Linguaggi di Programmazione
Docente: Cataldo Musto

Capitolo 2 – Grammatiche e Linguaggi

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

Linguaggi formali e monoidi liberi

- Il concetto di linguaggio formale è strettamente correlato a quello di *monoide libero* (generato da un insieme).

Linguaggi formali e monoidi liberi

- Il concetto di linguaggio formale è strettamente correlato a quello di *monoide libero* (generato da un insieme).

- **Recap**
  - **Semigruppo:** struttura algebrica che verifica le proprietà di chiusura e di associatività
    - Esempi di semigruppi: **(N,+) (N,*) ….**
  - **Monoide:** semigruppo dotato anche di elemento neutro
    - Esempio di monoide: **(N,+) (N,*) ….**

Definizione di Alfabeto

- Un insieme *X* finito e non vuoto di simboli è un *alfabeto*.
- **Esempi**
  - L'alfabeto latino, con l'aggiunta dei simboli di interpunzione e dello spazio bianco: *a b c …z* ; , . :
  - L'insieme delle dieci cifre arabe: 0 1 … 9
- Con i simboli primitivi dell'alfabeto si formano le parole (es.: *abc*, 127, *casa*,…).

Definizione di Parola o Stringa

- Una sequenza finita di simboli $x_1 x_2 \ldots x_n$, dove ogni x_i è preso da uno stesso alfabeto *X* è una *parola* (su *X*).
- **Come si ottiene una parola, dato un alfabeto?**

- Esempio
  - $X = \{0,1\}$.
  - 001110 è una parola su *X*

Definizione di Parola o Stringa

- Una sequenza finita di simboli $x_1 x_2 \ldots x_n$, dove ogni x_i è preso da uno stesso alfabeto X è una *parola* (su X).
- **Come si ottiene una parola, dato un alfabeto?**

- Esempio
  - $X = \{0,1\}$.
  - 001110 è una parola su X
- **Una parola è ottenuta giustapponendo o concatenando simboli (caratteri) dell'alfabeto.**
  - Se una stringa ha m simboli (non necessariamente distinti) allora diciamo che ha lunghezza m.

Lunghezza di una Parola o Stringa

- La lunghezza di una stringa *w* è denotata con |*w*|.
- Le parole di lunghezza 1 sono i simboli di *X*.
  - Quindi 001110 è una parola di lunghezza 6

$$|001110| = 6$$

- La parola vuota (o stringa vuota), denotata con λ, è una stringa priva di simboli ed ha lunghezza 0

$$|\lambda| = 0$$

- Uguaglianza tra stringhe
  - Due stringhe sono *uguali* se i loro caratteri, letti ordinatamente da sinistra a destra, coincidono.

Definizioni

- *X**
  - L'insieme di tutte le stringhe di lunghezza finita sull'alfabeto *X* si denota con *X**.

- Esempio
  - Se $X = \{0,1\}$, allora $X^* = \{\lambda, 0, 1, 00, 01, 10, 11, \ldots\}$

- *X** ha un numero di elementi che è un infinito numerabile.
  - Dalla definizione, segue che $\lambda \in X^*$, per ogni insieme *X*.

Definizioni

- Concatenazione o prodotto
  - Sia $\alpha \in X^*$ una stringa di lunghezza *m* e $\beta \in X^*$ una stringa di lunghezza *n*, la *concatenazione* di α e β, denotata con $\alpha\beta$ o $\alpha \cdot \beta$, è definita come la stringa di lunghezza *m*+*n*, i cui primi *m* simboli costituiscono una stringa uguale a α ed i cui ultimi *n* simboli costituiscono una stringa uguale a β.

Definizioni

- Concatenazione o prodotto
  - Sia $\alpha \in X^*$ una stringa di lunghezza *m* e $\beta \in X^*$ una stringa di lunghezza *n*, la *concatenazione* di α e β, denotata con $\alpha\beta$ o $\alpha \cdot \beta$, è definita come la stringa di lunghezza *m*+*n*, i cui primi *m* simboli costituiscono una stringa uguale a α ed i cui ultimi *n* simboli costituiscono una stringa uguale a β.
  - Quindi se $\alpha = x_1 x_2 \ldots x_m$ e $\beta = x'_1 x'_2 \ldots x'_n$, si ha:

$$\alpha\beta = x_1 x_2 \ldots x_m x'_1 x'_2 \ldots x'_n$$

  - Se X = alfabeto latino

$$\alpha = capo \quad \beta = stazione \qquad \alpha\beta = capostazione$$

Operazione di concatenazione

- La concatenazione di stringhe su *X* è una operazione binaria su *X**:

$$\cdot\ :\ X^* \times X^* \to X^*$$

Operazione di concatenazione

- La concatenazione di stringhe su *X* è una operazione binaria su *X**:

$$\cdot \; : \; X^* \times X^* \to X^*$$

  - è associativa: $(\alpha\beta)\gamma = \alpha(\beta\gamma) = \alpha\beta\gamma, \quad \forall \alpha, \beta, \gamma \in X^*$
  - non è commutativa: $\exists \alpha, \beta \in X^* \; : \; \alpha\beta \neq \beta\alpha$
    - *capostazione ≠ stazionecapo*
  - ha elemento neutro λ: $\lambda\alpha = \alpha\lambda = \alpha, \quad \forall \alpha \in X^*$

- Dunque $\left(X^*, \cdot\,\right)$ è un monoide (non commutativo).

Osservazione

- In base alla definizione di prodotto, ogni parola non vuota $\alpha = x_1 x_2 \ldots x_n$ si può scrivere in uno ed un solo modo come prodotto di parole di lunghezza 1, cioè di elementi di *X*.

- Ciò si esprime dicendo che:
  - $\left(X^*, \cdot\right)$ è il **monoide libero generato** dall'insieme *X*.

Definizioni

- Prefisso, Suffisso
  - Se $\gamma \in X^*$ è della forma $\gamma = \alpha\beta$, ove $\alpha, \beta \in X^*$, allora α è un *prefisso* di γ e β è un *suffisso* di γ.

Definizioni

- Prefisso, Suffisso
  - Se $\gamma \in X^*$ è della forma $\gamma = \alpha\beta$, ove $\alpha, \beta \in X^*$, allora α è un *prefisso* di γ e β è un *suffisso* di γ.
- Sottostringa
  - Se $\delta, \beta \in X^*$ e δ è della forma $\delta = \alpha\beta\gamma$, ove $\alpha, \beta \in X^*$ allora β è una *sottostringa* di δ.
- Esempio
  - Sia γ = *00110*
  - **Proviamo a costruire i relativi insiemi.**

Definizioni

- Prefisso, Suffisso
  - Se $\gamma \in X^*$ è della forma $\gamma = \alpha\beta$, ove $\alpha, \beta \in X^*$, allora α è un *prefisso* di γ e β è un *suffisso* di γ.
- Sottostringa
  - Se $\delta, \beta \in X^*$ e δ è della forma $\delta = \alpha\beta\gamma$, ove $\alpha, \beta \in X^*$ allora β è una *sottostringa* di δ.
- Esempio
  - Sia γ = *00110*. Allora:
    $\{\lambda, 0, 00, 001, 0011, \gamma\}$ è l'insieme dei prefissi di γ

Definizioni

- Prefisso, Suffisso
  - Se $\gamma \in X^*$ è della forma $\gamma = \alpha\beta$, ove $\alpha, \beta \in X^*$, allora α è un *prefisso* di γ e β è un *suffisso* di γ.
- Sottostringa
  - Se $\delta, \beta \in X^*$ e δ è della forma $\delta = \alpha\beta\gamma$, ove $\alpha, \beta \in X^*$ allora β è una *sottostringa* di δ.
- Esempio
  - Sia γ = *00110*. Allora:
    $\{\lambda, 0, 00, 001, 0011, \gamma\}$ è l'insieme dei prefissi di γ
    $\{\lambda, 0, 10, 110, 0110, \gamma\}$ è l'insieme dei suffissi di γ

Definizioni

- Prefisso, Suffisso
  - Se $\gamma \in X^*$ è della forma $\gamma = \alpha\beta$, ove $\alpha, \beta \in X^*$, allora α è un *prefisso* di γ e β è un *suffisso* di γ.
- Sottostringa
  - Se $\delta, \beta \in X^*$ e δ è della forma $\delta = \alpha\beta\gamma$, ove $\alpha, \beta \in X^*$ allora β è una *sottostringa* di δ.
- Esempio
  - Sia γ = *00110*. Allora:

    $\{\lambda, 0, 00, 001, 0011, \gamma\}$ è l'insieme dei prefissi di γ

    $\{\lambda, 0, 10, 110, 0110, \gamma\}$ è l'insieme dei suffissi di γ

    $\{\lambda, 0, 1, 00, 01, 10, 11, 001, 011, 110, 0011, 0110, \gamma\}$ è l'insieme delle sottostringhe di γ.

