

# Concetti fondamentali

---

a.a. 2021-2022

Corso di Fondamenti di Informatica - 1 modulo

Corso di Laurea in Informatica

Università di Roma "Tor Vergata"

Prof. Giorgio Gambosi



Data una proposizione  $P(n)$  definita per un generico numero naturale  $n$ , si ha che essa è vera per tutti i naturali se

- $P(0)$  è vera (**passo base** dell'induzione);
- per ogni naturale  $k$ ,  $P(k)$  vera (ipotesi induttiva) implica  $P(k + 1)$  vera (**passo induttivo**).

$$(P(0) \wedge \forall k' (P(k') \implies P(k' + 1))) \implies \forall n P(n)$$

# Esempio di induzione matematica

Dimostrare che  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$

- Passo base:  $\sum_{i=0}^0 i = \frac{0(0+1)}{2} = 0$
- Passo induttivo:

$$\begin{aligned}\sum_{i=0}^{k+1} i &= \sum_{i=0}^k i + (k+1) = \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k^2 + 3k + 2}{2} = \frac{(k+1)(k+2)}{2}\end{aligned}$$

# Esempio di induzione matematica

Dimostrare che

$$\sum_{i=0}^{n-1} 2^i = 2^n - 1$$

per  $n \geq 1$

# Induzione completa

Data una proposizione  $P(n)$  definita per un generico numero naturale  $n \geq n_0$  si ha che essa è vera per tutti gli  $n \geq n_0$  se:

- $P(n_0)$  è vera (passo base dell'induzione);
- per ogni naturale  $k \geq n_0$ ,  $P(i)$  vera per ogni  $i$ ,  $n_0 \leq i \leq k$  (ipotesi induttiva) implica  $P(k+1)$  vera (passo induttivo).

$$(P(n_0) \wedge \forall k' (P(n_0) \wedge \dots \wedge P(k') \implies P(k'+1))) \implies \forall n P(n)$$

# Insiemi infiniti

Due insiemi  $A$  e  $B$  si dicono **equinumerosi** se esiste una biiezione tra di essi

Dato un insieme finito  $A$ , la sua cardinalità  $|A|$  è definita come:

$$|A| = \begin{cases} 0 & \text{se } A = \emptyset \\ n & \text{se } A \text{ è equinumeroso a } \{0, 1, \dots, n-1\}, \text{ con } n \geq 1. \end{cases}$$

# Insiemi infiniti

- Un insieme si dice **numerabile** se esso è equinumeroso a  $\mathbb{N}$ .
- Un insieme si dice **contabile** se esso è finito o numerabile.
- Per indicare la cardinalità degli insiemi infiniti equinumerosi ad  $\mathbb{N}$  si utilizza il simbolo  $\aleph_0$
- Se un insieme  $A$  è equinumeroso a un insieme  $B$ , con  $B \subseteq C$ , dove  $C$  è un insieme contabile, allora anche  $A$  è contabile.

L'insieme  $\mathbb{Z}$  degli interi relativi risulta essere numerabile (cioè  $|\mathbb{Z}| = \aleph_0$ ) poiché i suoi elementi possono essere posti in corrispondenza biunivoca con  $\mathbb{N}$  tramite la biiezione  $f : \mathbb{Z} \mapsto \mathbb{N}$  definita nel seguente modo:

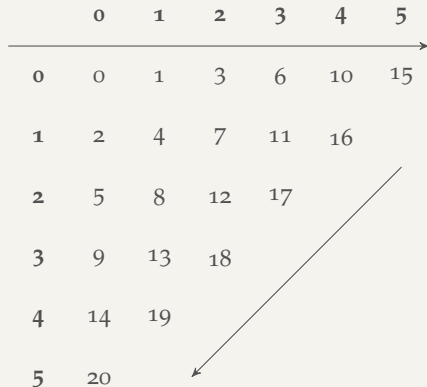
$$f(i) = \begin{cases} -2i & \text{se } i \leq 0 \\ 2i - 1 & \text{se } i > 0. \end{cases}$$



L'insieme  $\mathbb{N}^2$  delle coppie di naturali risulta essere numerabile. La corrispondenza biunivoca può essere stabilita con la seguente biiezione, frequentemente chiamata **funzione coppia di Cantor**

$$p(i, j) = \frac{(i + j)(i + j + 1)}{2} + i.$$

# Coppia di Cantor



A diagram illustrating the Cantor pairing function. It features a 6x6 grid of numbers. The columns are indexed 0 to 5 at the top, and the rows are indexed 0 to 5 on the left. A horizontal arrow points to the right above the grid. A diagonal arrow points from the top-right towards the bottom-left, passing through the cell (5, 2) which contains the number 20.

