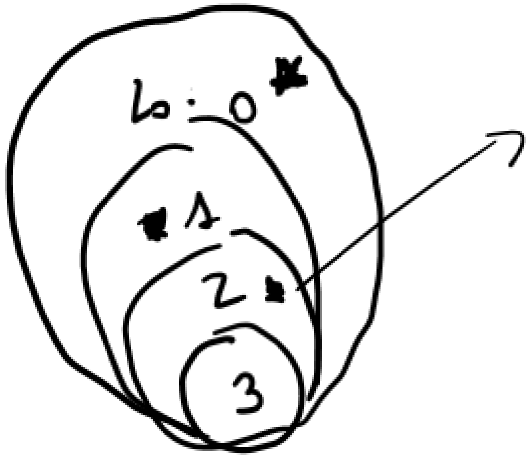
GERARCHIA DI CHOMSKY



$L = \{ a^n b^n \mid n \geq 1 \}$

SIA DA GRAMMATICHE DI TIPO 0

perché questo ammette ε-prod

sia da grammatiche di tipo 2 (non contestuali)

NON TIPO 3

$L_0 \in$ TIPO 0

TIPO 1

SE ANALIZZIAMO GRAMMATICHE DI TIPO 2 e 3 è consentita la presenza di una/più ε-produzione/i, mentre nel caso di $\boxed{\text{tipo 1}}$ è ammessa una ε-produzione solo sull'assioma ossia su S, a condizione che questo non compaia nelle parte destra di una produzione.

GRAMMATICA: $\boxed{\text{TIPO 3}}$?

$$\mathcal{G}: \; S \to aS \mid bA$$
$$A \to bA \mid \varepsilon$$

$$A \to \varepsilon$$

$$\mathcal{G} : \{ \quad , \underline{A}, P \}$$

TOGLIAMO PER FARE ESERCIZIO TOLGO $A \to \varepsilon$

$$\mathcal{G}: \; S \to aS \mid bA$$
$$A \to bA \mid \varepsilon$$

RELATIVA ESPRESSIONE REGOLARE

a^*bb^*

a^*b^+

q_p, q_1, q_2

CAPITOLO 4

LINGUAGGI NON CONTESTUALI

inizialmente queste sono state introdotte come strumento per definire le strutture del linguaggio naturale.

Possiamo caratterizzare le fasi di un linguaggio con soggetto e predicato e quindi comporre una frase,

soggetto = sostantivo

$F \to SP$ FRASE → SOGGETTO + PREDICATO

$P \to V|VC$ PREDICATO → VERBO | VERBO + COMPLEMENTO OGGETTO

QUESTE SONO STRUTTURE SUFFICIENTEMENTE SEMPLICI e quindi in alcuni contesti inadeguate come strumento per il linguaggio naturale.

I linguaggi di programmazione rappresentano un sottoinsieme proprio dei linguaggi non contestuali e li caratterizziamo come di seguito.

ALBERI DI DERIVAZIONE O ALBERI SINTATTICI

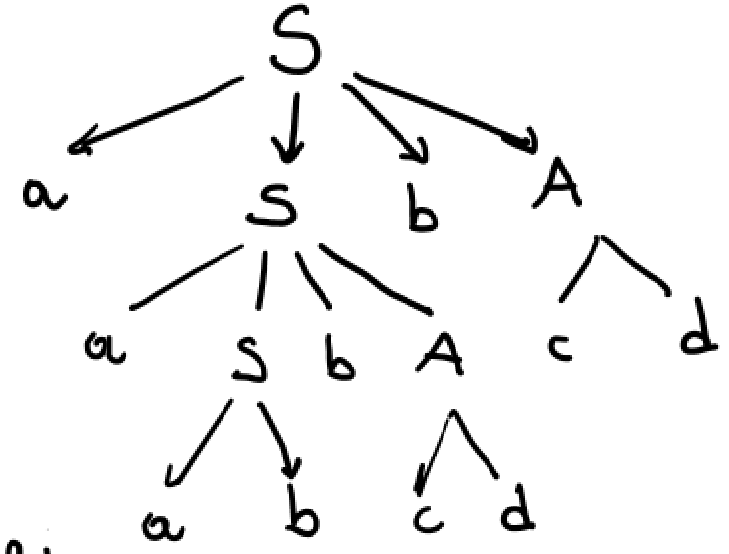
G: $S \to aSbA \mid ab$

$A \to cAd \mid cd$

$\to aSbA \to aabbA \to aabbcd \in \mathcal{L}(G)$

LA GRAMMATICA INTRODUCE il linguaggio $\mathcal{L}(G)$ $\quad aaabbcdbcd \in \mathcal{L}(G)$

QUESTI LINGUAGGI USANO UNA MEMORIA GESTITA A PILA



Un esempio di albero di derivazione di una stringa $\in \mathcal{L}(G)$

e nel caso di linguaggi di programmazione questa corrisponde ad un'analisi sintattica "parsing" relativa al programma di "tradurre" la stringa in un'altra stringa + elementare.

CAPITOLO 5

MACCHINE DI TURING e CALCOLABILITÀ SECONDO TURING
(MT)

Le MT sono il modello di calcolo di riferimento fondamentale sia nell'ambito della teoria di calcolabilità sia in quella della teoria della complessità computazionale.