Definizioni

- Potenza di una stringa
  - Data una stringa α su X, la *potenza h-esima* di α è definita (induttivamente) come segue:

$$\alpha^h = \begin{cases} \lambda & \text{se } h = 0 \\ \alpha\alpha^{h-1} & \text{altrimenti} \end{cases}$$

    con h = 0, 1, 2, …

  - La potenza *h*-esima di una stringa è un caso speciale di concatenamento (in quanto la si ottiene concatenando una stringa *h* volte con se stessa).

Definizioni

- Potenza di un alfabeto
  - Sia *X* un alfabeto, poniamo:

    1) $X^1 = X$

    2) $X^2 = \{x_1 x_2 \mid x_1, x_2 \in X,\ \ x_1 x_2 \equiv x_1 \cdot x_2\}$

La potenza di un alfabeto (X^k) è l'insieme di tutte le **stringhe di lunghezza k**

Definizioni

- Potenza di un alfabeto
  - Sia X un alfabeto, poniamo:

    1) $X^1 = X$

    2) $X^2 = \{x_1 x_2 \mid x_1, x_2 \in X,\ x_1 x_2 \equiv x_1 \cdot x_2\}$

    3) $X^3 = \{x_1 x_2 x_3 \mid x_1 x_2 \in X^2,\ x_3 \in X,\ x_1 x_2 x_3 \equiv x_1 x_2 \cdot x_3\}$

    ..) ....

    i) $X^i = \{x_1 x_2 \ldots x_{i-1} x_i \mid x_1 x_2 \ldots x_{i-1} \in X^{i-1},\ x_i \in X,\ x_1 x_2 \ldots x_i \equiv x_1 x_2 \ldots x_{i-1} \cdot x_i\}$

  - La potenza di un alfabeto (X^k) è l'insieme di tutte le **stringhe di lunghezza k**

Definizioni

- Potenza di un alfabeto
  - Esempio
    - $X = \{0,1\}$
    - $X^2=$
    - $X^3=$

Definizioni

- Potenza di un alfabeto
  - Esempio
    - $X = \{0,1\}$
    - $X^2 = \{00,01,10,11\}$
    - $X^3 =$

Definizioni

- Potenza di un alfabeto
  - Esempio
    - $X = \{0,1\}$
    - $X^2 = \{00,01,10,11\}$
    - $X^3 = \{000,001,010,011$
      $100,101,110,111\}$

Definizioni

- **Potenza di un alfabeto**
  - Definizione

$$X^h = \begin{cases} \{\lambda\} & \text{se } h = 0 \\ X \cdot X^{h-1} & \text{altrimenti} \end{cases}$$

- $X = \{0,1\}$
- $X^2 = \{00,01,10,11\}$
- $X^3 = \{000,001,010,011$
  $100,101,110,111\}$

Definizioni

- **Potenza di un alfabeto**
  - Definizione

$$X^h = \begin{cases} \{\lambda\} & \text{se } h = 0 \\ X \cdot X^{h-1} & \text{altrimenti} \end{cases}$$

  - Se $i \geq 2$ si ha:

$$X^+ = X \cup X^2 \cup \ldots \cup X^i \cup \ldots = \bigcup_{i=1}^{+\infty} X^i$$

Definizioni

- **Potenza di un alfabeto**
  - Se $i \geq 2$ si ha:

$$X^{+} = X \cup X^{2} \cup \ldots \cup X^{i} \cup \ldots = \bigcup_{i=1}^{+\infty} X^{i}$$

  - Se λ è la parola vuota e prendiamo un $w \in X^{+}$ tale che $w \cdot \lambda = \lambda \cdot w = w$ si ha:

$$X^{*} = \{\lambda\} \cup X^{+}$$

Definizioni

- **Potenza di un alfabeto**
  - Se $i \geq 2$ si ha:

$$X^+ = X \cup X^2 \cup \ldots \cup X^i \cup \ldots = \bigcup_{i=1}^{+\infty} X^i$$

  - Se λ è la parola vuota e prendiamo un $w \in X^+$ tale che $w \cdot \lambda = \lambda \cdot w = w$ si ha:

$$X^* = \{\lambda\} \cup X^+$$

X^* è l'insieme di tutte le **stringhe di lunghezza arbitraria costruite a partire da X**

Definizioni

- Linguaggio formale
  - Un *linguaggio formale* *L* su un alfabeto *X* è un sottoinsieme di *X**.

$$L \subseteq X^*$$

Definizioni

- Linguaggio formale
  - Un *linguaggio formale* *L* su un alfabeto *X* è un sottoinsieme di *X**.

$$L \subseteq X^*$$

- Esempio:
  - Il linguaggio delle parentesi ben formate è un linguaggio formale in quanto, denotato con *M* tale linguaggio, si ha:

$$M \subset \{ (\,,) \}^*$$

  - I linguaggi formali possono essere di natura molto diversa l'uno dall'altro.

Esempi di linguaggi formali

- **Un programma per un linguaggio di programmazione** può essere costruito a partire dall'alfabeto *X* dei simboli sulla tastiera.
  - L'insieme, finito o infinito, dei programmi ben costruiti sintatticamente (ossia, che rispettano la sintassi) costituisce un linguaggio.

Esempi di linguaggi formali

- **Un programma per un linguaggio di programmazione** può essere costruito a partire dall'alfabeto *X* dei simboli sulla tastiera.
  - L'insieme, finito o infinito, dei programmi ben costruiti sintatticamente (ossia, che rispettano la sintassi) costituisce un linguaggio.
- A partire dai simboli del nostro linguaggio posso anche costruire il linguaggio della chimica
  - Formule chimiche corrette si ottengono attraverso la concatenazione di numeri e simboli del nostro linguaggio.

Generazione e riconoscimento di linguaggi formali

- A noi interessano i linguaggi formali da almeno due *punti di vista*
  - Descrittivo/Generativo
  - Riconoscitivo

Generazione e riconoscimento di linguaggi formali

- *Punto di vista* Descrittivo/Generativo
  - Come possiamo *generare* gli elementi di un dato linguaggio *L*?
    - Un linguaggio finito può essere descritto/generato per elencazione degli elementi (se il numero non è troppo grande).
    - **E se il linguaggio è infinito?**

Generazione e riconoscimento di linguaggi formali

- *Punto di vista* Descrittivo/Generativo
  - Come possiamo *generare* gli elementi di un dato linguaggio L?
    - Un linguaggio finito può essere descritto/generato per elencazione degli elementi (se il numero non è troppo grande).
    - **Un linguaggio infinito non è elencabile.**
      - I linguaggi infiniti sono i più interessanti, perché devono essere specificati **necessariamente attraverso una *proprietà* che ne caratterizza gli elementi,**
      - Tale proprietà può essere vista come una regola da seguire per generare gli elementi del linguaggio.
      - Il vero problema è trovare la(e) regola(e) generativa(e) (di produzione) di un linguaggio.

Generazione di linguaggi formali

- Esempio
  - Non è possibile “elencare” tutti i teoremi della teoria dei gruppi, perché sono infiniti i teoremi realizzabili combinando quelli noti.
  - **Allo stesso modo, questo accade anche per i linguaggi che impariamo: non è possibile memorizzare tutte le frasi del linguaggio.**
  - **Per descrivere gli elementi appartenenti ad un linguaggio, utilizzeremo una notazione insiemistica. Tale notazione si chiama regola di produzione.**

Generazione di linguaggi formali

- Esempio
  - Sia *L* il linguaggio su $X=\{0\}$ costituito da tutte e sole le stringhe che hanno un numero pari di 0, cioè:

$$L=\{\lambda, 00, 0000, 000000, ...\}$$

Come possiamo esprimere la regola di produzione di L?

