

## forme sintattiche di Backus

La definizione di linguaggi di preprogrammazione consentono di considerare un linguaggio di programmazione come l'insieme di tutte le strutture derivabili dalle grammatiche.

Per le definizioni sintattiche dei linguaggi di preprogrammazione vengono adottate grammatiche non contestuali, di solito rappresentate mediante una notazione specifica, dette

### FORMA NORMALE DI BACKUS (BNF)

La BNF è una notazione per grammatiche contest free dove "più espansive" mediante delle notazioni che analisano e definire

Queste notazioni sono "appositamente" riposte alle estensioni regolari

1) Simboli non terminali sono sempre costituiti da stringhe che demarcano delle categorie sintattiche racchuse tra parentesi tonde <...>

<espressione> chiamati identificatori

2) Il segno di associazione ( $\rightarrow$ ) si può sostituire dal simbolo ( $::=$ ) pur non confonderlo con i simboli:  $\rightarrow, =, ::$  che sono simboli terminali usati in vari linguaggi di preprogrammazione

3) Le parentesi graffe  $\{ \dots \}$  vengono per indicare l'iterazione illimitata ( $0, 1, 2, \dots, n$ , volte). Analogamente molchiamo con  $\{ \dots \}^n$  l'iterazione per un numero di

voltre almeno perci' od in

usando quelle precedenti  
lo siamo scritto  
di volta

PER ESEMPIO:

$A ::= B A | e$  può essere scritto  $A ::= \{B\}^*$

$B A \rightarrow BB A \rightarrow \dots \rightarrow B^n A \rightarrow B^n e$

oppure

$A ::= x \{xy \mid xyx \mid xyy \mid xyyy \mid xyyyy\}$  può essere scritto

come  $A ::= x \{y\}^5$

a) Le parentesi quadre vengono utilizzate per indicare l'opzione (possibile ensemble di posse di une stringhe)

$A ::= x y \mid y$  può essere scritto  $A ::= [x]y$  ma non è possibile

5) Le parentesi tonde (...) vengono utilizzate per evitare la sovrapposizione, solo si dice le une in comune di une rottamazione

Per esempio

$A ::= x u \mid x v \mid x y$  può essere scritto come

$A ::= x(u \mid v \mid y)$

# FORMA NORMALE DI CHOMSKY

## Definizione:

Sia  $G = \langle V_T, V_N, P, S \rangle$  context free. La grammatica  $G$  si dice in forme normale di Chomsky oppure Chomsky Normal Form (CNF)

Se tutte le regole di produzione di  $G$  sono delle forme:

$$A \rightarrow BC \quad \text{con } A, B, C \in V_N$$

$$\text{oppure } A \rightarrow a \quad \text{con } a \in V_T$$

## Proposizione

Dato una qualsiasi grammatica context free di tipo 2 ne esiste una equivalente in CNF del tipo

$$S \rightarrow a S b \mid b S$$

e questo si dimostra tramite una dimostrazione costruttiva

## Esempio

$A \rightarrow a x b y$  grammatica di tipo 2 (in questo momento non è in forme normale)

Allora eseguo delle trasformazioni

$$\left\{ \begin{array}{l} A \rightarrow B x C y \\ B \rightarrow a \quad 1^{\circ} \text{ trasformazione} \\ C \rightarrow b \end{array} \right.$$

$$\left\{ \begin{array}{l} A \rightarrow B z_1 \\ z_1 \rightarrow x z_2 \\ z_2 \rightarrow a y \quad 2^{\circ} \text{ trasformazione} \end{array} \right.$$

Le regole di produzioni aumentano ma il risultato è uguale e più breve

Significa che abbiamo definito una grammatica equivalente a quelle di tipo 2 definite in forme normale

## FORMA NORMALE DI GREIBACH

2 definito:  $A \rightarrow \alpha \in V_N \quad \alpha \in V_T$

$\beta \in V_N^*$ insieme delle stringhe non  
Terminali

Le produzioni sono del tipo

$$S \xrightarrow[G]{*} x \beta \quad x \in V_T^* \quad \beta \in V_N^*$$

Nell'ultima è possibile un punto equivalente per escludere le  
stringhe  $x$  e scindere sul resto  $\beta$ . Anche in questo caso come per  
le grammatiche di tipo 2 non esiste una forma normale  
di Greibach. Si noti che in forma normale NON ESISTE  
descrizione simbolica e quindi possiamo scrivere che

$$A \xrightarrow[G]{*} AB$$

Esempio di Backus

$$V_T = \{ e, b, \dots, 2, A, B, \dots, 2, 0, \dots, 9 \}$$

$$\langle \text{identificatore} \rangle ::= \langle \text{alfabetico} \rangle \{ \langle \text{alfanumerico} \rangle \}^{m-1}$$

$$\langle \text{alfabetico} \rangle ::= e | b | \dots | 2 | A | B | \dots | 2$$

$$\langle \text{alfanumerico} \rangle ::= \langle \text{alfabetico} \rangle | \langle \text{cifre} \rangle$$

