Capitolo 2:
Descrivere un linguaggio di programmazione

Linguaggi (Naturali o Artificiali)

La descrizione di un linguaggio avviene su 3 dimensioni:

- <u>SINTASSI</u> (regole di formazione / quando una frase è corretta? "relazione tra segni")
- <u>SEMANTICA</u> (attribuzione di significato / cosa significa una frase corretta? / "relazione tra segni e significato"
- <u>PRAGMATICA</u> (in quale modo frasi corrette e sensate sono usate / "relazione tra segni, significati e utente"

Per un linguaggio eseguibile:

- <u>IMPLEMENTAZIONE</u> (come eseguire una frase corretta, rispettandone la semantica)

Sintassi

- Aspetto Lessicale (parole che si possono usare)
    - descrizione del lessico
        - Dizionari (x ling. naturali)
        - Strutture più complesse per i ling. artificiali
    - errore? Vocabolo inesistente
      es: "canu", "thon"...
- Aspetto Grammaticale (frasi corrette che si possono costruire con il lessico)
    - descrizione attraverso regole grammaticali (in numero finito)
    - Le frasi generabili sono infinite
    - errore? frase scorretta
      es: "La cane abbaiano"
      (lessico corretto, ma frase grammaticamente scorretta)

SEMANTICA

- Per il lessico ⟶ Dizionari
- Per le frasi, devo sapere
  1) a quale ling. la frase appartiene!!

     es: I VITELLI DEI ROMANI SONO BELLI

     (VA, O VITELLIO, AL SUONO DEL DIO ROMANO DELLA GUERRA)

  2) su quale linguaggio basarmi per dare significato (bisogna basarsi su qualcosa di noto, cioè che non abbia a sua volta bisogno di essere spiegato!)

Frasi corrette in $\mathcal{L}_1$ $\xrightarrow[\text{ling. naturale}]{\text{traduzione}}$ Significato espresso con frasi corrette in $\mathcal{L}_2$ (ad es. italiano)

Frasi corrette in $\mathcal{L}$ $\xrightarrow[\text{ling. di prog. sequenziale}]{\text{funzione di interpretazione semantica}}$ Funzioni calcolate (da input a output)

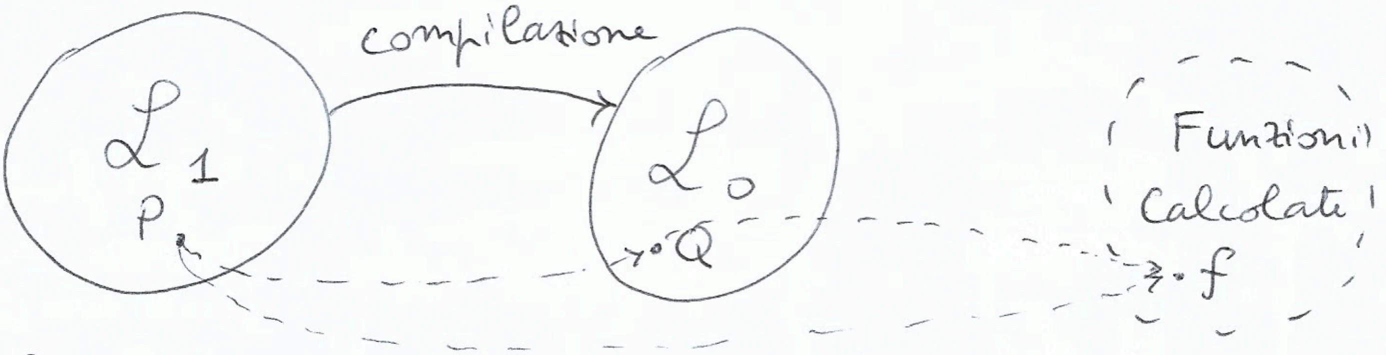
PRAGMATICA

- insieme di regole che guidano l'uso

Es: • usare il "lei" invece del "tu" quando ci si rivolge ad uno sconosciuto
• evitare di usare i "goto" in programmazione

IMPLEMENTAZIONE

IMPLEMENTAZIONE: eseguire una frase sintatticamente corretta rispettandone la semantica



Programma sorgente

P: $I = I + 1$

Tradotto in $\mathcal{L}_0$

Q: MOV I R_0
ADD 1 R_0
MOV R_0 I

La semantica di P è la funzione f, che è pure la semantica del programma compilato Q: l'implementazione Q di P preserva la semantica di P!

Lessico e Frasi di un Linguaggio

Alfabeto a, b, c, ... (tipicamente finito)

→ **Lessico**: insieme di sequenze finite (parole) costituite con caratteri (simboli) dell'alfabeto

→ **Frasi**: insieme di sequenze finite costituite con parole del lessico (tipicamente infinito contabile)

Oss: "Lessico" è un "Alfabeto" per "Frasi"!

