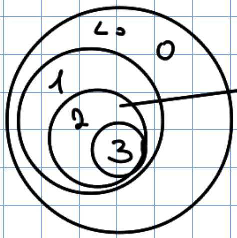
Gerarchia di Chomsky



$L = \{a^n b^n \mid n \geq 1\}$ può essere generato da grammatiche di tipo 0 perché ammette ε-produzioni sia da grammatiche di tipo 2 (non contestuali) non tipo 3

L_0 è tipo 0 ma potrebbe non appartenere a tipo 1|2|3

Osservazione:

Se analizziamo grammatiche di tipo 2/3 è consentita la presenza di una (più) ε-produzione, mentre nel caso di tipo 1 è ammessa una ε-produzione solo sull'assioma su S, a condizione che questo non compaia mai nelle parti destre di una produzione

Un linguaggio si dice regolare se: "possiamo procedere senza fermarci"

Grammatica $\boxed{\text{Tipo 3?}}$

$$G: S \to aS \mid bA$$
$$A \to bA \mid \varepsilon$$

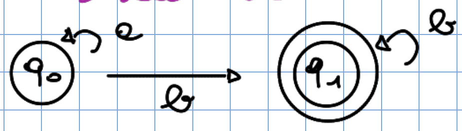
Togliamo la ε-produzione per fare esercizio

$$G: S \to aS \mid bA$$
$$A \to bA$$

→ relativa espressione regolare

$a^* b b^*$ oppure $a^* b^+$

ASFD relativo



CAPITOLO 4: linguaggi non contestuali

Inizialmente queste sono state introdotte come strumento per definire la struttura del linguaggio naturale.

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

$F \rightarrow SP$ frase → soggetto + predicato

$P \rightarrow V \mid VC$ predicato → verbo | verbo + complemento oggetto

Queste strutture sono sufficientemente semplici e quindi in alcuni contesti inadeguate come strumento per il linguaggi naturali.

I linguaggi generati non contestuali rappresentano un sottoinsieme proprio dei linguaggi

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

Alberi di derivazione o alberi sintattici

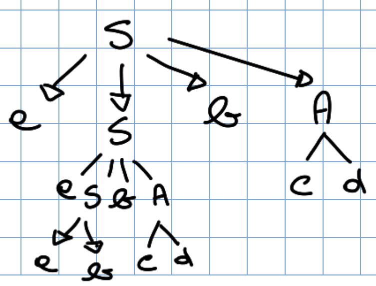
G: $S \rightarrow aSbA \mid ab$

$A \rightarrow cAd \mid cd$

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

La grammatica introduce il linguaggio $\mathcal{L}(G)$ $aaabbcdbcd \in \mathcal{L}(G)$

un'esempio di albero di derivazione di una stringa $\in \mathcal{L}(G_1)$



Queste grammatiche usano una memoria gestita a pila e nel caso di linguaggi di programmazione queste corrisponde ad un'analisi sintattica "parsing" relativa al programma di "tradurre" la stringa in un'altra stringa più elementare

capibile dal PC

CAPITOLO 5:

Macchine di turing e calcolabilità secondo turing

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

L'obbiettivo 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

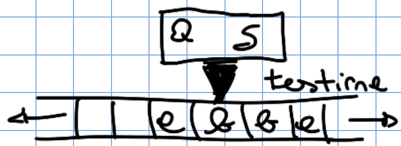
Macchine di Turing a nastro singolo

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 celle non contenente caratteri

↳ gamma

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

Si può leggere o scrivere caratteri dell'alfabeto Γ opp $\bar{b}$



Ad ogni istante la MT si trova in uno stato appartenente ad un 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 → stato

Definizione MT

Una macchina di Turing deterministica (MTD) è una sestupla $M = \langle \Gamma, \bar{b}, Q, q_0, F, \delta \rangle$

Γ = alfabeto

$\bar{b}$ = blank $\notin \Gamma$ "b sovrassegnato"

$\bar{\Gamma} = \Gamma \cup \{\bar{b}\}$

Q = insieme degli stati

q_0 = stato iniziale

F = stati finali $F \subseteq Q$

δ = funzione di transizione definita così:

$$\delta : \underbrace{(Q - F)}_{\text{STATO NON FINALE}} \times \underbrace{(\Gamma \cup \{\bar{b}\})}_{\text{SIMBOLO}} \mapsto \underbrace{Q}_{\text{stato}} \cup \underbrace{(\Gamma \cup \{\bar{b}\})}_{\text{simbolo}} \times \underbrace{\{d, s, i\}}_{\text{spostamento}}$$

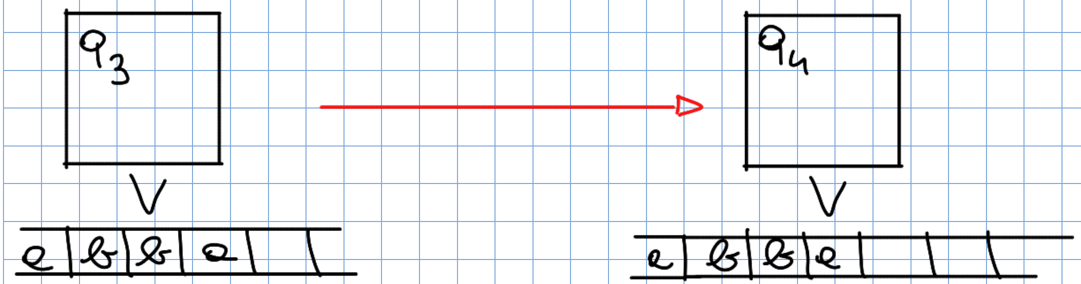
$\{d, s, i\}$ indicano lo spostamento a dx, sx o assenza di spostamento della testina

$\bar{\Gamma} = \Gamma \cup \{\bar{b}\}$ per convenzione

e soprattutto $\Sigma \subseteq \bar{\Gamma}$

Esempio:

$$\delta : (Q - F) \times (\Gamma \cup \{\bar{b}\}) \to Q \times (\Gamma \cup \{\bar{b}\}) \times \{d, s, i\}$$



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

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

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

Funzione caratteristica del linguaggi che è definita da Σ^* a $\{0, 1\}$ ed è ovviamente una funzione che assume valore 1 per ogni stringa del linguaggio o valore 0 altrimenti

Forma normale di Greibach

"$A \to \alpha a$" $\quad A \in V_N \quad a \in V_T$

$\alpha \in V_N^*$