

  In presenza di grammatiche di tali tipi è **necessario impedire che le derivazioni da $S_2$ si servano di precedenti derivazioni da $S_1$ e/o viceversa**.

- **In altri termini, è necessario evitare interferenze tra derivazioni da $S_1$ e derivazioni da $S_2$ nella definizione dei contesti.** È possibile ottenere ciò considerando copie distinte di *X* in derivazioni distinte (da $S_1$ e da $S_2$), in modo che nessun *NT* derivato da $S_1$ possa far uso di parte di una forma di frase derivata da $S_2$ come contesto per l’applicazione di una produzione (e viceversa).

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Siano pertanto: $X' = \{x' \mid x \in X\}$ e $X'' = \{x'' \mid x \in X\}$

  due copie distinte di *X* tali che:

  $$X' \cap X'' = \varnothing \qquad X' \cap X = \varnothing \qquad X' \cap V = \varnothing$$

  $$X'' \cap X = \varnothing \qquad X'' \cap V = \varnothing$$

  Ad esempio, se nel mio insieme di terminali ho b, creo dei simboli b’ e b’’ ← attenzione, **sono non-terminali!**

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Siano pertanto: $X' = \{x' \mid x \in X\}$ e $X'' = \{x'' \mid x \in X\}$

  due copie distinte di *X* tali che:

  $$X' \cap X'' = \varnothing \qquad X' \cap X = \varnothing \qquad X' \cap V = \varnothing$$

  $$X'' \cap X = \varnothing \qquad X'' \cap V = \varnothing$$

  e sia $P_1'$ l'insieme delle produzioni ottenute da $P_1$ sostituendo ogni occorrenza di un terminale *x* in *X* con il corrispondente nonterminale $x'$ in $X'$:

  $$P_1' = P_1\left[{}^{x'}\!/_{x}\right]$$

  Similmente, sia $P_2''$ l'insieme delle produzioni ottenute da $P_2$ sostituendo ogni occorrenza di un terminale *x* in *X* con il corrispondente $x''$ in $X''$:

  $$P_2'' = P_2\left[{}^{x''}\!/_{x}\right]$$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  In questo modo evitiamo l'interferenza tra contesti.

  Costruiamo ora la grammatica: $G_6 = (X,\ V \cup X' \cup X'',\ S,\ P_6)$

  ove: $P_6 = \{S \to S_1 S_2\} \cup P_1' \cup P_2'' \cup \{x' \to x \mid x \in X\} \cup \{x'' \to x \mid x \in X\}$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  In questo modo evitiamo l'interferenza tra contesti.

  Costruiamo ora la grammatica: $G_6 = (X,\ V \cup X' \cup X'',\ S,\ P_6)$

  ove: $P_6 = \{S \to S_1 S_2\} \cup P_1' \cup P_2'' \cup \{x' \to x \mid x \in X\} \cup \{x'' \to x \mid x \in X\}$

  Sostanzialmente, per evitare le interferenze, **si riscrivono i terminali in non-terminali** e si aggiungono delle ulteriori produzioni che trasformino i non-terminali in terminali, alla fine del processo di concatenazione.

# Dimostrazione Teorema di chiusura

G$_1$ con $P_1 = \{S_1 \rightarrow b\}$
G$_2$ con $P_2 = \{bS_2 \rightarrow bb\}$.

- CONCATENAZIONE

Il controesempio visto in precedenza non dà più problemi:

$$P_1' = \{S_1 \rightarrow b'\} \qquad\qquad P_2'' = \{b''S_2 \rightarrow b''b''\}$$

e: $G_6 = (X,\ V \cup \{b'\} \cup \{b''\},\ S,\ P_6)$ dove:

# Dimostrazione Teorema di chiusura

G$_1$ con $P_1 = \{S_1 \to b\}$
G$_2$ con $P_2 = \{bS_2 \to bb\}$.