⇒ Astraiamoci e definiamo cos'è un LINGUAGGIO FORMALE

- Un alfabeto A è un insieme finito i cui elementi sono detti simboli
- Una parola su alfabeto A è una sequenza finita di simboli in A
- Un linguaggio formale L su A è un insieme di parole su A

Un ling. formale L su alfabeto A è un sottoinsieme di A^* ($L \subseteq A^*$), dove

$$A^* = \bigcup_{n \geq 0} A^n \qquad \text{dove} \quad A^0 = \{\varepsilon\}$$

parola vuota (sequenza senza alcun simbolo)

$$\text{e} \quad A^{n+1} = A \circ A^n \qquad n \geq 0$$

$$\text{con} \quad A \circ A^n = \{ aw \mid a \in A \text{ e } w \in A^n \}$$

concatenato

Osservazione: A^* è un insieme infinito contabile

Dato un ordinamento $<$ sui simboli di A (ad es: $a < b < c \ldots$), possiamo elencare tutte le parole in A^* come segue:

- prima elenco la parola vuota ε (A^0)
- poi elenco le parole di lunghezza 1 (A^1) secondo l'ordinamento $<$

  $a, b, c, \ldots$

- poi elenco le parole in A^2 secondo $<$

  $aa, ab, ac, \ldots, ba, bb, bc, \ldots$

- poi elenco le parole in A^3 secondo $<$

⋮

e così via

Se ammettessimo alfabeti con un numero infinito di simboli? (6 bis

$$A = \{ a_0, a_1, a_2, \dots \}$$

A^* sarebbe ancora contabile? Cioè è possibile elencare tutte le parole in A^*?

La risposta è affermativa!

A è in corrispondenza biunivoca con $\mathbb{N}$

- $\mathbb{N} \times \mathbb{N}$ è numerabile (dove-tailing)

  $f^2 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ biunivoca

- $\mathbb{N}^K$ è numerabile

$$f^K(x_1, x_2, \dots, x_K) = \text{if } K = 2 \text{ then } f^2(x_1, x_2) \text{ else } f^2(x_1, f^{K-1}(x_2, \dots x_K))$$

- $\mathbb{N}^* = \bigcup_{K \geq 0} \mathbb{N}^K$ è numerabile

$$f(x_1, \dots, x_K) = f^2(K, f^K(x_1, \dots, x_K))$$

Dove - Tailing

$x_2 \backslash x_1$	0	1	2	3	4	5	...
0	0	1	3	6	10	15	
1	2	4	7	11	16		
2	5	8	12	17			
3	9	13	18				
4	14	19					
5	20						
⋮							

si riempie la tabella "per diagonali"

Funzione di codifica: $f^2 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

$$f^2(x_1, x_2) = \frac{(x_1+x_2)(x_1+x_2+1)}{2} + x_2$$

Es: $f^2(2,1) = \frac{3 \cdot 4}{2} + 1 = 7$

Oss: • $x_1 + x_2$ è il numero di diagonali completamente riempite al di sotto della entry (x_1, x_2)

• la somma di tutti gli elementi in queste x_1+x_2 diagonali è:

$$1 + 2 + 3 + \dots + (x_1+x_2) = \frac{(x_1+x_2)\cdot(x_1+x_2+1)}{2}$$

(metodo di Gauss)

6 quater

Funzione di decodifica

$f_1^2 : \mathbb{N} \to \mathbb{N} \qquad f_2^2 : \mathbb{N} \to \mathbb{N}$

tali che $f_1^2\left(f^2(x_1, x_2)\right) = x_1$

$f_2^2\left(f^2(x_1, x_2)\right) = x_2$

Bisogna trovare
- $K_x = \max\left\{ i \mid \frac{i \cdot (i+1)}{2} \leq x \right\}$
- allora $K_x = x_1 + x_2$
- e quindi

$$x_2 = x - \frac{K_x(K_x+1)}{2}$$

$$x_1 = K_x - x_2$$

Es: $f_1^2(7) = 2$ perché $K_7 = 3$ $\left(\frac{3 \cdot 4}{2} \leq 7\right)$

e quindi

$$x_2 = 7 - \frac{3 \cdot 4}{2} = 1$$

$$x_1 = 3 - 1 = 2$$

Notazioni e Definizioni Ausiliarie

- **Lunghezza** di una parola (o stringa)

$$|\varepsilon| = 0 \qquad |aw| = 1 + |w| \qquad \text{es: } |abc| = 3$$

- **Concatenazione**: x concatenata a y, xy, è la parola ottenuta giustapponendo x e y.

