Cardinalità transfinita

Dato un insieme finito A di cardinalità n, l'insieme $\mathcal{P}(A)$ ha cardinalità 2^n.

Per analogia, dato un insieme A numerabile, e quindi di cardinalità $\aleph_0$, si dice che l'insieme $\mathcal{P}(A)$ ha cardinalità $2^{\aleph_0}$.

Gli insiemi aventi cardinalità $2^{\aleph_0}$ vengono detti *insiemi continui*.

Teorema 40 *L'insieme $\mathbb{R}$ dei reali è un insieme continuo.*

insiemi	cardinalità
$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{N}^n, \mathbb{Z}^n, \mathbb{Q}^n$	$\aleph_0$
$\mathcal{P}(\mathbb{N}), \mathbb{R}, \mathbb{R}^n$	$2^{\aleph_0}$

Caratteristiche dei linguaggi

Definizione 41 *Un insieme finito non vuoto* Σ *di simboli (detti* caratteri*) prende il nome di* alfabeto*.*

Definizione 42 *Una sequenza finita di elementi di un alfabeto prende il nome di* parola *o* stringa*. La parola vuota si denota con* ε*. Con* Σ^* *si denota l'insieme di tutte le parole ottenute da* Σ*, compresa la parola vuota.*

Definizione 43 *La* concatenazione *di due parole* x *e* y *si ottiene giustapponendo* x *e* y *(e si denota con* xy *o* $x \circ y$*).*
Si ha che $x \circ \varepsilon = \varepsilon \circ x = x$ *e, in generale,* $x \circ y \neq y \circ x$*.*
x^h *denota la concatenazione di* x *con se stessa* h *volte.*

Definizione 44 *La* lunghezza *di una parola* ($|x|$) *é il numero di caratteri che la costituiscono.* $|\varepsilon| = 0$*.*

Definizione 45 *Dato un alfabeto* Σ, *si definisce* linguaggio *un qualsiasi sottoinsieme di* Σ^*. *Il linguaggio che non contiene alcuna stringa si chiama* linguaggio vuoto *(Λ).*

Definizione 46 *Data una parola* x, *chiamiamo inversa di* x *la stringa* $\tilde{x}$ *tale che:*

$$\tilde{x} = \begin{cases} x & \text{se } x = \varepsilon \text{ o } x = a, \\ a\tilde{y} & \text{se x=ya.} \end{cases}$$

per ogni $a \in \Sigma$.

L'insieme delle stringhe palindrome *é* $\{x | x = \tilde{x}\}$.

Operazioni su linguaggi

Siano L_1 e L_2 due linguaggi definiti sullo stesso alfabeto Σ.

intersezione: $L_1 \cap L_2 = \{x \in \Sigma^* | x \in L_1 \wedge x \in L_2\}$

unione: $L_1 \cup L_2 = \{x \in \Sigma^* | x \in L_1 \vee x \in L_2\}$

complemento: $\bar{L_1} = \{x \in \Sigma^* | x \notin L_1\}$

concatenazione o prodotto:
$L_1 \circ L_2 = \{x \in \Sigma^* | \exists y_1 \in L_1 \wedge \exists y_2 \in L_2,\ x = y_1 \circ y_2\}$

potenza: $L^0 = \{\varepsilon\}$; $L^h = L \circ L^{h-1}$

chiusura: $L^+ = \bigcup_{h=1}^{\infty} L^h$

chiusura riflessiva: $L^* = \bigcup_{h=0}^{\infty} L^h$

Espressioni regolari: consentono di rappresentare linguaggi mediante una interpretazione dei simboli che le compongono.