- CONCATENAZIONE

Il controesempio visto in precedenza non dà più problemi:

$$P_1' = \{S_1 \to b'\} \qquad P_2'' = \{b''S_2 \to b''b''\}$$

e: $G_6 = (X,\ V \cup \{b'\} \cup \{b''\},\ S,\ P_6)$ dove:

$$P_6 = \{S \to S_1S_2,\ S_1 \to b',\ b''S_2 \to b''b'',\ b' \to b,\ b'' \to b\}$$

ed $L(G_6) = \varnothing$ in quanto $S \overset{*}{\underset{G_6}{\not\Rightarrow}} bb$ , $S \Rightarrow S_1S_2 \Rightarrow b'S_2 \Rightarrow bS_2$

Risulta così dimostrato che <u>$\mathcal{L}_0$ ed $\mathcal{L}_1$ sono chiuse rispetto alla concatenazione</u>. Manca ancora <u>$\mathcal{L}_3$</u> !

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  È immediato osservare che, se $G_1$ e $G_2$ sono di tipo '3' né $G_5$ né $G_6$ sono di tipo '3', per la presenza della produzione $S \rightarrow S_1 S_2$.

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  È immediato osservare che, se $G_1$ e $G_2$ sono di tipo '3' né $G_5$ né $G_6$ sono di tipo '3', per la presenza della produzione $S \to S_1S_2$.

  **Dobbiamo simulare l'effetto della produzione** $S \to S_1S_2$ **che determina la concatenazione delle parole generate da $S_1$ e da $S_2$.** A tale scopo osserviamo che, data una grammatica di tipo '3', ogni forma di frase derivata dal simbolo iniziale di tale grammatica ha due peculiarità:

  **Quali?**

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

È immediato osservare che, se $G_1$ e $G_2$ sono di tipo ‘3’ né $G_5$ né $G_6$ sono di tipo ‘3’, per la presenza della produzione $S \to S_1 S_2$.

**Dobbiamo simulare l’effetto della produzione $S \to S_1 S_2$ che determina la concatenazione delle parole generate da $S_1$ e da $S_2$.** A tale scopo osserviamo che, data una grammatica di tipo ‘3’, ogni forma di frase derivata dal simbolo iniziale di tale grammatica ha due peculiarità:

(1) in essa compare al più un NT

(2) se in essa compare un NT, questo è il simbolo più a destra

$$\left( S \Rightarrow x_1 A \Rightarrow x_1 x_2 B \overset{*}{\Rightarrow} x_1 x_2 x_3 \ldots x_n N \Rightarrow x_1 x_2 x_3 \ldots x_n x_{n+1} \right)$$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Modifichiamo pertanto ogni produzione del tipo $A \to b$ in $G_1$ in modo che essa non costituisca l'ultima produzione applicata in una derivazione da $S_1$ in $G_1$, **ma possa innescare una derivazione da $S_2$ in $G_2$**.

Poniamo dunque a destra della *b* il simbolo iniziale $S_2$ di $G_2$.

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Modifichiamo pertanto ogni produzione del tipo $A \to b$ in $G_1$ in modo che essa non costituisca l'ultima produzione applicata in una derivazione da $S_1$ in $G_1$, **ma possa innescare una derivazione da $S_2$ in $G_2$**.

Poniamo dunque a destra della *b* il simbolo iniziale $S_2$ di $G_2$.

Le produzioni del tipo $A \to b$ in $G_1$ vengono trasformate in: $A \to bS_2$ . Costruiamo dunque la grammatica:

$$G_7 = (X,\ V - \{S\},\ S_1,\ P_7)$$

ove:
$$P_7 = \{A \to bB \mid A \to bB \in P_1\} \cup \{A \to bS_2 \mid A \to b \in P_1,\ b \neq \lambda\} \cup$$
$$\cup \{A \to bS_2 \mid A \to bB \in P_1,\ B \to \lambda \in P\} \cup \qquad (*)$$
$$\cup \{S_1 \to w \mid S_2 \to w \in P_2,\ S_1 \to \lambda \in P_1\} \cup \qquad (**)$$
$$\cup P_2$$

# Dimostrazione Teorema di chiusura

- CONCATENAZIONE

  Le (*) e (**) sono state introdotte per garantire la correttezza della grammatica generata, anche in presenza di $\lambda$-produzioni in $G_1$.

