Definizione

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

Es negativo

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

$$abab \in L^+$$

$$a \cdot b \cdot a \cdot b \quad \dots \quad ab \cdot ab$$

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:

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

Es positivo

$$L = \{ aa, ab, b \} \text{ é codice prefisso}$$

CODICE ASCII ESTESO

Ogni carattere è codificato con <u>8-bit</u>
e da origine:

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

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

Perché é importante che C_A sia un codice?

Risposta:

file a caratteri $\xrightarrow[\text{CODIFICA}]{\text{SALVO}}$ file binario $\xrightarrow[\text{decodifica}]{\text{RIAPRO}}$ file a caratteri *

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

- ancora C_A é prefisso
  ogni byte si differenzia dagli altri per almeno un bit

  Es 00011001 ≠ 00001001

- Dato che é 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 <=> Problema P_L

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

OUTPUT:

w soddisfa P?

$\underbrace{\text{sì} \qquad \text{no}}_{\text{Problema di decisione}}$

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

- Sistema generativo ↓ genera le parole di L
- Sistema riconoscitivo ↓ Stabilisco se w appartiene ad L

2) Se L ammette un sistema riconoscitivo trovare quello migliore

Es I = linguaggio indirizzi internet =

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

Software in rete si chiedono se un indirizzo é corretto

↓

$x \in I$

⇓

trovare un sistema riconoscitivo per I

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

$\Downarrow$

trovare un sistema generativo per $\hat{P}$

Domanda:

Tutti i problemi di decisione ammettono soluzione automatica?

Risposta: Teoria della calcolabilità

da dei risultati indipendenti dalla tecnologia

Concetti base:

- procedura: sequenza finita di istruzioni che possono portare ad un risultato
- algoritmo: é una procedura che termina su ogni input
- programma: cos'é?

ω

① aspetto sintattico ② aspetto semantico

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

② Un programma è una funzione e si indica:

F_w = semantica del programma w

$F_w(x)$ = 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:

- o non terminano
- o se terminano mi danno in uscita $<^0_1$

NOTAZIONE:

$F_w(x)\uparrow$ w su input x non termina

$F_w(x)\downarrow$ " " " " termina

$F_w(x) = 1$ se output è 1

$F_w(x) = 0$ " " è 0

Es Problema: calcolare la parità dei numeri binari positivi

cioè la parità di stringhe della forma

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

Programma

```
bin_parità (x = x_1 x_2 ... x_n ∈ {0,1}*) {
    if (x_1 = 1) then return 1 - x_n;
    loop;
}
```

$\underbrace{1-x_n}_{\neg x_n}$

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

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

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

$$F_v : \underset{\text{input}}{\{0,1\}^*} \to \underset{\text{output}}{\{1, 0, \perp\}}$$