VIII

SINTASSI ASTRATTA

54.-Descrizione algebrica di grammatiche BNF

Consideriamo il semplice linguaggio programmativo descritto dalla seguente grammatica in BNF

<Comando> =:: (<Identificatore> ← <Cifra>) |
(<u>If</u> <Booleano> <u>then</u>(<Comando>)<u>else</u> (<Comando>))

<Booleano>=:: (<Identificatore> = <Identificatore>) |
true | false

<Identificatore> =:: I | J

<Cifra> =:: Ø | 1 |.... | 9

Definiamo l'algebra $\underline{\underline{L}}$

$\underline{\underline{L}} = (L_{cmd}, L_{bool}, L_{id}, L_{cfr}, true, false, I, J, \emptyset, \ldots, 9, assign,$
$equal, cond)$

$L_{id} = \{I, J\}$

$L_{cfr} = \{\emptyset, 1, \ldots, 9\}$

$L_{bool} = \{True, False\} \cup \{H = K \mid H, K \in L_{id}\}$

$L^{0}_{cmd} = \{(H \leftarrow n) \mid H \in L_{id},\ n \in L_{cfr}\}$

$L^{i+1}_{cmd} = \{(\underline{If}\ B\ \underline{then}\,(C_1)\,\underline{else}\,(C_2)) \mid B \in L_{bool},\ C_1, C_2 \in L^{i}_{cmd}\}$

$L_{cmd} = \bigcup_{i \in N} L^{i}_{cmd}$

$assign : L_{id} \times L_{cfr} \rightarrow L_{cmd}$

$equal : L_{id} \times L_{id} \rightarrow L_{bool}$

$cond : L_{bool} \times L_{cmd} \times L_{cmd} \rightarrow L_{cmd}$

ove

$assign(H,n) = (H \leftarrow n)\ \forall H \in L_{id},\ n \in L_{cfr}$

$equal(H,K) = (H = K)\ \forall H, K \in L_{id}$

$$\mathrm{cond}(B,C_1,C_2) = (\underline{\mathrm{If}}\ B\ \underline{\mathrm{then}}(C_1)\underline{\mathrm{else}}(C_2))$$

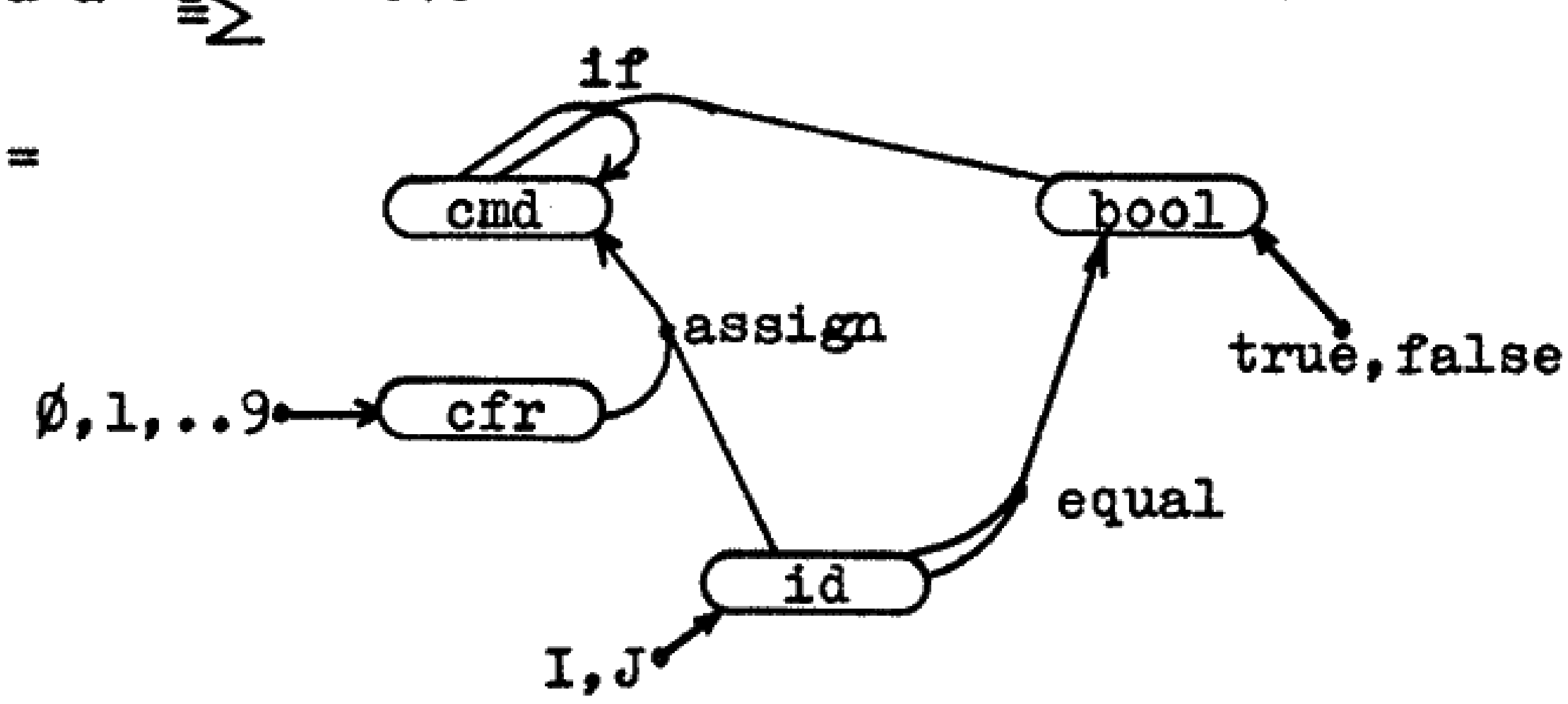
$$\forall\ B \in L_{bool},\ C_1, C_2 \in L_{cmd}$$