  $G_7$ è di tipo '3' se $G_1$ e $G_2$ lo sono ed inoltre:

$$L(G_7) = L_1 \cdot L_2$$

  ed <u>$\mathcal{L}_3$ è chiusa rispetto alla concatenazione</u>.

$$P_7 = \{A \to bB \mid A \to bB \in P_1\} \cup \{A \to bS_2 \mid A \to b \in P_1,\ b \neq \lambda\} \cup$$
$$\cup \{A \to bS_2 \mid A \to bB \in P_1,\ B \to \lambda \in P\} \cup \qquad (*)$$
$$\cup \{S_1 \to w \mid S_2 \to w \in P_2,\ S_1 \to \lambda \in P_1\} \cup \qquad (**)$$
$$\cup P_2$$

# Dimostrazione Teorema di chiusura

- ITERAZIONE

  Costruiamo la grammatica:

$$G_8 = (X,\ V_1 \cup \{S\},\ S,\ P_8)$$

  ove:

$$P_8 = \{S \rightarrow \lambda,\ S \rightarrow S_1 S\} \cup P_1$$

# Dimostrazione Teorema di chiusura

- ITERAZIONE

  Costruiamo la grammatica:

$$G_8 = (X,\ V_1 \cup \{S\},\ S,\ P_8)$$

  ove:

$$P_8 = \{S \rightarrow \lambda,\ \ S \rightarrow S_1 S\} \cup P_1$$

  Data la grammatica $G_1$ = ($X$, $V_1$, $S_1$, $P_1$) che genera $L_1$, la grammatica $G_8$ genera la parola vuota $\lambda$ e tutte le parole che si possono ottenere per concatenazione di parole generate da $S_1$ in $G_1$. Si ha infatti:

$$S \overset{n}{\underset{G_8}{\Rightarrow}} \underbrace{S_1 S_1 \ldots S_1}_{n} S \Rightarrow S_1 S_1 \ldots S_1 \overset{*}{\Rightarrow} w_1 w_2 \ldots w_n$$

  con: $w_i \in L_1,\ \ i = 1, 2, \ldots, n$

# Dimostrazione Teorema di chiusura

- ITERAZIONE

  Vediamo per quali classi di grammatiche la $G_8$ è la grammatica che stiamo cercando:
  - se $G_1$ è di tipo '3', $G_8$ non è di tipo '3', perché non è lin. dx;

# Dimostrazione Teorema di chiusura

- ITERAZIONE

  Vediamo per quali classi di grammatiche la $G_8$ è la grammatica che stiamo cercando:

  - se $G_1$ è di tipo ‘3’, $G_8$ non è di tipo ‘3’, perché non è lin. dx;
  - se $G_1$ è di tipo ‘2’, $G_8$ è di tipo ‘2’ e si ha: $L(G_8) = L_1^*$
    e risulta dimostrato che $\mathcal{L}_2$ è chiusa rispetto all’iterazione.

# Dimostrazione Teorema di chiusura

- ITERAZIONE

  Vediamo per quali classi di grammatiche la $G_8$ è la grammatica che stiamo cercando:

  - se $G_1$ è di tipo ‘3’, $G_8$ non è di tipo ‘3’, perché non è lin. dx;
  - se $G_1$ è di tipo ‘2’, $G_8$ è di tipo ‘2’ e si ha: $L(G_8) = L_1^*$
    e risulta dimostrato che <u>$\mathcal{L}_2$ è chiusa rispetto all’iterazione.</u>
  - se $G_1$ è di tipo ‘1’, $G_8$ non è di tipo ‘1’, perché $S$ compare nella parte destra della produzione $S \to S_1S$ ed $S \to \lambda$ è una produzione in $P_8$;