	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					

I numeri razionali corrispondono alle classi d'equivalenza della relazione binaria  $R$  definita sull'insieme  $\mathbb{Z} \times \mathbb{Z}^+$ :

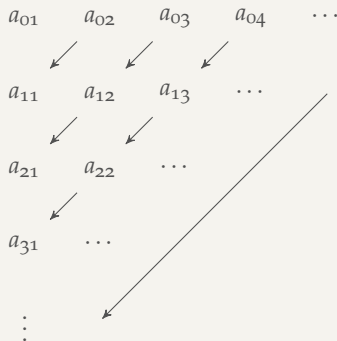
$$R(\langle a, b \rangle, \langle c, d \rangle) \text{ se e solo se } ad = bc.$$

L'insieme  $\mathbb{Q}$  è dunque equinumeroso all'insieme  $\mathbb{Z} \times (\mathbb{Z}^+)/_R$ .

D'altronde poiché  $\mathbb{Z}$  è contabile, anche  $\mathbb{Z}^2$  lo è e così anche l'insieme  $\mathbb{Z} \times (\mathbb{Z}^+)/_R$  che è equinumeroso ad un sottoinsieme proprio di  $\mathbb{Z}^2$ .  
Quindi  $\mathbb{Q}$  è contabile.

# Insiemi infiniti

L'unione di una quantità contabile di insiemi contabili è ancora un insieme contabile. L'enumerazione può essere effettuata applicando ancora il metodo di Cantor, come si può vedere in figura, dove si suppone che la riga  $i$ -esima contenga gli elementi dell' $i$ -esimo insieme.



L'insieme  $\mathbb{R}$  dei reali non è numerabile.

L'insieme aperto  $(0, 1)$  e l'insieme  $\mathbb{R}$  sono equinumerosi (una possibile biiezione è  $1/(2^x + 1)$ , con dominio  $\mathbb{R}$  e codominio  $(0, 1)$ ).

Basta dunque mostrare che l'insieme dei reali in  $(0, 1)$  non è numerabile. A tal fine, consideriamo l'insieme delle sequenze infinite di cifre decimali che i reali in  $(0, 1)$  e mostriamo che tale insieme non è numerabile.

Si supponga per assurdo di aver trovato una qualsiasi corrispondenza tra i naturali e le sequenze: questa corrispondenza definirebbe una enumerazione  $\Phi = \langle \phi_0, \phi_1, \dots \rangle$  delle sequenze.

Introduciamo ora la sequenza  $\phi$  avente come  $i$ -esima cifra, per  $i = 0, 1, 2, \dots$ , il valore ottenuto sommando  $1 \pmod{10}$  alla  $i$ -esima cifra di  $\phi_i$ .

$$\phi[i] = (\phi_i[i] + 1) \bmod 10$$

# Insiemi infiniti

	0	1	2	3	...
$\phi_0$	0	0	...	...	...
$\phi_1$	0	1	0	1	...
$\phi_2$	2	1	3	3	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\phi_k$	...	...	...	...	...

# Insiemi infiniti

La sequenza  $\phi$  viene a costituire elemento diagonale dell'enumerazione  $\phi_0, \phi_1, \dots$  in quanto differisce da ogni altra sequenza  $\phi_i$  nella posizione  $i$ .

Quindi, dopo aver supposto per assurdo di poter enumerare tutte le rappresentazioni decimali di reali nell'intervallo  $(0, 1)$ , è stato possibile costruire per diagonalizzazione un'ulteriore rappresentazione che, seppure relativa ad un reale in  $(0, 1)$ , non appartiene all'enumerazione, il che contrasta con l'ipotesi che l'insieme delle rappresentazioni dei reali sia numerabile.

La non numerabilità dei reali in  $(0, 1)$  deriva da quanto detto ed osservando inoltre che ogni numero reale ha al più due rappresentazioni distinte (ad esempio,  $0.01000\dots$  e  $0.00999\dots$ )



L'insieme delle parti di  $\mathbb{N}$ ,  $\mathcal{P}(\mathbb{N})$ , non è numerabile.