È evidente che $\underline{\underline{L}}$ rappresenta completamente la grammatica sopra data , infatti per ogni simbolo di categoria sintattica , in $\underline{\underline{L}}$ vi è l'insieme dei costrutti del linguaggio relativi alla categoria sintattica e ogni produzione è rappresentata da una operazione dell'algebra.

Se vogliamo esprimere la peculiarità della sintassi di tale linguaggio a prescindere da tutte le possibili varianti concrete (per esempio l'uso di parole chiave diverse per l'If o per l'uguaglianza) possiamo utilizzare in modo naturale la rappresentazione algebrica ottenuta affermando che la sintassi astratta di L è la classe di tutte le algebra isomorfe di $\underline{\underline{L}}$ che in quanto tali identificano varianti concrete della stessa sintassi astratta. Tale procedimento può applicarsi in modo ovvio a qualsiasi grammatica in BNF.

Inoltre poichè una segnatura Σ individua univocamente l'algebra $\underline{\underline{T}}_{\Sigma}$ una sintassi astratta può essere rappresentata tramite una segnatura. Se consideriamo per esempio il linguaggio di sopra è facile verificare che $\underline{\underline{L}}$ è isomorfa a $\underline{\underline{T}}_{\Sigma}$ ove

$\Sigma =$



Analogamente qualsiasi grammatica BNF (non ambigua) può essere identificata con una segnatura e in tal senso essere vista come sintassi astratta.

In generale per determinare una sintassi astratta di un qualsiasi linguaggio formale basta descriverne una sintassi come **algebra** .

55.- Esempi di sintassi algebrica di linguaggi formali

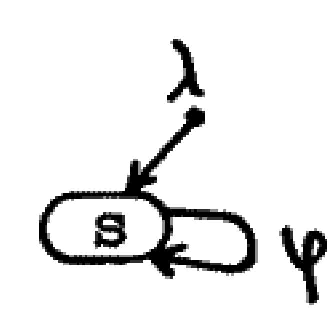
Consideriamo per esempio una sintassi astratta per

$$L_1 = \{a^m b^n \mid n \in N\}$$

$$\underline{\underline{L}}_1 = (L_1, \varphi_1, \lambda) \qquad \text{ove} \quad \begin{cases} \varphi_1 : L_1 \to L_1 \\ \varphi_1 x = axb \quad \forall\, x \in L_1 \end{cases}$$

È evidente che $\underline{\underline{L}}_1$ è isomorfo a $\underline{\underline{T}}_{\Sigma_1}$ ove