# Dimostrazione Teorema di chiusura

- ITERAZIONE

  Vediamo per quali classi di grammatiche la $G_8$ è la grammatica che stiamo cercando:

  - se $G_1$ è di tipo '3', $G_8$ non è di tipo '3', perché non è lin. dx;
  - se $G_1$ è di tipo '2', $G_8$ è di tipo '2' e si ha: $L(G_8) = L_1^*$
    e risulta dimostrato che <u>$\mathcal{L}_2$ è chiusa rispetto all'iterazione.</u>
  - se $G_1$ è di tipo '1', $G_8$ non è di tipo '1', perché $S$ compare nella parte destra della produzione $S \rightarrow S_1S$ ed $S \rightarrow \lambda$ è una produzione in $P_8$;

    Possiamo trasformare facilmente $G_8$ nella grammatica equivalente $G'_8$, definita come segue:

$$G'_8 = (X,\ V_1 \cup \{S, S'\},\ S,\ P'_8)$$

$$P'_8 = \{S \rightarrow \lambda \mid S',\ S' \rightarrow S_1 \mid S_1S'\} \cup P_1$$

ma $G'_8$ incorre nello stesso problema di interferenza nella definizione dei contesti visto per la concatenazione.

# Dimostrazione Teorema di chiusura

- ITERAZIONE

  Controesempio:

  Consideriamo $G_1$ con $P_1 = \{S_1 \to b,\ bS_1 \to bc\}$ allora
  $L_1 = L(G) = \{b\}$ e $L_1^* = \{b\}^*$

  Del resto: $P_8 = \left\{ S \overset{(1)}{\to} \lambda \mid \overset{(2)}{S'},\ S' \overset{(3)}{\to} S_1 \mid \overset{(4)}{S_1 S'},\ S_1 \overset{(5)}{\to} b,\ bS_1 \overset{(6)}{\to} bc \right\}$

  Se consideriamo la derivazione: $S \underset{(2)}{\Rightarrow} S' \underset{(4)}{\Rightarrow} S_1 S' \underset{(3)}{\Rightarrow} S_1 S_1 \underset{(5)}{\Rightarrow} bS_1 \underset{(6)}{\Rightarrow} bc$

  si ottiene che: $bc \in L(G_8')$ e quindi $L(G_8') \neq L_1^*$

  Il problema dell'interferenza dei contesti può essere quindi risolto come segue.

# Dimostrazione Teorema di chiusura

- ITERAZIONE
  Eliminiamo dapprima le produzioni del tipo $S_1 \rightarrow \lambda$
  Utilizziamo gli insiemi ausiliari $X'$ e $X''$ di *NT* per evitare interferenze:

$$X' = \{x' \mid x \in X\} \qquad X'' = \{x'' \mid x \in X\}$$

$$X' \cap X'' = \emptyset \qquad X' \cap X = \emptyset \qquad X' \cap V_1 = \emptyset$$

$$X'' \cap X = \emptyset \qquad X'' \cap V_1 = \emptyset$$

# Dimostrazione Teorema di chiusura

- ITERAZIONE
  Eliminiamo dapprima le produzioni del tipo $S_1 \to \lambda$
  Utilizziamo gli insiemi ausiliari $X'$ e $X''$ di *NT* per evitare interferenze:

$$X' = \{x' \mid x \in X\} \qquad X'' = \{x'' \mid x \in X\}$$

$$X' \cap X'' = \varnothing \qquad X' \cap X = \varnothing \qquad X' \cap V_1 = \varnothing$$

$$X'' \cap X = \varnothing \qquad X'' \cap V_1 = \varnothing$$

- Costruiamo due copie di $G_1$:

$$G_1' = (X,\ V_1 \cup X',\ S_1,\ P_1')$$

$$G_1'' = (X,\ V_1 \cup X'',\ S_2,\ P_1'')$$

ove:

$$P_1' = P_1\left[{x'}\big/{x}\right] \qquad P_1'' = P_1\left[{x''}\big/{x}\right]$$