Generazione di linguaggi formali

- Esempio
  - Sia *L* il linguaggio su $X = \{0\}$ costituito da tutte e sole le stringhe che hanno un numero pari di 0, cioè:

$$L = \{\lambda, 00, 0000, 000000, \ldots\}$$

La regola di produzione di *L* viene espressa come:

$$L = \{\lambda\} \cup \left\{ w^n \mid w = 00, n = 1,2,\ldots \right\}$$

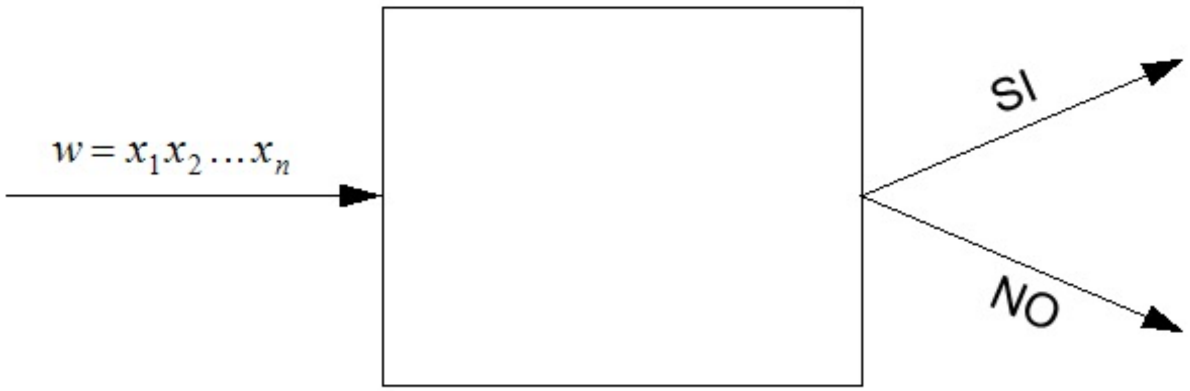
Generazione e riconoscimento di linguaggi formali

- *Punto di vista* Riconoscitivo
  - **Come possiamo *riconoscere* gli elementi di un dato linguaggio *L*?** Questo secondo punto di vista ha come obiettivo la costruzione di "macchine" in grado di decidere/stabilire se una stringa è un elemento di *L* oppure no. Si intende costruire una "macchinetta" cui dare in ingresso una particolare parola e che produce una tra due possibili risposte:

$$s\grave{\imath} \equiv '\in L' \quad \text{e} \quad no \equiv '\notin L'$$

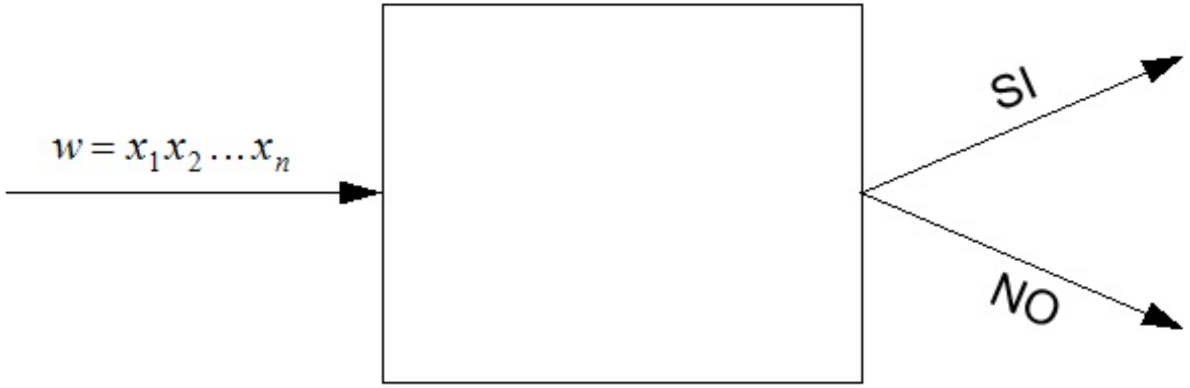
Riconoscimento di linguaggi formali

- **Esempio**
  - **E' possibile eseguire tutti i programmi che scriviamo?**



Riconoscimento di linguaggi formali

- **Esempio**
  - **No! L'esecuzione di un programma errato sintatticamente viene inibita**.
  - Questo è indice dell'esistenza di una "macchinetta" che stabilisce se il programma appartiene o no all'insieme dei programmi sintatticamente ben costruiti.



Generazione di linguaggi formali: esempio

- Sia dato l'alfabeto: $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -\}$
  Voglio generare il linguaggio *L* dei numeri interi relativi. Ovviamente: $L \subseteq X^*$
  - **Assumiamo, per semplicità, che un numero relativo sia costituito da una serie di cifre precedute da + o -**

Generazione di linguaggi formali: esempio

- Sia dato l'alfabeto: $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -\}$
  Voglio generare il linguaggio *L* dei numeri interi relativi. Ovviamente: $L \subseteq X^*$
  - **Assumiamo, per semplicità, che un numero relativo sia costituito da una serie di cifre precedute da + o**

  Più precisamente, $L \subset X^*$ poiché, ad esempio,

$$1 + + - 5 \notin L$$

  Non possiamo elencare gli elementi di *L*. **Cerchiamo dunque una serie di regole mediante le quali è possibile produrre tutti e soli gli elementi di *L*.**

Generazione di linguaggi formali: esempio

- Cominciamo elencando un po' di numeri, per capirne le caratteristiche essenziali
  - +1
  - -5
  - +204
  - +4888
  - -360
  - -95199

Cosa hanno in comune questi numeri? Possiamo intravedere delle regolarità nel meccanismo di costruzione?

Generazione di linguaggi formali: esempio

- Cominciamo elencando un po' di numeri, per capirne le caratteristiche essenziali
  - +1
  - -5
  - +204
  - +4888
  - -360
  - -95199

Cosa hanno in comune questi numeri? Possiamo intravedere delle regolarità nel meccanismo di costruzione?

Ad esempio, ogni numero parte o col più o col meno, seguito da un sequenza di interi

Generazione di linguaggi formali: esempio

- Traduciamo l'intuizione in BNF per descrivere le produzioni:

$$<S> ::= + <I>$$
$$<S> ::= - <I>$$

- **Come si «produce» un numero intero I?**
  - Un intero può avere una lunghezza arbitraria, dunque può essere immaginato come una concatenazione di singole cifre decimali

+1
-5
+204
+4888
-360
-95199

Generazione di linguaggi formali: esempio

- Traduciamo l'intuizione in BNF per descrivere le produzioni:

$$< S > ::= + < I >$$
$$< S > ::= - < I >$$

- **Come si «produce» un numero intero I?**
  - Un intero può avere una lunghezza arbitraria, dunque può essere immaginato come una concatenazione di singole cifre decimali

$$< I > \;::=\; < D >$$
$$< I > \;::=\; < I >< D >$$

+1
-5
+204
+4888
-360
-95199

Generazione di linguaggi formali: esempio

- Traduciamo l'intuizione in BNF per descrivere le produzioni:

$$< S > ::= + < I > \mid - < I >$$

$$< I > ::= < D > \mid < I > < D >$$

+1
-5
+204
+4888
-360
-95199

Generazione di linguaggi formali: esempio

- Traduciamo l'intuizione in BNF per descrivere le produzioni:

$$< S > ::= + < I > \mid - < I >$$

$$< I > ::= < D > \mid < I > < D >$$

$$< D > ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

  - *I* è il simbolo nonterminale (da ora in poi, talvolta abbreviato in *NT*), anche detto *categoria sintattica*, che sta ad indicare (e da cui si genera) la classe dei numeri interi.
  - ***I* è definito ricorsivamente o come una cifra oppure come un intero seguito da una cifra.**
  - Ogni intero relativo è generato da queste regole e niente che non sia un intero relativo può essere generato da queste regole.