$\Sigma_1 =$



basta definire per induzione h $\begin{cases} h(\lambda) = \lambda \\ h(axb) = h(\varphi_1 x) = \varphi x \end{cases}$

per avere un isomorfismo da L_1 in T_{Σ_1}

$\underline{\underline{T}}_{\Sigma_1}$ individua anche la sintassi astratta del lin= guaggio $L_2 = \{a^n b^n c^n \mid n \in N\}$ basta a tal fine definire

$$\underline{\underline{L}}_2 = (L_2, \varphi_2, \lambda) \quad \text{ove} \quad \varphi_2(a^n b^n c^n) = a^{n+1} b^{n+1} c^{n+1} \quad \forall n \in N$$

Come sopra si verifica facilmente l'isomorfismo tra $\underline{\underline{L}}_2$ e T_{Σ_1}

Ricordiamo che L_1 è un linguaggio context- free mentre L_2 è contex-sensitive, questo ci fa notare che la sintassi astratta di un linguaggio esprime proprietà strutturali più profonde della classificazione secondo **Chomsky**(cfr. [21]).

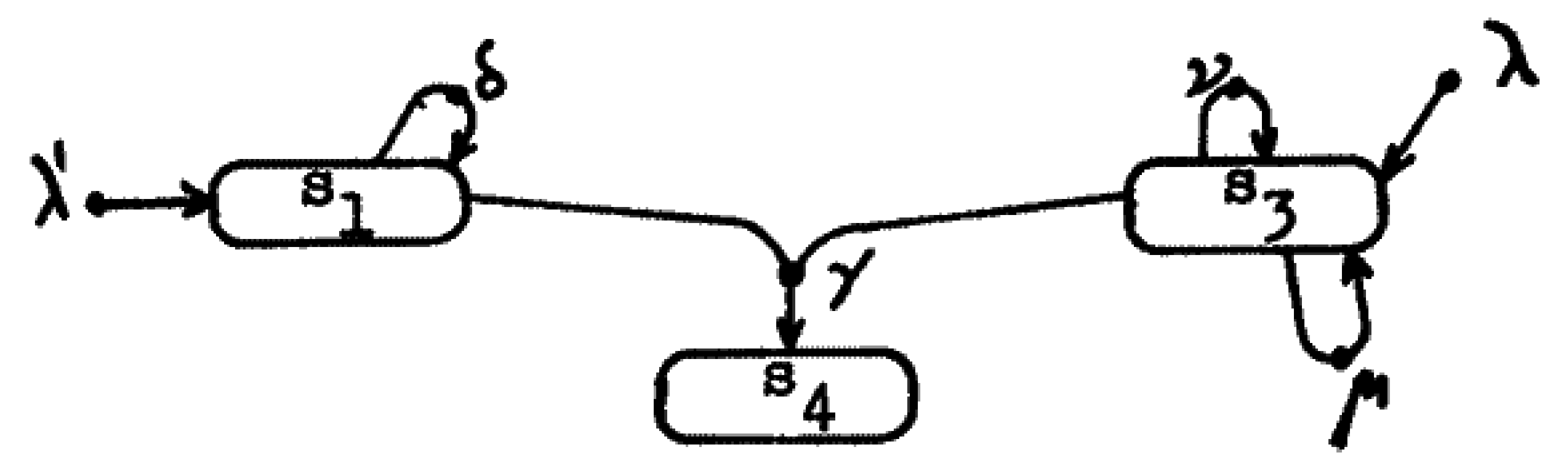
La sintassi astratta è in sostanza una sorta di ponte tra sintassi concreta e semantica, in quanto evidenzia unicamente le categorie sintattiche dei costrutti del linguaggio e per ogni costruzione sintattica le categorie sintattiche degli argomenti e quelle del risultato.