Definizione 47 *Dato un alfabeto* Σ *e l'insieme dei simboli* $\{+,^*,(,),\cdot,\varnothing\}$ *si definisce* espressione regolare *sull'alfabeto* Σ *una stringa* $r \in (\Sigma \cup \{+,^*,(,),\cdot,\varnothing\})^+$ *tale che valga una delle seguenti condizioni:*

1. $r = \varnothing$*;*

2. $r \in \Sigma$*;*

3. $r = (s + t)$*, oppure* $r = (s \cdot t)$*, oppure* $r = s^*$*, con* s *e* t *espressioni regolari sull' alfabeto* Σ*.*

espr. regolari	linguaggi
$\emptyset$	Λ
a	$\{a\}$
$(s+t)$	$\mathcal{L}(s) \cup \mathcal{L}(t)$
$(s \cdot t)$	$\mathcal{L}(s) \circ \mathcal{L}(t)$
s^*	$(\mathcal{L}(s))^*$

Convenzioni: s, t due espressioni regolari. $(s \cdot t)$ si scrive brevemente (st). Assumendo le seguenti precedenze $*, \cdot, +$ e tenendo conto dell'associatività si possono spesso eliminare alcune parentesi. Ad esempio, $(a+(b\cdot(c\cdot d)))$ definita su Σ si scrive $a+bcd$.

Esempio 48 *Scrivere l'espressione regolare che rappresenta il linguaggio* $\{x|x \in \{a,b\}^+, x$ *termina con* $a\}$.

Esempio 49 *Scrivere l'espressione regolare che, sull'alfabeto* $\{a,b\}$, *definisce l'insieme delle stringhe il cui terzultimo carattere é una* b .

Esempio 50 *Determinare il linguaggio definito dall'espressione regolare* $a^*((aa)^*b + (bb)^*a)b^*$.

Cardinalitá dei linguaggi

Definizione 51 *Sia* $\Sigma = \{a_1, \ldots, a_n\}$ *un alfabeto. Si definisce* ordinamento lessicografico *delle stringhe in* Σ^* *l'ordinamento* $<$ *ottenuto stabilendo un ordinamento fra i caratteri di* Σ *(*$a_1 < \ldots < a_n$*) e tale che* $x < y$ *sse una delle due condizioni seguenti é verificata:*
1. $|x| < |y|$*;*
2. $|x| = |y|$ *ed esiste* $z \in \Sigma^*$ *tale che* $x = za_iu$ *e* $y = za_jv$*, con* $u, v \in \Sigma^*$ *e* $i < j$*.*

ordinamento lessicografico $\Longrightarrow |\Sigma^*| = \aleph_0$;

l'insieme di tutti i linguaggi su Σ é equinumeroso a $\mathcal{P}(\Sigma)$, che ha cardinalitá $2^{\aleph_0}$, quindi non é numerabile.

Problema: riconoscere se una stringa appartiene ad un linguaggio (compilatore).

Se Σ_P è l'alfabeto del Pascal $\Longrightarrow$ un programma é una stringa di Σ_P^* accettata dal compilatore.

Qual è la cardinalitá dell'insieme dei programmi? $\aleph_0$, perché possiamo enumerarli lessicograficamente.

Dato un linguaggio $L \subseteq \Sigma^*$, esiste un programma Pascal che, data in input una stringa $x \in \Sigma^*$, ne decida l'appartenenza a L?

I programmi Pascal sono contabili, mentre i linguaggi hanno cardinalitá del continuo. Esistono piú linguaggi *da riconoscere* che programmi che *riconoscono*.
Quindi esistono linguaggi per i quali non esiste alcun programma Pascal di riconoscimento.

Notazione asintotica

Siano f, g due funzioni $\mathbb{N} \longmapsto \mathbb{N}$

$O(f(n)) = \{g | (\exists c > 0)(\exists n_0 > 0)(\forall n \geq n_0)(g(n) \leq cf(n))\}$

$\Omega(f(n)) = \{g | (\exists c > 0)(\exists n_0 > 0)(\forall n \geq n_0)(g(n) \geq cf(n))\}$

$\Theta(f(n)) = \{g | (\exists c_1 > 0)(\exists c_2 \geq c_1)(\exists n_0 > 0)(\forall n \geq n_0)(c_1 f(n) \leq g(n) \leq c_2 f(n))\}$

$g(n) = o(f(n))$ indica che $\lim_{n \to \infty} \frac{g(n)}{f(n)} = 0$