$$w = xy \iff \begin{cases} \cdot\ |w| = |x| + |y| \\ \cdot\ w(j) = x(j) & \text{per } 1 \leq j \leq |x| \\ \cdot\ w(|x|+j) = y(j) & \text{per } 1 \leq j \leq |y| \end{cases}$$

($w(j)$ indica il j-esimo simbolo di w)

Leggi della concatenazione

$$x(yz) = (xy)z \qquad \text{associatività } xyz$$

$$x\varepsilon = x = \varepsilon x \qquad \varepsilon \text{ è elemento neutro}$$

- **Sottostringa** v di una stringa w

$\iff \exists\, x, y \in A^*$ tali che $w = xvy$
(dove x e y possono essere ε)

Oss: 1) Ogni stringa è sottostringa di se stessa

2) ε è sottostringa di ogni stringa

- Suffisso

v è suffisso di w sse $\exists x \in A^*.\ w = xv$

- Prefisso

v è prefisso di w sse $\exists x \in A^*.\ w = vx$

- Potenza n-esima di una stringa w

$$w^0 = \varepsilon \qquad w^{n+1} = w w^n \qquad \forall n \geq 0$$

Es: $(ab)^1 = ab \quad (ab)^2 = abab$

$(ab)^3 = ababab \quad (ab)^4 = abababab$

- Linguaggio L su alfabeto A : $L \subseteq A^*$

Se $A = \{a\}$, allora $\emptyset$, $\{\varepsilon\}$, $\{a, aaa\}$ sono ling. finiti.

$$L_1 = \{a, aa, aaa, \dots\} = \{a^n \mid n \geq 1\}$$
$$= A^* \setminus \{\varepsilon\}$$
$$\uparrow$$
$$= \{a^n \mid n \geq 0\}$$

$$L_2 = \{\varepsilon, aa, aaaa, \dots\} = \{a^{2n} \mid n \geq 0\}$$

$$L_3 = \{a, aaa, aaaaa, \dots\} = \{a^{2n+1} \mid n \geq 0\}$$

Operazioni su Linguaggi

- Complemento $\bar{L} = \{w \in A^* \mid w \notin L\} = A^* \setminus L$

- Unione e Intersezione (ovvi)

$$L_1 \cup L_2 = \{w \mid w \in L_1 \text{ o } w \in L_2\}$$
$$L_1 \cap L_2 = \{w \mid w \in L_1 \text{ e } w \in L_2\}$$

- Concatenazione

$$L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1 \text{ e } w_2 \in L_2\}$$

Oss: Concatenazione $\neq$ Prodotto Cartesiano

$$L_1 \times L_2 = \{(w_1, w_2) \mid w_1 \in L_1 \text{ e } w_2 \in L_2\}$$

I due insiemi non sono sempre equipotenti perché valgono le leggi della concatenazione

$$x(yz) = (xy)z \qquad x\varepsilon = x = \varepsilon x$$

Es: $L_1 = \{a, ab\}$ $\qquad L_2 = \{bc, c\}$

$$L_1 \cdot L_2 = \{abc, ac, abbc\}$$
$$L_1 \times L_2 = \{(a, bc), (a, c), (ab, bc), (ab, c)\}$$

N.B: $a(bc) = (ab)c = abc$

Esempi

- $L_1 = \{ a^n \mid n \geq 0 \} \quad L_2 = \{ b \}$

  $L_1 \circ L_2 = \{ a^n b \mid n \geq 0 \}$

- $L_1 = \{ a^{2n} \mid n \geq 0 \} \quad L_2 = \{ b^m \mid m \geq 0 \}$

  $L_1 \cdot L_2 = \{ a^{2n} b^m \mid n, m \geq 0 \}$

- $L_1 = \{ a^n b^n \mid n \geq 0 \} \quad L_2 = \{ b^n \mid n \geq 0 \}$

  $L_1 \cdot L_2 = \{ a^n b^{n+m} \mid n, m \geq 0 \}$

  $= \{ a^n b^m \mid m \geq n \geq 0 \}$

- $L_1 = \{ a^n \mid n \geq 1 \} \quad L_2 = \{ a^n b^n \mid n \geq 0 \}$

  $L_1 \cdot L_2 = \{ a^{n+m} b^m \mid n \geq 1, m \geq 0 \}$

  $= \{ a^n b^m \mid n > m \geq 0 \}$

- $L_1 = \{ a b^n \mid n \geq 1 \} \quad L_2 = \{ a, c \} \cup \{ b^n \mid n \geq 1 \}$

  $L_1 \cdot L_2 = \{ a b^n a \mid n \geq 1 \} \cup \{ a b^n c \mid n \geq 1 \}$

  $\cup \{ a b^n \mid n \geq 2 \}$

- $A = \{0, 1\} \quad L_1 = \{ w \in A^* \mid$ w contiene un numero pari di "0" $\}$