Generazione di linguaggi formali: esempio

- Generazione ad albero:
  - proviamo a generare l'intero relativo -375
  - Nella notazione vista per il linguaggio delle parentesi ben formate, tipica per i linguaggi formali, la grammatica diventa:

$$S \to +I \mid -I$$

$$I \to D \mid I\,D$$

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

$$S \overset{*}{\Rightarrow} -375 \quad \Leftrightarrow \quad -375 \in L$$

- Tale albero prende il nome di *albero di derivazione*.

Generazione di linguaggi formali: esempio

S

- Generazione ad albero:
  - proviamo a generare l'intero relativo -375
  - Nella notazione vista per il linguaggio delle parentesi ben formate, tipica per i linguaggi formali, la grammatica diventa:

$$S \to +I \mid -I$$

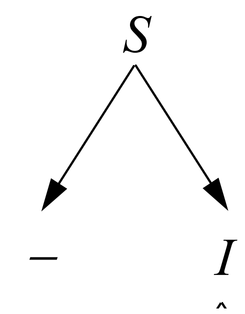
$$I \to D \mid I\,D$$

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

$$S \overset{*}{\Rightarrow} -375 \quad \Leftrightarrow \quad -375 \in L$$

- Tale albero prende il nome di *albero di derivazione*.

Generazione di linguaggi formali: esempio



- Generazione ad albero:
  - proviamo a generare l'intero relativo -375
  - Nella notazione vista per il linguaggio delle parentesi ben formate, tipica per i linguaggi formali, la grammatica diventa:

$$S \to +I \mid -I$$

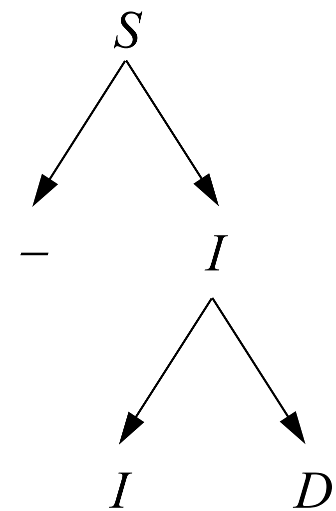
$$I \to D \mid I\,D$$

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

$$S \overset{*}{\Rightarrow} -375 \quad \Leftrightarrow \quad -375 \in L$$

- Tale albero prende il nome di *albero di derivazione*.

Generazione di linguaggi formali: esempio



- Generazione ad albero:
  - proviamo a generare l'intero relativo -375
  - Nella notazione vista per il linguaggio delle parentesi ben formate, tipica per i linguaggi formali, la grammatica diventa:

$$S \rightarrow +I \mid -I$$

$$I \rightarrow D \mid I\,D$$

$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

$$S \overset{*}{\Rightarrow} -375 \quad \Leftrightarrow \quad -375 \in L$$

- Tale albero prende il nome di *albero di derivazione*.

Generazione di linguaggi formali: esempio

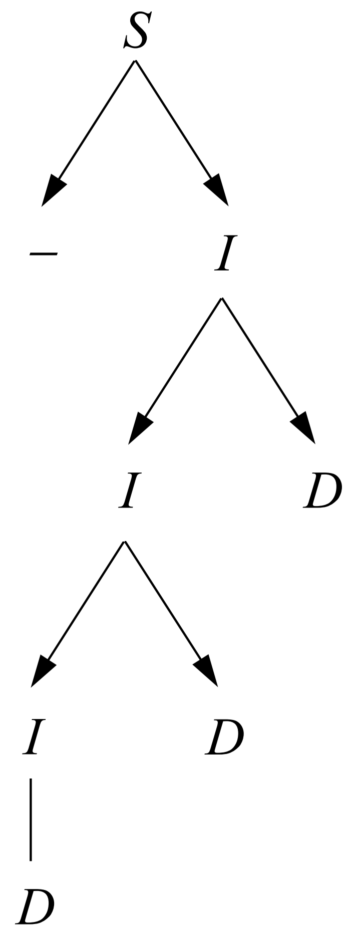
- Generazione ad albero:
  - proviamo a generare l'intero relativo -375
  - Nella notazione vista per il linguaggio delle parentesi ben formate, tipica per i linguaggi formali, la grammatica diventa:

$$S \rightarrow +I \mid -I$$

$$I \rightarrow D \mid I\,D$$

$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

$$S \overset{*}{\Rightarrow} -375 \quad \Leftrightarrow \quad -375 \in L$$



- Tale albero prende il nome di *albero di derivazione*.

Generazione di linguaggi formali: esempio

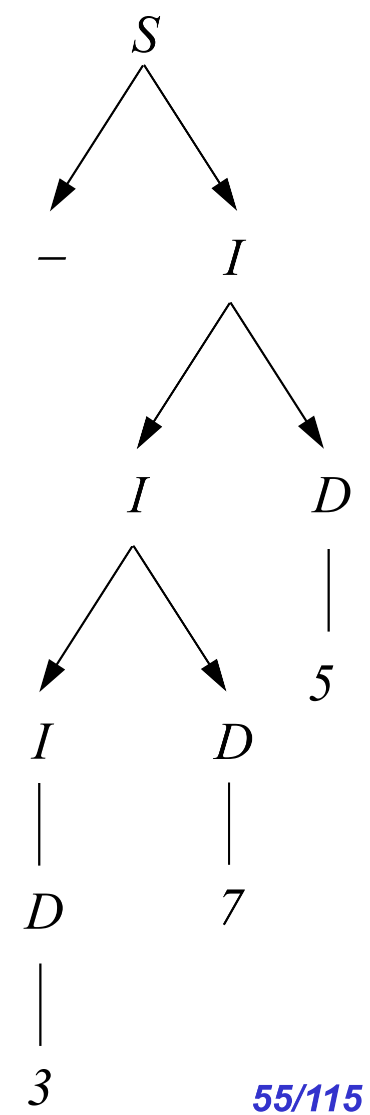
- Generazione ad albero:
  - proviamo a generare l'intero relativo -375
  - Nella notazione vista per il linguaggio delle parentesi ben formate, tipica per i linguaggi formali, la grammatica diventa:

$$S \rightarrow +I \mid -I$$

$$I \rightarrow D \mid I\,D$$

$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

$$S \overset{*}{\Rightarrow} -375 \quad \Leftrightarrow \quad -375 \in L$$



- Tale albero prende il nome di *albero di derivazione*.

Grammatiche generative

- Dagli esempi di linguaggi visti, abbiamo compreso **Per generare un linguaggio sono necessari dei simboli e delle regole.** **Riusciamo a categorizzarli?**

Grammatiche generative

- Dagli esempi di linguaggi visti abbiamo visto che per generare un linguaggio sono necessari:
  - **un simbolo speciale *S***, scelto tra i nonterminali, da cui far partire la generazione delle parole del linguaggio. Tale simbolo è detto *assioma* o *scopo* o *simbolo distintivo* o *simbolo di partenza* o *simbolo iniziale*;
  - **un insieme *P* di produzioni,** espresse in un formalismo quali regole di riscrittura, BNF (*a* ::= *b*), carte sintattiche, …

Grammatiche generative

- Dagli esempi di linguaggi visti abbiamo visto che per generare un linguaggio sono necessari:
  - **un simbolo speciale *S***, scelto tra i nonterminali, da cui far partire la generazione delle parole del linguaggio. Tale simbolo è detto *assioma* o *scopo* o *simbolo distintivo* o *simbolo di partenza* o *simbolo iniziale*;
  - **un insieme *P* di produzioni,** espresse in un formalismo quali regole di riscrittura, BNF (*a* ::= *b*), carte sintattiche, …
  - **un insieme *X* di simboli primitivi** con cui si formano le parole del linguaggio, detto *alfabeto dei simboli terminali* o *alfabeto terminale*;
  - **un insieme *V* di simboli ausiliari o variabili** con cui si identificano le categorie sintattiche del linguaggio, detto *alfabeto dei simboli nonterminali (ausiliari)* o *alfabeto nonterminale* o *alfabeto delle variabili*;

Definizione di Grammatica generativa o a struttura di frase

- Una *grammatica generativa* o *a struttura di frase* G è una quadrupla

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

ove:

  - X è l'*alfabeto terminale* per la grammatica;
  - V è l'*alfabeto nonterminale* o delle variabili per la grammatica;
  - S è il *simbolo di partenza* per la grammatica;
  - P è l'insieme delle *produzioni* della grammatica

  ed inoltre valgono le seguenti condizioni:

$$X \cap V = \varnothing \quad \text{e} \quad S \in V$$

Definizione di Produzione

- Una *produzione* è una coppia (v,w) nella forma **v→w, formalizziamone la definizione.**

- Riprendiamo delle produzioni già viste, per capirne meglio la formalizzazione
  - Ovviamente **v** è la parte sinistra, **w** la parte destra.

$$S \to +I \mid -I$$
$$I \to D \mid I\,D$$
$$D \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

Definizione di Produzione

- Una *produzione* è una coppia (v,w) nella forma **v→w,** ove $v \in (X \cup V)^+$
  e v contiene un *NT* $\Leftrightarrow v \in (X \cup V)^* V (X \cup V)^*$

Importante: $v \in (X \cup V)^+$ *significa che la parte sinistra può avere lunghezza anche maggiore di 1*

$$S \rightarrow +I \mid -I$$
$$I \rightarrow D \mid I\,D$$
$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

Definizione di Produzione

- Una *produzione* è una coppia (*v*,*w*) nella forma **v→w,** ove $v \in (X \cup V)^+$

  e *v* contiene un *NT* $\Leftrightarrow v \in (X \cup V)^* V (X \cup V)^*$

  $w \in (X \cup V)^*$ (*w può essere anche* λ*).*

$$S \rightarrow +I \mid -I$$
$$I \rightarrow D \mid I\,D$$
$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

Definizione di Produzione

- Una *produzione* è una coppia (*v*,*w*) nella forma **v→w,** ove $v \in (X \cup V)^+$
 e *v* contiene un *NT* $\Leftrightarrow v \in (X \cup V)^* V (X \cup V)^*$

 $w \in (X \cup V)^*$ *(w può essere anche λ).*

 Un elemento (v,w) di P viene comunemente scritto nella forma:

$$v \to w$$

 Una produzione deve, in qualche modo, riscrivere un NT.

Definizione di Produzione

- La notazione $\alpha \rightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_k$ è impiegata come abbreviazione della seguente:

$$
\begin{gathered}
\alpha \rightarrow \beta_1 \\
\alpha \rightarrow \beta_2 \\
\ldots \\
\alpha \rightarrow \beta_k
\end{gathered}
$$

- Per convenzione, **gli elementi di *X* sono rappresentati di solito con lettere minuscole** (con o senza pedici e di solito sono le prime lettere dell'alfabeto) o cifre ed operatori (connettivi)
- **Mentre gli elementi di *V* sono rappresentati con lettere maiuscole** (con o senza pedici) o con stringhe delimitate dalle parentesi angolari “<“ e “>“.

Esempi di grammatiche

- La grammatica per il linguaggio delle parentesi ben formate

- La grammatica per il linguaggio dei numeri interi relativi

Esempi di grammatiche

- La grammatica per il linguaggio delle parentesi ben formate

$$G_1 = \left(\{\, (\,,\,)\,\},\quad \{S\},\quad S,\quad \{S \to (\ ),\ S \to (S),\ S \to SS\} \right)$$

- La grammatica per il linguaggio dei numeri interi relativi

Esempi di grammatiche

- La grammatica per il linguaggio delle parentesi ben formate

$$G_1 = \left(\{\, (\, ,\,)\, \}, \quad \{S\}, \quad S, \quad \{S \rightarrow (\ \ ),\ S \rightarrow (S),\ S \rightarrow SS\} \right)$$

- La grammatica per il linguaggio dei numeri interi relativi

$$G_2 = \Big(\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, \text{-}\}, \quad \{S, I, D\}, \quad S,$$
$$\{S \rightarrow +I,\ S \rightarrow -I,\ I \rightarrow D,\ I \rightarrow ID,\ D \rightarrow 0,\ D \rightarrow 1,\ \ldots,\ D \rightarrow 9\} \Big)$$

Definizione di derivazione o produzione diretta

- Intuitivamente, una derivazione è il **risultato dell’applicazione di una regola di produzione**
- **Formalizziamolo**

Definizione di derivazione o produzione diretta

- Sia $G = (X, V, S, P)$ una grammatica e siano ***y* e *z*** due stringhe finite di simboli in $X \cup V$ (stringhe di terminali e nonterminali) tali che:

$$y \in (X \cup V)^+, \; z \in (X \cup V)^*,$$

Le stringhe **y** e **z** possono essere descritte come $y = \gamma\alpha\delta$ e $z = \gamma\beta\delta$ tale che

$$\beta, \gamma, \delta \in (X \cup V)^* \qquad \alpha \in (X \cup V)^+ \text{ e } \alpha \text{ contiene un NT}$$

Definizione di derivazione o produzione diretta

- Sia $G = (X, V, S, P)$ una grammatica e siano y e z due stringhe finite di simboli in $X \cup V$ (stringhe di terminali e nonterminali) tali che:

  $y = \gamma\alpha\delta$ e $z = \gamma\beta\delta$, ove $y \in (X \cup V)^+$, $z \in (X \cup V)^*$,
  $\beta, \gamma, \delta \in (X \cup V)^*$ $\quad \alpha \in (X \cup V)^+$ e α contiene un NT

  1) Scriviamo

  $$y \Rightarrow z$$

  e diciamo che y *produce direttamente* z o che z è *derivata direttamente* *da* y se:

  $$\alpha \rightarrow \beta \in P$$

  ossia se esiste in G una produzione $\alpha \rightarrow \beta$

Definizione di derivazione o produzione diretta

- Ad esempio, data la **grammatica delle parentesi ben formate**
- Se
  - **`y = (S)`**
  - **`z = ( ( ) )`**
  - Possiamo dire che **y** produce direttamente **z** ?
    - Esiste una regola che **«riscriva»** Y in Z?

$$G_1 = \left(\{\, (\, ,\,)\, \},\quad \{S\},\quad S,\quad \{S \rightarrow (\ \ ),\ S \rightarrow (S),\ S \rightarrow SS\}\right)$$

Definizione di derivazione o produzione diretta

- Ad esempio, se **y = (S)** e **z = (( )),** possiamo dire che **y** produce direttamente **z** ?
  - Si, perché esiste una regola che permette di trasformare y in z.
  - Tale regola è **S → ( )**
  - **Formalizzando**
    - γ =
    - δ =
    - α =
    - β =

$$y = \gamma\alpha\delta \quad z = \gamma\beta\delta$$

$$y \in (X \cup V)^+, \; z \in (X \cup V)^*,$$

Definizione di derivazione o produzione diretta

- Ad esempio, se **y = (S)** e **z = (( ))**, possiamo dire che **y** produce direttamente **z** ?
  - Si, perché esiste una regola che permette di trasformare y in z.
  - Tale regola è **S → ( )**
  - **Formalizzando**
    - **γ = (**
    - **δ =)**
    - **α = S**
    - **β = ( )**

$$y = \gamma\alpha\delta \quad z = \gamma\beta\delta$$

$$y \in (X \cup V)^+, \; z \in (X \cup V)^*,$$

Definizione di derivazione o produzione diretta

- Ad esempio, se **y = (S)** e **z = (( )),** possiamo dire che **y** produce direttamente **z** ?
  - Si, perché esiste una regola che permette di trasformare y in z.
  - Tale regola è **S → ( )**
- Invece, se **y = (S)** e **z = ((( ))), si può parlare di produzione diretta?**

Definizione di derivazione o produzione diretta

- Ad esempio, se **y = (S)** e **z = (( )),** possiamo dire che **y** produce direttamente **z** ?
  - Si, perché esiste una regola che permette di trasformare y in z.
  - Tale regola è **S → ( )**
- NO, se **y = (S)** e **z = ((( ))), non possiamo parlare di produzione diretta**, perché tale regola non esiste.