L'OBIETTIVO è quello di formalizzare il concetto di calcolo allo scopo di stabilire l'esistenza di metodi algoritmici per il riconoscimento dei teoremi nell'ambito dell'ARITMETICA

- ELEVATA SEMPLICITÀ STRUTTURALE
- POTERE COMPUTAZIONALE

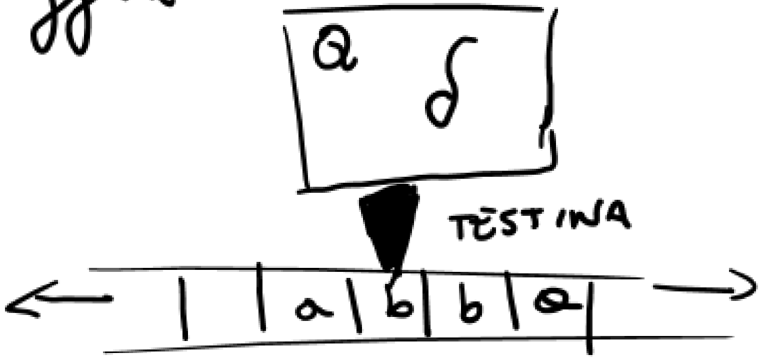
LA MACCHINA DI TURING VIENE DEFINITA come un dispositivo operante su stringhe contenute su una memoria esterna (nastro) mediante passi elementari definiti da funz. di transizione.

MACCHINA DI TURING A NASTRO SINGOLO (MT)

Nella MT vi è un nastro potenzialmente infinito diviso in celle contenenti ciascuna un simbolo appartenente ad un dato alfabeto Γ ampliato con il carattere speciale $\bar{b}$ (blank) che rappresenta la situazione di cella non contenente caratteri.

La MT lavora su un nastro, e vi è una testina che scorre su di esso in entrambe le direzioni.

Su ogni cella si può leggere o scrivere caratteri dell'alfabeto Γ opp $\bar{b}$



Ad ogni istante la MT si trova in uno stato appartenente ad insieme finito Q e il meccanismo che fa evolvere la computazione della macchina è la funzione di transizione.

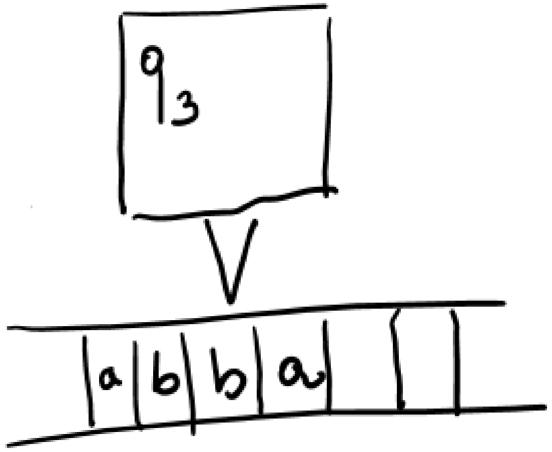
δ parte da uno stato e da un carattere $\longrightarrow$ otteniamo un altro stato

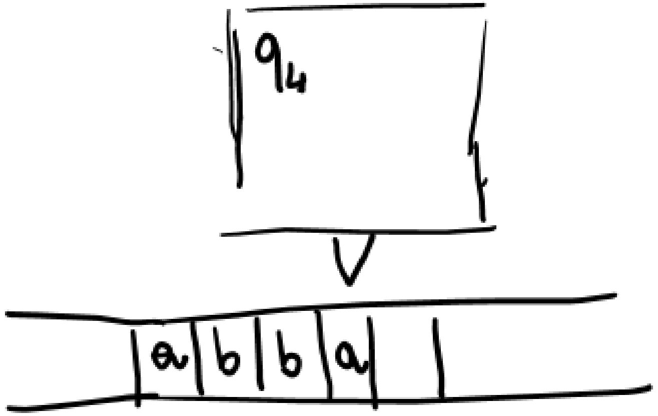
DEFINIZIONE MT $\quad \bar{\Gamma} = \Gamma \cup \{\bar{b}\} \quad \Sigma \subseteq \bar{\Gamma} \quad \Gamma \subseteq \bar{\Gamma}$

Una macchina di Turing deterministica (MTD) è una sestupla $\mathcal{M} = \langle \Gamma, \bar{b}, Q, q_0, F, \delta \rangle$, dove Γ è l'alfabeto, $\bar{b}$ è il simbolo blank, $\bar{b} \notin \Gamma$, Q è l'insieme degli stati $\neq \emptyset$, q_0 è lo stato iniziale, $F \subseteq Q$ è l'insieme degli stati finali, δ è la funzione di transizione è definita

$$\delta : \overset{\text{STATO NON FINALE}}{(Q - F)} \times \overset{\text{SIMBOLO O BLANK}}{(\Gamma \cup \{\bar{b}\})} \mapsto \overset{\text{STATO}}{Q} \cup \overset{\text{SIMBOLO O BLANK}}{(\Gamma \cup \{\bar{b}\})} \times \overset{\text{SPOSTAMENTO / MOVIMENTO}}{\{d, s, i\}}$$

d, s, i indicano lo spostamento a dx, sx o assenza di spostamento della testina





$$\delta \; (Q - F) \times (\Gamma \cup \{\not b\}) \longrightarrow Q \times (\Gamma \cup \{\not b\}) \times \{d, s, i\}$$

$$(q_3, b) \mapsto (q_4, a, d)$$

$$\delta(q_3, b) = (q_4, a, d)$$

Le macchine usate per accettare stringhe vengono dette riconoscitori, mentre quelle usate per calcolare funzioni vengono dette di tipo trasduttore

FUNZIONE CARATTERISTICA

FUNZIONE CARATTERISTICA del linguaggio che è definita da Σ^* a $\{0,1\}$ è una funzione che assume valore 1 per ogni stringa del linguaggio e valore 0 altrimenti

"$A \to \alpha a$ $\quad A \in V_N$ $\boxed{a \in V_T}$ $\boxed{\alpha \in V_N^*}$ "