$$\langle \text{cifre} \rangle ::= 0 | 1 | \dots | 9$$

↑

Capitolo 2 Linguaggi Formale esempio 2.24  
esempio 2.25 BNF per il Pascal

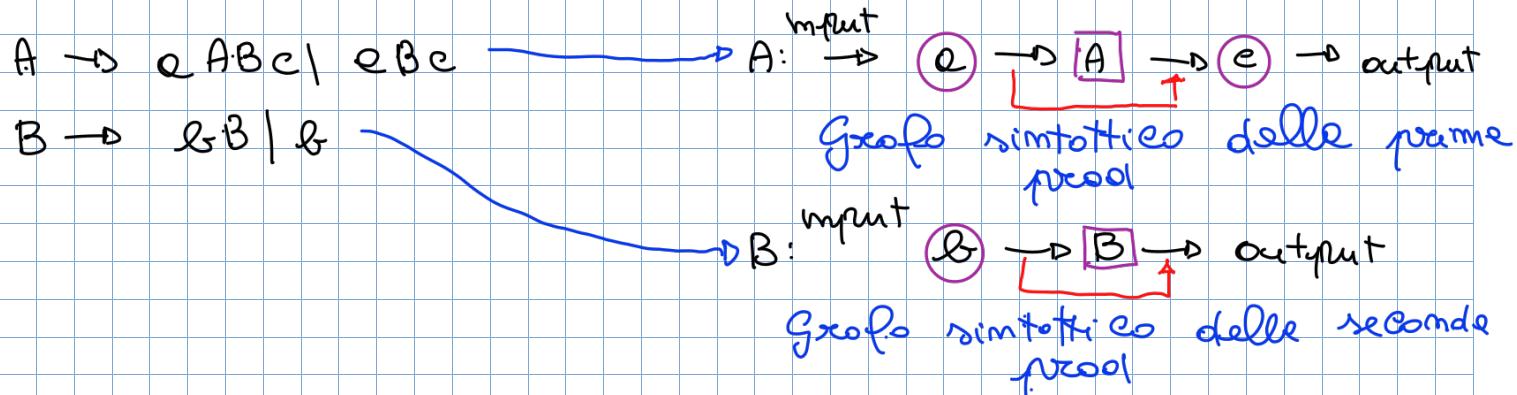
Un'altra notazione usata per descrivere i linguaggi di  
programmazione è costituita dai disegni sintattici

↓

li possiamo rappresentare  
come profili orientati

Con un imprenditore, un'ente e i cui modi sono collegati tra loro, tramite giochi scientifici ed etici. Con uno o più concorrenti terminali e non terminali. I concorrenti terminali vengono nelle forme: cerchio e i concorrenti non terminali sono inclusi all'interno di un quadrato.

Definiamo il diagramma sintattico delle regole grammaticali



Esercizi sulle grammatiche

1) Verifica che le grammatiche si regolese

S:

$$S \rightarrow \epsilon S | \epsilon A$$

$$A \rightarrow \epsilon A | E$$

Ogni produzione

$$S \rightarrow \epsilon S \quad \checkmark$$

$$S \rightarrow \epsilon A \quad \checkmark$$

$$A \rightarrow \epsilon A \quad \checkmark$$

Per essere regolare

$$A \rightarrow \epsilon B$$

} Verifico queste condizioni

$A \rightarrow E$  è un non terminale di fine catena

1) Queste grammatiche si regolese destre

2) Generazione del linguaggio da una grammatica

$$G: S \rightarrow eS \mid bS \mid E$$

Quale linguaggio genera  $G$ ?

$$S \rightarrow eS \rightarrow ebeS \rightarrow ebbeS$$

$$\downarrow \rightarrow ebbeS$$

Ogni produzione come  $e$  o  
una  $b$  e richiede  $S$   
oppure termine con  $E$

2) Generare tutte le stringhe (anche le vuote) su  $\Sigma = \{e, b\}$   
e quindi il linguaggio generato da  $G$  è  $\{e, b\}^*$

3) Classificazione delle grammatiche

$$G = \langle \{e, b\}, \{S\}, P, S \rangle$$

$$S \rightarrow eSb$$

portando da un non terminale a

$$S \rightarrow E$$

produce come terminale o non terminale

3)  $E$  di tipo 1

Coprire linee  
destre / sinistre

4) Generare il linguaggio dell'esercizio 3

$$S \rightarrow E$$

$$\rightarrow eSb \rightarrow eeSbb \rightarrow eeeSbbb$$

$$L(G) = \{e^m b^m \mid m \geq 0\}$$

5) Classificazione

$$S \rightarrow eA$$

$$A \rightarrow eB$$

$$B \rightarrow b$$

Sono rimeori destri della produzione del tipo:

$$X \rightarrow w_1 \text{ opp } X \rightarrow w$$

5') È di tipo 3

$$S \rightarrow eA \rightarrow eeB \rightarrow eeb$$

$$L(S) = \{e^i b\}$$

6) Classificazione

$$eS \rightarrow Sb$$

$$S \rightarrow eb$$

$$eS \rightarrow eb$$

$$\rightarrow eeb$$

6') Tipo 1

7)  $S \rightarrow \epsilon$

$$SAB \rightarrow eSb$$

7) Tipo 0

8) Costruire il linguaggio  $L = \{e^n \mid n \geq 1\}$

$$S \rightarrow eS | e$$

8) Soluzione

PS: è di tipo 3