Supponiamo per assurdo che  $\mathcal{P}(\mathbb{N})$  sia numerabile e sia  $P_0, P_1, \dots$  una sua enumerazione. A ciascun  $P_i$ , con  $i = 0, 1, 2, \dots$ , associamo una sequenza  $b_{i0}, b_{i1}, b_{i2}, \dots$ , dove

$$b_{ij} = \begin{cases} 0 & \text{se } j \notin P_i \\ 1 & \text{se } j \in P_i \end{cases}$$

L'insieme diagonale  $P = p_0, p_1, \dots$  è definito come  $p_i = 1 - b_{ii}$  e differisce da ciascuno degli insiemi  $P_i$  poiché, per costruzione,  $i \in P \iff i \notin P_i$ . Avendo dunque supposto che sia possibile enumerare gli elementi di  $\mathcal{P}(\mathbb{N})$ , si è riusciti a costruire un insieme  $P \in \mathcal{P}(\mathbb{N})$  che non fa parte della enumerazione, il che falsifica tale ipotesi.

# Insiemi infiniti

L'insieme delle funzioni caratteristiche  $\{f \mid f : \mathbb{N} \mapsto \{0, 1\}\}$  non è numerabile.

Si supponga per assurdo di avere una corrispondenza tra l'insieme delle funzioni caratteristiche e i naturali.

	0	1	2	3	4	...	$j$	...
$f_0$	0	0	0	0	0	...	$f_0(j)$	...
$f_1$	0	1	0	1	0	...	$f_1(j)$	...
$f_2$	0	1	1	0	1	...	$f_2(j)$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$
$f_i$	$f_i(0)$	$f_i(1)$	$f_i(2)$	$f_i(3)$	$f_i(4)$	...	$f_i(j)$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$

A partire dalle funzioni di questa enumerazione, si può costruire una nuova funzione, definita come:

$$\hat{f}(i) = 1 - f_i(i).$$

$\hat{f}(i)$  assume solo valori in  $\{0, 1\}$  ed è diversa da tutte quelle enumerate per almeno un valore dell'argomento, il che dimostra l'asserto.

# Insiemi infiniti

Dato un insieme  $A$  numerabile, e quindi di cardinalità  $\aleph_0$ , si dice che l'insieme  $\mathcal{P}(A)$  ha cardinalità  $2^{\aleph_0}$  (o, con un leggero abuso di notazione,  $2^{\mathbb{N}}$ ).

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

L'insieme delle funzioni da interi a interi è continuo

$$|\{f \mid f : \mathbb{N} \mapsto \mathbb{N}\}| = 2^{\aleph_0}$$

L'insieme delle funzioni da reali a reali ha cardinalità

$$|\{f \mid f : \mathbb{R} \mapsto \mathbb{R}\}| = 2^{2^{\aleph_0}}$$

Un insieme finito non vuoto  $\Sigma$  di simboli (detti **caratteri**) prende il nome di **alfabeto**.

Dato un alfabeto  $\Sigma$ , denotiamo come  $\langle \Sigma^*, \circ, \varepsilon \rangle$  il **monoide libero definito su  $\Sigma$** . Gli elementi di  $\Sigma^*$  vengono detti **parole** o **stringhe**.

L'elemento  $\varepsilon$  viene detto **parola vuota**. L'operazione  $\circ : \Sigma^* \times \Sigma^* \mapsto \Sigma^*$  definita sul monoide è chiamata **concatenazione** e consiste nel giustapporre due parole di  $\Sigma^*$ :

$$x_{i_1} \dots x_{i_n} \circ y_{j_1} \dots y_{j_m} = x_{i_1} \dots x_{i_n} y_{j_1} \dots y_{j_m},$$

con  $x_{i_1}, \dots, x_{i_n}, y_{j_1}, \dots, y_{j_m} \in \Sigma$ .

Dato un alfabeto  $\Sigma$  ed il monoide sintattico definito su di esso vale la proprietà:

$$\forall x \quad x \circ \varepsilon = \varepsilon \circ x = x.$$

La concatenazione di due stringhe  $x$  e  $y$  è frequentemente indicata omettendo il simbolo  $\circ$ , cioè scrivendo  $xy$  anziché  $x \circ y$ .

Con la notazione  $|x|$  indichiamo la **lunghezza** di una parola  $x$ , ovvero il numero di caratteri che la costituiscono. Chiaramente  $|\varepsilon| = 0$ . Si osservi inoltre che la concatenazione non gode della proprietà commutativa e quindi in generale:

$$x \circ y \neq y \circ x.$$

Un caso particolare di concatenazione è quello in cui la stringa viene concatenata con sé stessa: con  $x^h$  si denota la concatenazione di  $x$  con sé stessa iterata  $h$  volte. Per convenzione con  $x^0$  si intende la stringa vuota.