Definizione di derivazione o produzione diretta

2) Scriviamo

$$y \overset{*}{\Rightarrow} z$$

e diciamo che *y* *produce* *z* o che *z* è *derivabile da* *y* se *y* = *z* o esiste una sequenza di stringhe $w_1, w_2, \ldots, w_n$, con $w_1, w_2, \ldots, w_{n-1} \in (X \cup V)^+, \quad w_n \in (X \cup V)^*$
w_1 = *y* e w_n = *z* tali che $\forall i,\ i = 1,2,\ldots,n-1 : w_i \underset{G}{\Rightarrow} w_{i+1}$
(w_i produce direttamente w_{i+1}), cioè:

$$y \overset{*}{\Rightarrow} z \quad \Leftrightarrow \quad \begin{cases} y = z \\ \quad \text{oppure} \\ w_1 = y \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \ldots \Rightarrow w_{n-1} \Rightarrow w_n = z \end{cases}$$

Definizione di derivazione o produzione diretta

2) Scriviamo

$$y \overset{*}{\Rightarrow} z$$

e diciamo che *y* *produce* *z* o che *z* è *derivabile* *da* *y* se *y* = *z* o esiste una sequenza di stringhe $w_1, w_2, \ldots, w_n$, con $w_1, w_2, \ldots, w_{n-1} \in (X \cup V)^+, \quad w_n \in (X \cup V)^*$ w_1 = *y* e w_n = *z* tali che $\forall i,\ i = 1,2,\ldots,n-1 : w_i \underset{G}{\Rightarrow} w_{i+1}$ (w_i produce direttamente w_{i+1}), cioè:

Tornando all'esempio precedente, possiamo dire che da Y deriva Z perché esiste una sequenza di applicazione di regole **che ci permetta di costruire Z a partire da Y.**

Osservazione

- **La nozione di derivazione diretta stabilisce una relazione binaria in $(X \cup V)^*$.**
  Date due stringhe *y* e *z*, il simbolo $\Rightarrow$ può esserci o meno; dipende dall'esistenza di una produzione.
- **Se la produzione diretta non esiste, si può introdurre il concetto di composizione di relazioni.**

Osservazione

- **La nozione di derivazione diretta stabilisce una relazione binaria in $(X \cup V)^*$.**
  Date due stringhe *y* e *z*, il simbolo $\Rightarrow$ può esserci o meno; dipende dall'esistenza di una produzione.
  Allora possiamo anche definire una composizione di relazioni:

  $$y \overset{2}{\Rightarrow} z \quad \overset{def}{\Leftrightarrow} \quad \exists\, w : y \Rightarrow w \;\text{ e }\; w \Rightarrow z$$

  dove 2 è il numero di trascrizioni necessarie per passare da *y* a *z* (ossia, la *lunghezza della derivazione*).

Osservazione

- Da ciò si ha: $\stackrel{*}{\Rightarrow} \; = \; I \cup \Rightarrow \cup \stackrel{2}{\Rightarrow} \cup \stackrel{3}{\Rightarrow} \cup \ldots$

  ove *I* è la relazione identica e $\stackrel{n}{\Rightarrow}$ indica la composizione della relazione $\Rightarrow$ *n* volte con se stessa.

  $\stackrel{*}{\Rightarrow}$ è la chiusura riflessiva e transitiva della relazione di derivazione diretta;

  $\stackrel{+}{\Rightarrow}$ è la chiusura transitiva della stessa relazione.

Esempio

- Proviamo a costruire la **chiusura transitiva** $\overset{+}{\Longrightarrow}$ del linguaggio delle parentesi ben formate

- $S \overset{1}{\Longrightarrow} ?$

$$G_1 = \left(\{\,(\,,\,)\,\},\quad \{S\},\quad S,\quad \{S \rightarrow (\ ),\ S \rightarrow (S),\ S \rightarrow SS\}\right)$$

Esempio

- Proviamo a costruire la **chiusura transitiva** $\overset{+}{\Longrightarrow}$ del linguaggio delle parentesi ben formate

- $S \overset{1}{\Longrightarrow}$ ( ) | (S) | SS

$$G_1 = \left(\{ (\,,\,) \},\quad \{S\},\quad S,\quad \{ S \rightarrow (\ ),\ S \rightarrow (S),\ S \rightarrow SS \} \right)$$

Esempio

- Proviamo a costruire la **chiusura transitiva** $\overset{+}{\Longrightarrow}$ del linguaggio delle parentesi ben formate

- $S \overset{1}{\Longrightarrow}$ ( ) | (S) | SS

- $S \overset{2}{\Longrightarrow}$ ( ) | (( )) | ((S)) | (SS) | ( ) S | S ( ) | (S) S | S (S) ..
- .
- .
- continuate

$$G_1 = \big(\{\,(\,,\,)\,\},\quad \{S\},\quad S,\quad \{S \to (\ \ ),\ S \to (S),\ S \to SS\}\big)$$

Definizione di linguaggio generato da una grammatica

- Sia G = (X, V, S, P) una grammatica. Il *linguaggio generato da G*, denotato con $L(G)$, è l'insieme delle stringhe di terminali derivabili dal simbolo di partenza S.

$$L(G) = \left\{ w \in X^* \;\middle|\; S \overset{*}{\underset{G}{\Rightarrow}} w \right\}$$

- Sono, dunque, stringhe di $L(G)$ le stringhe che:
  - consistono di soli terminali;
  - possono essere derivate da S in G.

Definizione di forma di frase

- Sia $G = (X, V, S, P)$ una grammatica. Una stringa w, $w \in (X \cup V)^*$, è una *forma di frase* di G se:

$$S \overset{*}{\underset{G}{\Rightarrow}} w$$

- Quale è la differenza?

Definizione di forma di frase

- Sia $G = (X, V, S, P)$ una grammatica. Una stringa w, $w \in (X \cup V)^*$, è una *forma di frase* di G se:

$$S \overset{*}{\underset{G}{\Rightarrow}} w$$

- Quale è la differenza?
- Le forme di frase **non sono necessariamente composte da soli terminali.**

Definizione di forma di frase

- Sia $G = (X, V, S, P)$ una grammatica. Una stringa w, $w \in (X \cup V)^*$, è una *forma di frase* di G se:

$$S \overset{*}{\underset{G}{\Rightarrow}} w$$

- Alle forme di frase si applicano le stesse definizioni (es.: potenza) e gli stessi operatori (es.: concatenazione) dati per le stringhe.
- Proposizione:
  - Data una grammatica $G = (X, V, S, P)$
    - $L(G)$ è l'insieme delle forme di frase terminali (o *frasi*) di G.
    - Nel caso delle parentesi ben formate, **L(G) = (), ()(), ((())), etc.**

Definizione di grammatiche equivalenti

- Due grammatiche *G* e *G'* si dicono *equivalenti* se generano lo stesso linguaggio, ossia se

$$L(G)=L(G')$$

Esempio

- Sia G = (X, V, S, P), ove

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

  Determiniamo $L(G)$.

Esempio

- Sia G = (X, V, S, P), ove

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

Determiniamo $L(G)$.

$ab \in L(G)$ poiché $S \underset{(2)}{\Rightarrow} ab$

Se numeriamo le produzioni, possiamo indicare la produzione usata immediatamente al di sotto del simbolo $\Rightarrow$.

$\underset{(n)}{\Longrightarrow}$ $\equiv$ ho applicato la produzione n

$y \overset{k}{\Rightarrow} z$ $\equiv$ y produce z in k passi, dove k=lunghezza della derivazione

Esempio

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

- $a^2b^2 \in L(G)$?
- $a^3b^3 \in L(G)$?