# Dimostrazione Teorema di chiusura

- ITERAZIONE
  Infine, combiniamo $G_1'$, $G_1''$ con produzioni che costruiscono sequenze finite di copie di $S_1$ ed $S_2$ che si alternano, in modo da ottenere $L_1^*$.
  Dunque, la grammatica che otteniamo è:
  $$G_9 = (X,\ V_1 \cup X' \cup X'' \cup \{S, S_1', S_2, S_2'\},\ S,\ P_9)$$
  ove:
  $$P_9 = \{S \to \lambda \mid S_1' \mid S_2',\ S_1' \to S_1 \mid S_1 S_2',\ S_2' \to S_2 \mid S_2 S_1'\} \cup$$
  $$\cup P_1' \cup P''_1 \cup \{x' \to x \mid x \in X\} \cup \{x'' \to x \mid x \in X\}$$
  Se $G_1$ è di tipo $i$, $i$ = 0,1, lo è anche $G_9$ e si ha:
  $$L(G_9) = L_1^*$$
  e risulta dimostrato che <u>$\mathcal{L}_0$ e $\mathcal{L}_1$ sono chiuse rispetto all'iterazione.</u>

# Dimostrazione Teorema di chiusura

- ITERAZIONE
  Resta da dimostrare che $\mathcal{L}_3$ è chiusa rispetto all'iterazione.
  Per costruire la nuova grammatica, introduciamo dapprima un nuovo simbolo iniziale *S* e la produzione $S \to \lambda$ che genera la stringa vuota.
  Inoltre, eliminiamo da $P_1$, se c'era, la produzione $S_1 \to \lambda$ ed aggiungiamo una produzione $S \to w$ per ogni produzione "iniziale" $S_1 \to w$ in $P_1$.

# Dimostrazione Teorema di chiusura

- ITERAZIONE
  Infine, per ogni produzione la cui parte destra contiene solo un terminale, del tipo $A \to b$, nell'insieme delle produzioni che stiamo costruendo, aggiungiamo la produzione $A \to bS$
  **A cosa serve questa produzione?**

# Dimostrazione Teorema di chiusura

- ITERAZIONE
  Infine, per ogni produzione la cui parte destra contiene solo un terminale, del tipo $A \rightarrow b$, nell'insieme delle produzioni che stiamo costruendo, aggiungiamo la produzione $A \rightarrow bS$
  **A cosa serve questa produzione?**

- **Serve a innescare l'iterazione quando abbiamo terminato la derivazione della parola attraverso il terminale 'b'**
  - **La regola di produzione è ancora valida per una grammatica lineare destra.**

# Dimostrazione Teorema di chiusura

- ITERAZIONE
  Si noti che, per garantire la correttezza della grammatica generata anche in presenza di $\lambda$-produzioni in $G_1$, occorre aggiungere una produzione $A \to bS$ per ogni produzione $A \to bB$ nell'insieme delle produzioni che stiamo costruendo, quando $B \to \lambda$ è pure una produzione di tale insieme, alla stregua di quanto fatto per la concatenazione in $\mathcal{L}_3$.

# Dimostrazione Teorema di chiusura

- ITERAZIONE

  Costruiamo dunque la grammatica: $G_{10} = (X,\ V_1 \cup \{S\},\ S,\ P_{10})$

- ove: $$P_{10} = \{S \to \lambda\} \cup \left(P_1 - \{S_1 \to \lambda\}\right) \cup \{S \to w \mid S_1 \to w \in P_1\} \cup$$
$$\cup \{A \to bS \mid A \to b \in P_{10},\ b \neq \lambda\} \cup$$
$$\cup \{A \to bS \mid A \to bB \in P_{10},\ B \to \lambda \in P_{10}\}$$

- Si noti che la definizione di $P_{10}$ è ricorsiva. Se $G_1$ è di tipo '3', lo è anche $G_{10}$ e si ha: $L(G_{10}) = L_1^*$