Come altro esempio definiamo una sintassi astratta di

$$L_4 = \{\alpha\beta \mid \alpha \in L_1 ,\ \beta \in L_3\}$$

ove L_1 è come sopra e $L_3 = \{a^n b^m \mid n, m \in N\}$

Basta definire la segnatura Σ_4



e considerare

$$\underline{\underline{L}}_4 = (L_1, L_3, L_4,\ \lambda_4;\ \lambda'_4,\ \gamma_4,\ \delta_4,\ \nu_4, \mu_4)$$

ove $\lambda_4 = \lambda'_4 = \lambda$ $\qquad \gamma_4 : L_1 \times L_3 \rightarrow L_4$

$$\delta_4 : L_1 \rightarrow L_1$$

$$\nu_4, \mu_4 : L_3 \rightarrow L_3$$

$\gamma_4(x,y) = xy$ (concatenazione)

$\delta_4 x = axb$

$\nu_4 x = ax$

$\mu_4 x = bx$

T_{Σ_4} risulta evidentemente una sintassi astratta per L_4.

56.- Sintassi algebrica di linguaggi a struttura di frase

In generale dato un linguaggio L a struttura di frase generato dalla grammatica $G=(V,T,S,P)$ (dove V:alfabeto, T:terminali, S:simbolo iniziale, $P=\Pi_1 \ldots \Pi_n$ produzioni) la sua rappresentazione algebrica è data da

$$\underline{\underline{L}} = (L, V^*, N, \varphi_o, \varphi_i, S)$$

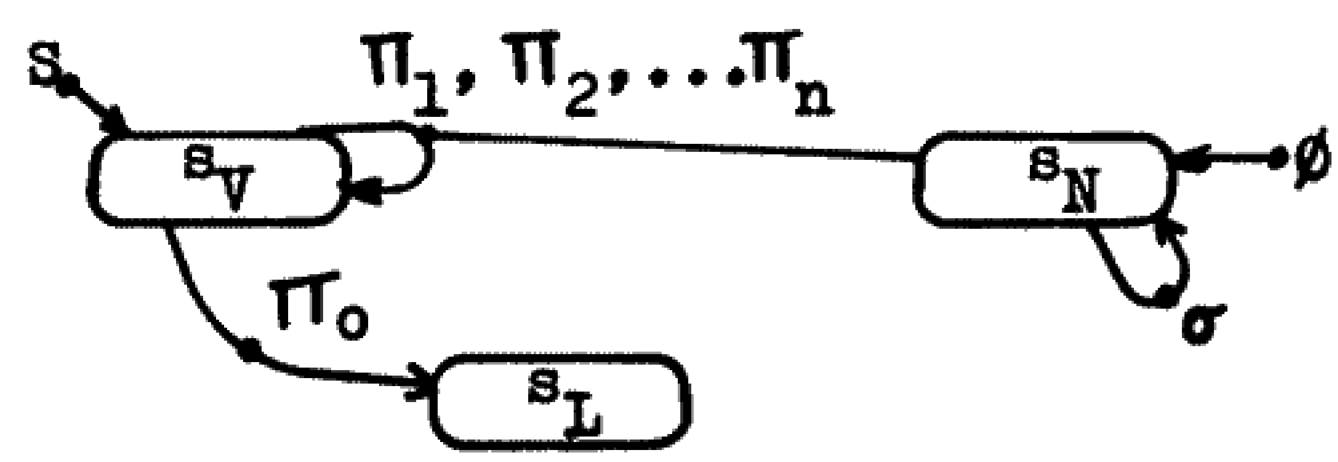
ove $\forall\, \Pi_i \in P^*$ $\quad \varphi_i : V^* \times N \rightarrow V^*$

$$\varphi_i(n,x) = \begin{cases} y & \text{se } \Pi_i : \alpha - \beta \text{ e } y \text{ è ottenuta da } x \\ & \text{sostituendo l'ennesima occorrenza} \\ & \text{di } \alpha \text{ con } \beta \\ x & \text{altrimenti} \end{cases}$$

$$\varphi_o(x) = \begin{cases} x & \text{se } x \in T^* \\ x_o & \text{altrimenti} \end{cases}$$

(x_o elemento fissato di L che supponiamo essere non vuoto).

Un'algebra che rappresenta la sintassi astratta di L è quindi un'algebra quoziente di T_Σ ove :

Σ =


(i naturali sono generati dallo zero tramite successore)

57.- ESERCIZI

Descrivere algebricamente sintassi astratte per i linguaggi

$$\{a^n b^m c^n \mid n,m \in N\}$$
$$\{a^n b^n b^n a^n \mid n \in N\}$$
$$\{a \alpha b \mid \alpha \in \{a,b\}^*\}$$