  $L_2 = \{ w \in A^* \mid w = 0y$ e $y \in \{1\}^* \}$

  $L_1 \cdot L_2 = \{ w \in A^* \mid$ w ha un numero dispari di "0" $\}$

POTENZA di un linguaggio

$$L^0 = \{\varepsilon\}$$

$$L^{n+1} = L \cdot L^n \qquad \forall n \geq 0$$

CHIUSURA / STELLA DI KLEENE / RIPETIZIONE

$$L^* = \bigcup_{n \geq 0} L^n$$

$$L^+ = \bigcup_{n \geq 1} L^n \qquad \text{(chiusura positiva)}$$

Esempi

$L = \{aa\}$ $\qquad L^* = \{\varepsilon, aa, aaaa, \dots\}$

$\qquad\qquad\qquad = \{(aa)^n \mid n \geq 0\}$

$L = \{a, aaa\}$ $\qquad L^* = \{a^n \mid n \geq 0\}$

$L = \{aa, aaa\}$ $\qquad L^* = \{a^n \mid n \neq 1\}$

Rappresentazione Finita dei Linguaggi

Se L è infinito, come possiamo rappresentarlo?
(e come possiamo determinare se una certa stringa $w \in L$?)

Esempio 1) $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ infinito contabile

- Non possiamo "memorizzare" $\mathbb{N}$; servirebbe una memoria infinita
- Possiamo trovare una rappresentazione finita implicita:

$$\frac{}{0 \in \mathbb{N}} \qquad \frac{x \in \mathbb{N}}{S(x) \in \mathbb{N}} \qquad \text{(Peano)}$$

GENERO $\left\{ \begin{array}{l} \mathbb{N} \text{ è il più piccolo insieme generato} \\ \text{dall'assioma e dalla regola!} \end{array} \right.$

$$\mathbb{N} = \{0, S(0), S(S(0)), S(S(S(0))), \ldots\}$$

Esempio 2) $\mathbb{P} = \{0, 2, 4, 6, 8, \ldots\}$

- Non possiamo "memorizzare" $\mathbb{P}$
- Possiamo trovare una rappresentazione finita implicita:

```
fun pari (x: integer): boolean
    pari := (x mod 2 = 0)
```

RICONOSCO SE $x \in \mathbb{P}$ $\left\{ \begin{array}{l} \text{Il programma è una rappresentazione finita} \\ \text{implicita della funzione (infinita) che calcola} \end{array} \right.$

PER I LING. ESISTONO 2 TECNICHE

1) GENERATIVO / SINTETICO:

Linguaggio = insieme delle stringhe generate da una struttura finita detta GRAMMATICA (a struttura di frase)

2) RICONOSCITIVO / ANALITICO:

Linguaggio = insieme delle stringhe riconosciute da una struttura finita detta AUTOMA

OSS: NON TUTTI I LINGUAGGI POSSONO ESSERE GENERATI DA GRAMMATICHE O RICONOSCIUTI DA AUTOMI !!

- A^* è equipotente a $\mathbb{N}$
- $L \subseteq A^*$ è contabile (finito o infinito)
- $\mathcal{P}(A^*)$ è l'insieme di tutti i ling. su alfabeto A, ma $\mathcal{P}(A^*)$ è equipotente a $\mathbb{R}$, l'insieme dei numeri reali!
- Un "formalismo" è un linguaggio su un alfabeto $\Rightarrow$ i suoi "programmi" (grammatiche / automi) sono numerabili!

Se di grammatiche su alfabeto A ce ne sono tante quante $\mathbb{N}$, e se i linguaggi su A ce ne sono tanti quanti $\mathbb{R}$, allora tantissimi linguaggi non possono essere rappresentati finitamente!

A^*	w_1	w_2	w_3	w_4 ...
$L(G_1)$	X		X	
$L(G_2)$			X	X
$L(G_3)$			X	
$L(G_4)$	X			X
⋮				

ARGOMENTO DIAGONALE DI CANTOR

$$L(G_1) = \{w_1, w_3, \dots\}$$
$$L(G_2) = \{w_3, w_4, \dots\}$$
$$L(G_3) = \{w_3, \dots\}$$

$$\overline{D} = \{w_i \mid w_i \notin L(G_i)\} \qquad \overline{D} = \{w_2, \dots\}$$

$\overline{D}$ è un linguaggio, ma non esiste alcuna G_k tale che $\overline{D} = L(G_k)$ perché:

- se $w_k \in L(G_k)$, allora $w_k \notin \overline{D}$
- se $w_k \notin L(G_k)$, allora $w_k \in \overline{D}$

$\Rightarrow$ $\overline{D}$ non compare nell'elenco dei ling. rappresentabili con una grammatica