- e risulta dimostrato che <u>$\mathcal{L}_3$ è chiusa rispetto all'iterazione.</u>

# Teorema di chiusura

| | *UNIONE* | *CONCATENAZIONE* | *ITERAZIONE* |
|---|---|---|---|
| $\mathcal{L}_0$ | $G_3=(X,V,S,P_3)$<br>$P_3=\{S\to S_1,\ S\to S_2\}\cup$<br>$\cup P_1\cup P_2$ | $X'=\{x'\mid x\in X\}$ $X''=\{x''\mid x\in X\}$<br>$X'\cap X''=\varnothing$ $X'\cap X=\varnothing$ $X'\cap V=\varnothing$<br>$X''\cap X=\varnothing$ $X''\cap V=\varnothing$<br>$P_1'=P_1\left[x'/x\right]$ $P_2''=P_2\left[x''/x\right]$<br>$G_6=(X,V\cup X'\cup X'',S,P_6)$<br>$P_6=\{S\to S_1S_2\}\cup P_1'\cup P_2''\cup$<br>$\cup\{x'\to x\mid x\in X\}\cup$<br>$\cup\{x''\to x\mid x\in X\}$ | Eliminiamo le produzioni del tipo:<br>$S_1\to\lambda$<br>$X'=\{x'\mid x\in X\}$ $X''=\{x''\mid x\in X\}$<br>$X'\cap X''=\varnothing$ $X'\cap X=\varnothing$ $X'\cap V_1=\varnothing$<br>$X''\cap X=\varnothing$ $X''\cap V_1=\varnothing$<br>Costruiamo due copie di $G_1$:<br>$G_1'=(X,V_1\cup X',S_1,P_1')$<br>$G_1''=(X,V_1\cup X'',S_2,P_1'')$<br>$P_1'=P_1\left[x'/x\right]$ $P_1''=P_1\left[x''/x\right]$<br>$G_9=(X,V_1\cup X'\cup X''\cup\{S,S_1',S_2,S_2'\},S,P_9)$<br>$P_9=\{S\to\lambda\mid S_1'\mid S_2',\ S_1'\to S_1\mid S_1S_2',\ S_2'\to S_2\mid S_2S_1'\}\cup$<br>$\cup P_1'\cup P_2'\cup\{x'\to x\mid x\in X\}\cup\{x''\to x\mid x\in X\}$ |
| $\mathcal{L}_1$ | | | |
| $\mathcal{L}_2$ | | $G_5=(X,V,S,P_5)$<br>$P_5=\{S\to S_1S_2\}\cup P_1\cup P_2$ | $G_8=\{X,V_1\cup\{S\},S,P_8\}$<br>$P_8=\{S\to\lambda\mid S_1S\}\cup P_1$ |
| $\mathcal{L}_3$ | $G_4=(X,V,S,P_4)$<br>$P_4=\{S\to w\mid S_1\to w\in P_1\}\cup$<br>$\cup\{S\to w\mid S_2\to w\in P_2\}\cup$<br>$\cup P_1\cup P_2$ | $G_7=(X,V-\{S\},S_1,P_7)$<br>$P_7=\{A\to bB\mid A\to bB\in P_1\}\cup$<br>$\cup\{A\to bS_2\mid A\to b\in P_1,\ b\neq\lambda\}\cup$<br>$\cup\{A\to bS\mid A\to bB\in P_1,\ B\to\lambda\in P_1\}\cup$<br>$\cup\{S_1\to w\mid S_2\to w\in P_2,\ S_1\to\lambda\in P_1\}\cup P_2$ | $G_{10}=(X,V_1\cup\{S\},S,P_{10})$<br>$P_{10}=\{S\to\lambda\}\cup\left(P_1-\{S_1\to\lambda\}\right)\cup$<br>$\cup\{S\to w\mid S_1\to w\in P_1\}\cup$<br>$\cup\{A\to bS\mid A\to b\in P_{10},\ b\neq\lambda\}\cup$<br>$\cup\{A\to bS\mid A\to bB\in P_{10},\ B\to\lambda\in P_{10}\}$ |