Dato un alfabeto  $\Sigma$ , si definisce **linguaggio** un qualsivoglia sottoinsieme di  $\Sigma^*$ . Si noti che poiché  $\Sigma \subseteq \Sigma^*$ , un alfabeto è a sua volta un linguaggio.

Si chiama **linguaggio vuoto**, e lo si indica con  $\Lambda$ , il linguaggio che non contiene stringa alcuna. Si noti che  $\Lambda \neq \{\varepsilon\}$ .

L'**intersezione** di due linguaggi  $L_1$  e  $L_2$  è il linguaggio  $L_1 \cap L_2$  costituito dalle parole di  $L_1$  e di  $L_2$ , cioè  $L_1 \cap L_2 = \{x \in \Sigma^* \mid x \in L_1 \wedge x \in L_2\}$ .

L'**unione** di due linguaggi  $L_1$  e  $L_2$  è il linguaggio  $L_1 \cup L_2$  costituito dalle parole appartenenti ad almeno uno fra  $L_1$  ed  $L_2$ , cioè  $L_1 \cup L_2 = \{x \in \Sigma^* \mid x \in L_1 \vee x \in L_2\}$ . Si noti che  $L_1 \cap \Lambda = \Lambda$  e  $L_1 \cup \Lambda = L_1$ .

Il **complemento** di un linguaggio  $L_1$  è il linguaggio  $\overline{L_1} = \Sigma^* - L_1$  costituito dalle parole appartenenti a  $\Sigma^*$  ma non ad  $L_1$ , cioè  $\overline{L_1} = \{x \in \Sigma^* \mid x \notin L_1\}$ .



La **concatenazione** (o **prodotto**) di due linguaggi  $L_1$  e  $L_2$  è il linguaggio  $L_1 \circ L_2$  delle parole costituite dalla concatenazione di una stringa di  $L_1$  e di una stringa di  $L_2$ , cioè

$$L_1 \circ L_2 = \{x \in \Sigma^* \mid \exists y_1 \in L_1 \exists y_2 \in L_2 (x = y_1 \circ y_2)\}.$$

Si noti che  $L \circ \{\varepsilon\} = \{\varepsilon\} \circ L = L$ , e che  $L \circ \Lambda = \Lambda \circ L = \Lambda$ .

La **potenza**  $L^h$  di un linguaggio è definita come

$$L^h = L \circ L^{h-1}, h \geq 1$$

con la convenzione secondo cui  $L^0 = \{\varepsilon\}$ . Si noti che, in base alla suddetta convenzione,  $\Lambda^0 = \{\varepsilon\}$ .

Il linguaggio  $L^*$  definito da

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

prende il nome di **chiusura riflessiva del linguaggio**  $L$  rispetto all'operazione di concatenazione, mentre l'operatore “ $*$ ” prende il nome di **iterazione** o **stella di Kleene**. Si noti che, dato un qualunque linguaggio  $L$ ,  $\varepsilon \in L^*$ , e che  $\Lambda^* = \{\varepsilon\}$ .

Si indica con  $L^+$  la **chiusura (non riflessiva)** definita da

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

Risulta ovviamente  $L^* = L^+ \cup \{\varepsilon\}$ .

# Espressioni regolari

Dato un alfabeto  $\Sigma$  e dato l'insieme di simboli

$$\{+, *, (, ), \cdot, \emptyset\}$$

si definisce **espressione regolare** sull'alfabeto  $\Sigma$  una stringa

$$r \in (\Sigma \cup \{+, *, (, ), \cdot, \emptyset\})^+$$

tale che valga una delle seguenti condizioni:

1.  $r = \emptyset$
2.  $r \in \Sigma$
3.  $r = (s + t)$ , oppure  $r = (s \cdot t)$ , oppure  $r = s^*$ , dove  $s$  e  $t$  sono espressioni regolari sull'alfabeto  $\Sigma$ .

# Espressioni regolari

Le espressioni regolari consentono di rappresentare linguaggi mediante una opportuna interpretazione dei simboli che le compongono. Nella tabella si mostra la corrispondenza tra un'espressione regolare  $r$  e il linguaggio  $\mathcal{L}(r)$  che essa rappresenta.