Esempio

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

- *$a^2b^2 \in L(G)$* *si,* poiché $S \underset{(1)}{\Rightarrow} aSb \underset{(2)}{\Rightarrow} a^2b^2$
- *$a^3b^3 \in L(G)$* si, poiché $S \overset{3}{\Rightarrow} a^3b^3$

……

Esempio

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

- $a^2b^2 \in L(G)$ *si,* poiché $S \underset{(1)}{\Rightarrow} aSb \underset{(2)}{\Rightarrow} a^2b^2$
- $a^3b^3 \in L(G)$ si, poiché $S \overset{3}{\Rightarrow} a^3b^3$

……

$$\{a^n b^n \mid n > 0\} \subseteq L(G)$$

Esempio

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

- $a^2b^2 \in L(G)$ *si,* poiché $S \underset{(1)}{\Rightarrow} aSb \underset{(2)}{\Rightarrow} a^2b^2$
- $a^3b^3 \in L(G)$ si, poiché $S \overset{3}{\Rightarrow} a^3b^3$

……

$$\{a^n b^n \mid n > 0\} \subseteq L(G)$$

- Inoltre, qualsiasi derivazione da *S* in *G* produce frasi del tipo $a^n b^n$.
  - Dunque $L(G) \subseteq \{a^n b^n \mid n > 0\}$

Esempio

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

- $a^2b^2 \in L(G)$ *si*, poiché $S \underset{(1)}{\Rightarrow} aSb \underset{(2)}{\Rightarrow} a^2b^2$
- $a^3b^3 \in L(G)$ si, poiché $S \overset{3}{\Rightarrow} a^3b^3$

$$\ldots\ldots$$

$$\{a^n b^n \mid n > 0\} \subseteq L(G)$$

- Inoltre, qualsiasi derivazione da *S* in *G* produce frasi del tipo $a^n b^n$.
  - Dunque $L(G) \subseteq \{a^n b^n \mid n > 0\}$ e quindi

$$L(G) = \{a^n b^n \mid n > 0\}$$

Notazione

- Per rendere più concisa la descrizione di una grammatica, **spesso ci limiteremo ad elencarne le produzioni, quando sia chiaro quale sia il simbolo di partenza e quali siano i terminali ed i nonterminali**.
- Inoltre, le produzioni con la stessa parte sinistra **vengono accorpate attraverso l'uso del simbolo "|" (preso a prestito dalla BNF).**
- Infine, ometteremo l'indicazione della grammatica dalla simbologia di derivazione e derivazione diretta quando sia chiaro dal contesto a quale grammatica si fa riferimento.

Esempio

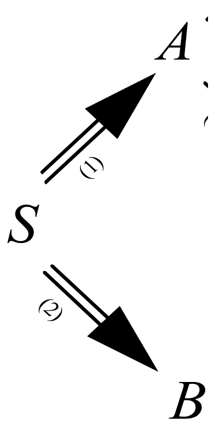
- Sia data la seguente grammatica:

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

Determinare *L(G).*

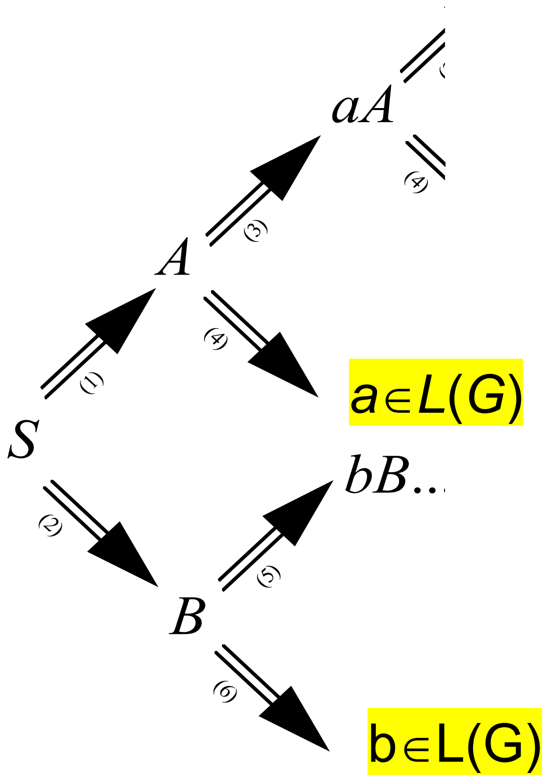
Esempio

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



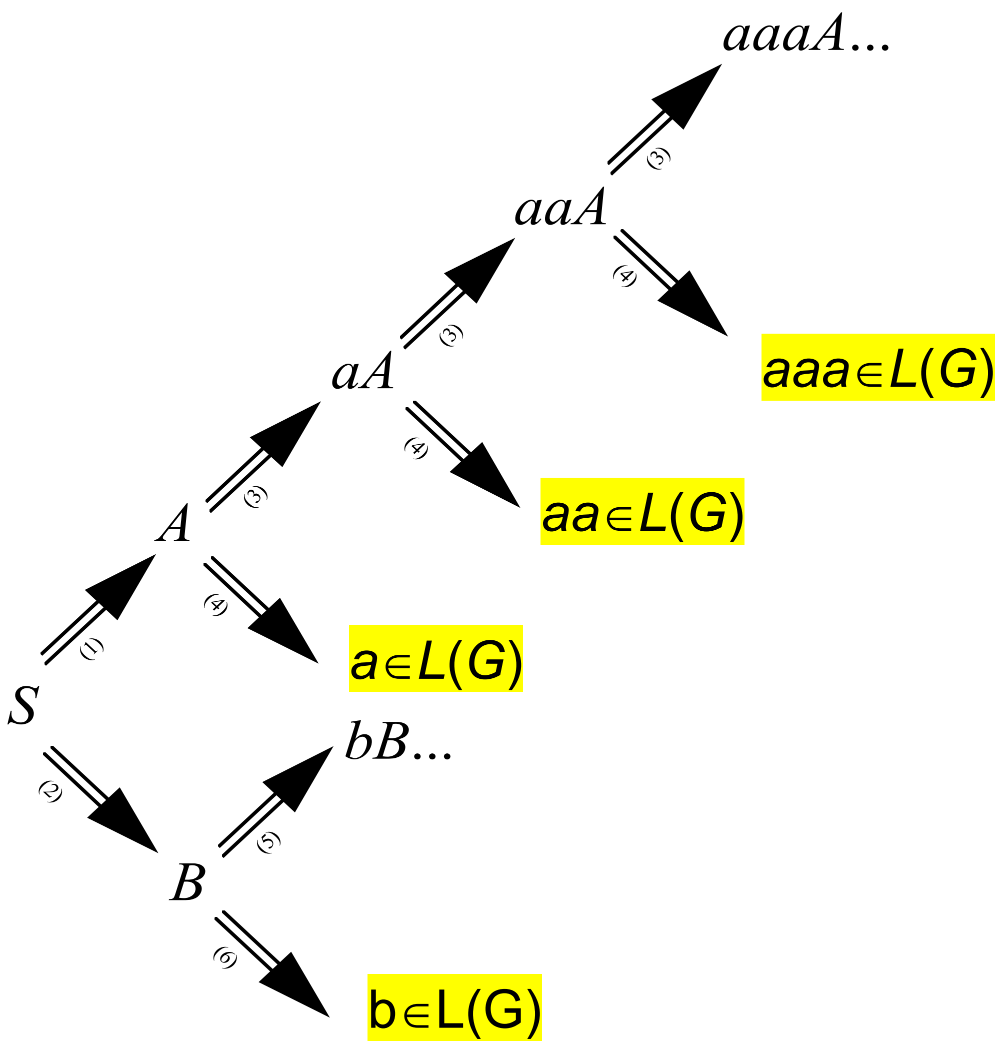
Esempio

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

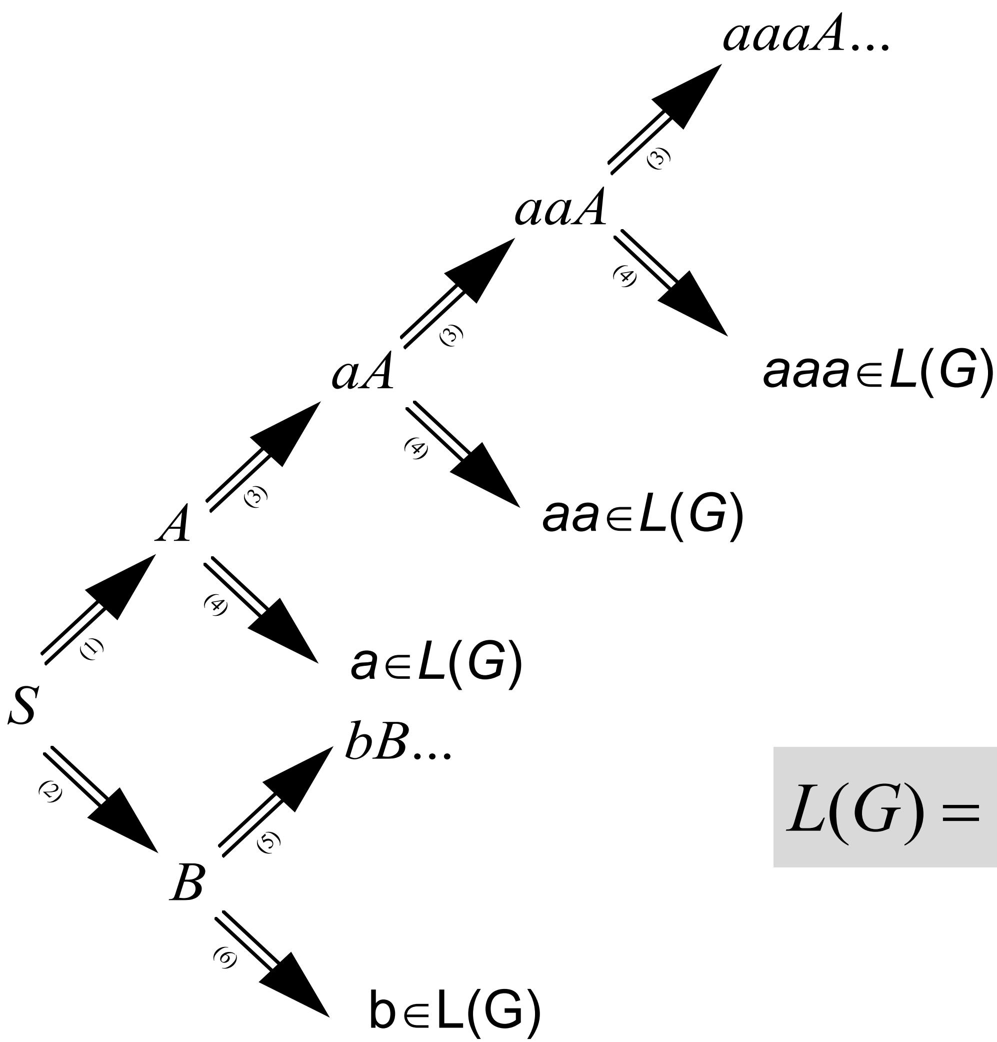


Esempio

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



Esempio



$$L(G) = \{a^n | n > 0\} \cup \{b^n | n > 0\}$$

Esempio

- Sia data la seguente grammatica:

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

Nota

Non sappiamo se applicare $S \overset{(1)}{\to} A$ oppure $S \overset{(2)}{\to} B$ inizialmente. **I meccanismi di costruzione di un linguaggio sono generalmente non deterministici**, poiché può non essere univoca la sostituzione da operare ad una forma di frase se uno stesso NT si trova a sinistra di 2 o più produzioni, come illustrato nella figura precedente.

Osservazione

- **Dunque, una grammatica è uno *strumento generativo* di un linguaggio** perché, data una qualsiasi parola di quel linguaggio, possiamo risalire mediante le produzioni al simbolo di partenza della grammatica.
  - **Punto di vista generativo!**

Osservazione

- **Dunque, una grammatica è uno *strumento generativo* di un linguaggio** perché, data una qualsiasi parola di quel linguaggio, possiamo risalire mediante le produzioni al simbolo di partenza della grammatica.
  - **Punto di vista generativo!**
- Viceversa, dato il simbolo di partenza di una grammatica, **seguendo uno qualsiasi dei cammini dell'albero di derivazione, si produce una parola "valida" del linguaggio**.
  - **Punto di vista riconoscitivo!**

Osservazione

- In generale, dato un linguaggio L ed una grammatica G, **non** esiste un algoritmo in grado di dimostrare che la grammatica genera il linguaggio, ossia che $L = L(G)$.
  Più specificamente, **non** esiste un algoritmo che stabilisce se una data stringa è generata o no dalla grammatica presa in considerazione.

Osservazione

- In generale, dato un linguaggio L ed una grammatica G, **non** esiste un algoritmo in grado di dimostrare che la grammatica genera il linguaggio, ossia che $L = L(G)$.
  Più specificamente, **non** esiste un algoritmo che stabilisce se una data stringa è generata o no dalla grammatica presa in considerazione.
- Tutto ciò si riassume nella seguente proposizione:
  - Il problema di dimostrare la correttezza di una grammatica **non è risolubile algoritmicamente, in generale**.

Osservazione

- In molti casi importanti, però, **è possibile dimostrare per induzione** che una particolare grammatica genera proprio un particolare linguaggio (**L=L(G)**).
- Queste dimostrazioni ci consentono di stabilire se, data una grammatica *G* ed un linguaggio *L*, risulta:
  - $w \in L(G) \Rightarrow w \in L$ cioè $L(G) \subseteq L$
  - $w \in L \Rightarrow w \in L(G)$ cioè $L \subseteq L(G)$

Osservazione

- In molti casi importanti, però, **è possibile dimostrare per induzione che** una particolare grammatica genera proprio un particolare linguaggio (**L=L(G)**).
- Queste dimostrazioni ci consentono di stabilire se, data una grammatica *G* ed un linguaggio *L*, risulta:
  - $w \in L(G) \Rightarrow w \in L$ cioè $L(G) \subseteq L$
  - $w \in L \Rightarrow w \in L(G)$ cioè $L \subseteq L(G)$

La grammatica *G* genera solo stringhe appartenenti al linguaggio *L*.

Osservazione

- **In molti casi importanti, però, è possibile dimostrare per induzione** che una particolare grammatica genera proprio un particolare linguaggio. (**L=L(G)**).
- Queste dimostrazioni ci consentono di stabilire se, data una grammatica *G* ed un linguaggio *L*, risulta:
  - $w \in L(G) \Rightarrow w \in L$ cioè $L(G) \subseteq L$
  - $w \in L \Rightarrow w \in L(G)$ cioè $L \subseteq L(G)$

Il linguaggio *L* comprende solo parole generabili dalla grammatica *G*.

Principio di induzione

- Sia n_0 un intero e sia ***P=P(n)*** **un enunciato** che ha senso per ogni intero maggiore o uguale ad n_0 .
- Se:
  - $P(n_0)$ è vero **(passo base)**
  - Per ogni $n>n_0$, $P(n-1)$ vero implica $P(n)$ vero

allora $P(n)$ è vero per tutti gli n maggiori o uguali ad n_0

(passo induttivo)

Domande?

Linguaggi di Programmazione
docente: Cataldo Musto
cataldo.musto@uniba.it

Esercizi

- Determinare una grammatica che genera il seguente linguaggio:

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

  e dimostrare questo risultato.

- Che tipo di grammatica genera *L* ?

Soluzione esercizio

Esercizi

- Determinare una grammatica che genera il seguente linguaggio:

$$L = \{a^n b^{2n} \mid n > 0\}$$

e dimostrare questo risultato.

- Di che tipo è la grammatica che genera *L* ?

Soluzione esercizio

Esercizi

- Sia data la seguente grammatica:

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

$$X = \{0, 1\} \qquad V = \{S, A, B\}$$

$$P = \left\{ S \to \overset{(1)}{0B} \mid \overset{(2)}{1A},\ A \to \overset{(3)}{0} \mid \overset{(4)}{0S} \mid \overset{(5)}{1AA},\ B \to \overset{(6)}{1} \mid \overset{(7)}{1S} \mid \overset{(8)}{0BB} \right\}$$

- Determinare il linguaggio generato da *G*.

[Soluzione esercizio]

Esercizi

- Dimostrare per induzione che il linguaggio *L* generato dalla seguente grammatica è vuoto:

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

$$X = \{a, b, c\} \qquad V = \{S, A, B\}$$

$$P = \left\{ S \rightarrow \overset{(1)}{aBS} \mid \overset{(2)}{bA},\ \ aB \rightarrow \overset{(3)}{Ac} \mid \overset{(4)}{a},\ \ bA \rightarrow \overset{(5)}{S} \mid \overset{(6)}{Ba} \right\}$$

Soluzione esercizio