# Altri teoremi di chiusura

- La classe dei linguaggi lineari destri (tipo ‘3’) è chiusa rispetto al complemento ed all’intersezione.
- La classe dei linguaggi non contestuali (tipo ‘2’) **non è chiusa** rispetto al complemento ed all’intersezione.
- La classe dei linguaggi contestuali (tipo ‘1’) è chiusa rispetto al complemento (e dunque anche rispetto all’intersezione).
- La classe dei linguaggi di tipo ‘0’ non è chiusa rispetto al complemento.

# Ricapitolando

- **Tipo 1 e Tipo 3**
  - Chiusi rispetto a intersezione e complemento
- **Tipo 2**
  - Non chiuso rispetto a intersezione e complemento
  - Perché questo risultato è interessante?
    - Perché definendo un linguaggio L come intersezione o complemento di due linguaggi più semplici (L1 e L2) di tipo 2, possiamo applicare il teorema di chiusura per dimostrare **che il linguaggio intersezione non è di tipo 2.**
    - **Schema di dimostrazione alternativo rispetto al pumping lemma!**

# Esempio

- La classe dei linguaggi non contestuali (tipo '2') non è chiusa rispetto al complemento ed all'intersezione.

  Consideriamo i linguaggi
  $L_1 = \{a^n b^n c^m \mid n, m > 0\}$ $L_2 = \{a^n b^m c^m \mid n, m > 0\}$
  $L_1$ e $L_2$ sono linguaggi liberi, mentre $L_1 \cap L_2$ non lo è:

  Il complemento è: $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$
  Dunque, se $L_1, L_2 \in \mathcal{L}_2$ e se $\mathcal{L}_2$ fosse chiusa rispetto al complemento, …

# Esempio

- La classe dei linguaggi non contestuali (tipo ‘2’) non è chiusa rispetto al complemento ed all’intersezione.

  Consideriamo i linguaggi
  $L_1 = \{a^n b^n c^m \mid n,m > 0\}$ $\qquad$ $L_2 = \{a^n b^m c^m \mid n,m > 0\}$
  *A cosa corrisponde* $L_1 \cap L_2$ *?*

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

- *Poiché L1 ed L2 sono linguaggi liberi da contesto (trovate le grammatiche!) si può affermare che il linguaggio intersezione non sarà di tipo 2 (dunque sarà di tipo 1).*

# Esempio

- La classe dei linguaggi non contestuali (tipo ‘2’) non è chiusa rispetto al complemento ed all’intersezione.

  Consideriamo i linguaggi
  $L_1 = \{a^n b^n c^m \mid n, m > 0\}$ $L_2 = \{a^n b^m c^m \mid n, m > 0\}$
  *A cosa corrisponde* $L_1 \cap L_2$ *?*

# Dimostrazione

- La classe dei linguaggi lineari destri (tipo ‘3’) è chiusa rispetto al complemento ed all’intersezione.

  Dimostriamola rispetto all’intersezione:

  $$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}} \quad \text{(Leggi di De Morgan)}$$

  La chiusura di $\mathscr{L}_3$ rispetto all’intersezione discende direttamente da questo risultato.

# Dimostrazione

- La classe dei linguaggi contestuali (tipo ‘1’) è chiusa rispetto al complemento (e dunque anche rispetto all’intersezione).

  Un risultato recente ha stabilito che $L_1$ è chiusa rispetto al complemento (e quindi all’intersezione). Non se ne conosce la dimostrazione.
- La classe dei linguaggi di tipo ‘0’ non è chiusa rispetto al complemento.

