

Definizione

Un Linguaggio L è un codice se ogni parola in L^+ è decomponibile in un unico modo in parole di L

es negativo

$$L = \{a, b, ab\}$$

$$\begin{array}{c} abab \in L^+ \\ \swarrow \quad | \quad \searrow \\ a \cdot b \cdot a \cdot b \quad \dots \quad ab \cdot ab \end{array}$$

NON è un codice

es positivo:

$$L = \{aa, ab, b\}$$

$$ab|aa|b|b \in L^+$$

L è un codice

Definizione

L è un codice prefisso o istantaneo se ogni parola di L non è prefisso di altre parole di L

es negativo:

$$L = \{0, 01\} \text{ è codice ma } \underline{\text{non}} \text{ prefisso}$$

es positivo

$L = \{aa, ab, b\}$ è codice prefisso

CODICE ASCII ESTESO

Ogni carattere è codificato con 8-bit
e da origine:

$$C_A = \{x \in \{0,1\}^* \mid |x| = 8\}$$

$$C_A^+ = \text{"file binari"}$$

Perché è importante che C_A sia un codice?

Risposta:



* visto che C_A è un codice si ottiene il file
originario a caratteri

- ancora C_A è prefisso

ogni byte si differenzia dagli altri per almeno un bit

es 00011001 ≠ 00001001

- Dato che è prefisso ammette un algoritmo
di decodifica on-line:

Taglio il file binario ogni 8-bit

LINGUAGGIO = PROBLEMA

Definizione di linguaggi

•) $L = \{w_1, w_2, \dots, w_n\}$ | estensiva
solo se è finito

•) $L = \{w \in \Sigma^* \mid P(w) = 1\}$ | intensivo
↓
Sigma star

Dove P è proprietà,

si può usare anche per L infinito

FATTO:

Ad ogni L è associato il problema P_L

Linguaggio $L \iff$ Problema P_L

$\{w \in \Sigma^* \mid P(w) = 1\}$ Input: $w \in \Sigma^*$

Output:

w soddisfa P ?

Sì No

Problema di decisione

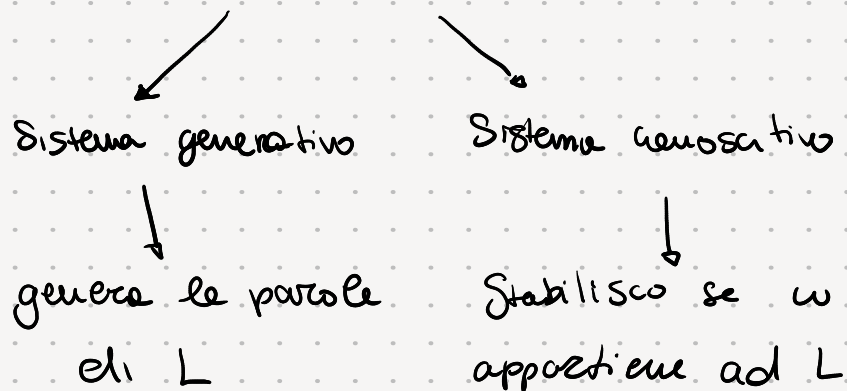
Dato P_L siamo interessati a:

1) sapere se P_L ammette una soluzione automatica

2) Se P_L ammette algoritmo, trovare il migliore

Tutto questo si traduce nella teoria dei linguaggi in:

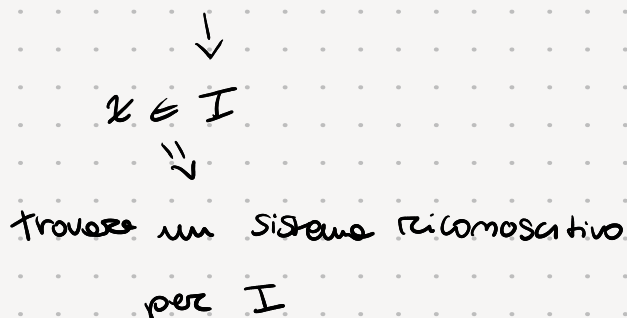
1) Sapere se L ammette un sistema formale



2) Se L ammette un sistema riconoscente trovare quello migliore

es $I = \text{linguaggio indirizzi internet} =$
 $= \{ x \in \{0, 1, .\}^* \mid x = x_1 \cdot x_2 \cdot x_3 \cdot x_4$
dove $x_i \in \{0, 1\}^* \wedge |x_i| = 8 \}$

Software in rete si chiedono se un indirizzo è corretto



es \hat{P} = linguaggio delle password di un sito web =

$$= \{ w \in \{ a, \dots, z, A, \dots, Z, 0, 1, \dots, 9 \}^* \mid |w| = 8 \wedge \exists i w_i \in \{ A, \dots, Z \} \wedge \exists j w_j \in \{ 0, \dots, 9 \} \}$$

si chiede che le password siano generate in maniera sistematica



trovare un sistema generativo per \hat{P}

Domanda:

Tutti i problemi di decisione ammettono soluzione automatica?

Risposta: Teoria della calcolabilità

da dei risultati indipendenti della tecnologia

Concetti base:

- procedura: sequenza finita di istruzioni che possono portare ad un risultato
- algoritmo: è una procedura che termina su ogni input

• programma: cos'è?



① un programma è una parola binaria
 $w \in \{0, 1\}^*$ grazie al codice ASCII

② un programma è una funzione
e si indica:

\mathcal{F}_w = semantica del programma w

$\mathcal{F}_w(x)$ = risultato di w su input x

Oss. Un qualsiasi input x per w è
binario: $x \in \{0, 1\}^*$

ma anche $w \in \{0, 1\}^*$

\Downarrow

quindi anche w può essere passato
ad un programma in input

Limitazione sui programmi:

- 0 non terminano
- 0 se terminano ma danno
un uscita $\begin{matrix} 0 \\ - \\ 1 \end{matrix}$

NOTAZIONE:

$\mathcal{F}_w(x) \uparrow$ w su input x non termina

$\mathcal{F}_w(x) \downarrow$ " " " " termina

$\mathcal{F}_w(x) = 1$ se output è 1

$\mathcal{F}_w(x) = 0$ " " " è 0

es Problema: calcolare la parità dei
 numeri binari positivi
 cioè la parità di stringhe della forma

$$x \in \{0, 1\}^*$$

Programma

```
bin-parità- ( $x = x_1 x_2 \dots x_n \in \{0, 1\}^*$ ) {
  IF ( $x_1 = 1$ ) then return  $\underbrace{1 - x_n}_{\neg x_n}$ ;
  loop;
}
```

$$F_v(x) = \begin{cases} 1 & \text{se } x \text{ è binario pari} \\ 0 & \text{se } x \text{ è binario dispari} \\ \uparrow & \text{se } x \text{ non è binario} \end{cases}$$

$$F_v(x) = \begin{cases} 1 & \text{se } x \in \{0, 1\}^* 0 \\ 0 & \text{se } x \in \{0, 1\}^* 1 \cup \{1\} \\ \perp & \text{se } x \in 0\{0, 1\}^* \cup \{\varepsilon\} \end{cases}$$

$F_v \leftarrow$ la semantica del programma v
 è una funzione:

$$F_v: \underbrace{\{0, 1\}^*}_{\text{input}} \rightarrow \underbrace{\{1, 0, \perp\}}_{\text{output}}$$