  Non lo dimostriamo

# L'operazione di riflessione

- Definizione di stringa riflessa
  Sia *w* una parola su un alfabeto $X = \{x_1, x_2, \ldots, x_k\}$, $w = x_{i_1} x_{i_2} \ldots x_{i_{n-1}} x_{i_n}$. Dicesi *stringa riflessa* (o *riflessione*) di *w* la stringa

$$w^R = x_{i_n} x_{i_{n-1}} \ldots x_{i_2} x_{i_1}$$

- Operazione di riflessione
  Sia *w* una parola su un alfabeto $X = \{x_1, x_2, \ldots, x_k\}$ e sia $w^R$ la stringa riflessa di *w*. L'operazione che trasforma *w* in $w^R$ è detta *operazione di riflessione*.

# L'operazione di riflessione

- Definizione di parola palindromica
  Un *palindromo* (o *parola palindromica*) è una parola la cui lettura a ritroso riproduce la parola di partenza:

$$w \ \text{palindromo} \overset{def}{\Leftrightarrow} w = w^R$$

  Un palindromo è dunque una parola che coincide con la sua riflessione.

# Esempio

- Alcuni palindromi sull'alfabeto $\{a, b, ..., z\}$ sono:
  - *a*;
  - *ii* (plurale di io?);
  - *non, ala, ara, ici*;
  - *osso, alla, arra*;
  - *radar, alalà, arerà* (ignorando l'accento);
  - *ossesso, ingegni*;
  - *avallava, ovattavo*;
  - *onorarono*;
  - *accavallavacca*, *accumolomucca*;
  - *feci nulla all'Unicef*, *ogni tela male tingo* (ignorando spazi bianchi, punteggiatura e differenza tra maiuscole e minuscole).

# Palindromi

- I palindromi (su un qualunque alfabeto) sono di due tipi:
  - palindromi di lunghezza pari: hanno un “asse di simmetria” costituito dalla parola vuota
  - palindromi di lunghezza dispari: hanno un “asse di simmetria” costituito da uno dei simboli dell’alfabeto

- Più precisamente, si ha la seguente caratterizzazione (senza dimostrazione):

# Teorema

- Sia *w* una parola su un alfabeto *X*. *w* è palindromo se e solo se

$$w = \alpha x \alpha^R, \;\; x \in X \cup \{\lambda\}$$

# Teorema

- La classe dei linguaggi non contestuali (tipo ‘2’) è chiusa rispetto all’operazione di riflessione.

  Dimostrazione
  Sia $G_1 = (X, V_1, S_1, P_1)$ una grammatica non contestuale.
  Dobbiamo dimostrare che:
  $$L(G_1) \text{ non contestuale} \Rightarrow \left(L(G_1)\right)^R = \{w^R \mid L(G_1)\}$$
  è non contestuale.
  Costruiamo la grammatica: $G_{11} = (X, V_1, S_1, P_{11})$
  ove: $P_{11} = \{A \to w^R \mid A \to w \in P_1\}$
  Risulta allora: $L(G_{11}) = \left(L(G_1)\right)^R$
  Quindi se in $\underline{P_1}$ abbiamo la produzione: $A \to BaC$
  in $\underline{P_{11}}$ avremo la produzione: $A \to CaB$

# Utilizzo proprietà di chiusura

| *Schema di Ragionamento per Utilizzo Proprietà di Chiusura* | |
|---|---|
| Siano $L$, $L_1$ ed $L_2$ tre linguaggi tali che: $L = \alpha(L_1, L_2)$ ove $\alpha = \cup, \cdot$ | |
| ***Esatto*** | ***Errato*** |
| Supponiamo che $L_2 \in \mathcal{L}_i$<br>Se $\mathrm{L} \notin L_i$ allora $\mathrm{L}_1 \notin L_i$ | Se $\underset{j=1,2}{L_j} \notin \mathcal{L}_i$ *allora* $L \notin \mathcal{L}_i$ |

Esempio: $\underset{n,m>0}{a^n b^m} = a^n b^n \cup \underset{n \neq m}{a^n b^m}$

Lineare dx    Non sono lineari dx