Espr. regolari	Linguaggi
$\emptyset$	$\Lambda$
$a$	$\{a\}$
$(s + t)$	$\mathcal{L}(s) \cup \mathcal{L}(t)$
$(s \cdot t)$	$\mathcal{L}(s) \circ \mathcal{L}(t)$
$s^*$	$(\mathcal{L}(s))^*$

L'espressione regolare  $(a + b)^*a$  rappresenta il linguaggio

$$\begin{aligned}\mathcal{L}((a + b)^*a) &= \mathcal{L}((a + b)^*) \circ \mathcal{L}(a) \\ &= (\mathcal{L}(a + b))^* \circ \mathcal{L}(a) \\ &= (\mathcal{L}(a) \cup \mathcal{L}(b))^* \circ \{a\} \\ &= (\{a\} \cup \{b\})^* \circ \{a\} \\ &= \{a, b\}^* \circ \{a\} \\ &= \{x \mid x \in \{a, b\}^+, x \text{ termina con } a\}.\end{aligned}$$

Posto  $c = \{0, 1, \dots, 9\}$ , l'espressione regolare che rappresenta i numeri reali nella forma  $c_{i_1}c_{i_2} \dots c_{i_n}.c_{f_1}c_{f_2} \dots c_{f_m}$  (dove  $c_{ij}$  rappresenta la  $j$ -esima cifra prima del punto decimale, e  $c_{fk}$  la  $k$ -esima cifra dopo il punto decimale) è

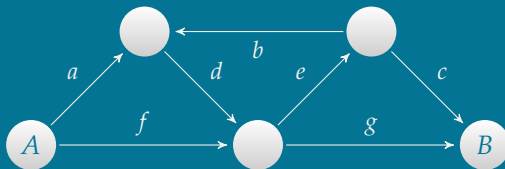
$$((1 + 2 + \dots + 9)(0 + 1 + \dots + 9)^* + 0).(0 + 1 + \dots + 9)^+.$$

Determinare l'espressione regolare che, sull'alfabeto  $\{a, b\}$ , definisce l'insieme delle stringhe il cui terzultimo carattere è una  $b$ .

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

# Espressioni regolari

Sia data la mappa stradale (con tratti stradali a senso unico e contrassegnati da caratteri dell'alfabeto) schematicamente indicata in figura. Fornire un'espressione regolare che definisca tutti i percorsi, anche passanti più volte per uno stesso nodo, tra  $A$  e  $B$ .





L'insieme dei linguaggi è in stretto rapporto con quello dei **problemi di decisione**.

## Problema di decisione

Un problema di decisione è definito su un insieme di possibili **istanze** e associa ad ognuna di esse un valore Vero/Falso.

L'insieme delle istanze è partizionato in istanze **positive** e **negative**: il problema è, per ogni istanza, riconoscere se è una istanza positiva

# Esempi di problemi di decisione

1. una data sequenza di interi è ordinata crescente?
  2. dati due capoluoghi di provincia italiani  $A$  e  $B$ , è possibile andare da uno all'altro percorrendo meno di  $x$  km?
  3. è possibile viaggiare tra tutti i capoluoghi di provincia italiani senza passare due volte per nessuno di essi?
  4. dato un programma contenente una determinata funzione e un input del programma stesso, la funzione viene eseguita?
- 
1. risolubile da un algoritmo che opera in tempo lineare
  2. risolubile da un algoritmo che opera in tempo polinomiale
  3. risolubile da un algoritmo in tempo esponenziale (polinomiale?)
  4. non risolubile da nessun algoritmo

# Problemi di decisione

I problemi di decisione rappresentano la tipologia più “semplice” di problemi:

- problemi di ricerca: restituzione di una soluzione, **se esiste** (esempio, un percorso da  $A$  a  $B$  di al più  $x$  km)
- problemi di enumerazione: restituzione di tutte le soluzioni, se esistono (esempio, tutti i percorsi da  $A$  a  $B$  di al più  $x$  km)
- problemi di ottimizzazione: restituzione della “migliore” soluzione possibile (esempio, il percorso più breve da  $A$  a  $B$ )

Relazione tra linguaggi e problemi di decisione:

- Dato un linguaggio  $L$  e una stringa  $x$ , determinare se  $x \in L$  è un problema di decisione
- Istanze: tutte le possibili stringhe;  $L$ : istanze positive

Relazione tra problemi di decisione e linguaggi:

- Dato un problema di decisione  $\mathcal{P}$ , l'insieme delle sue istanze è codificato per mezzo di uno **schema di codifica**: ad ogni istanza corrisponde una stringa
- Una stringa può corrispondere a una istanza positiva, una istanza negativa o nessuna istanza
- $L$  insieme delle stringhe corrispondenti a codifiche di istanze positive
- In generale, si assume che sia possibile distinguere facilmente (efficientemente) tra le stringhe che rappresentano istanze e le altre

# Rappresentazione di problemi o linguaggi

- Definizione data: un problema di decisione è caratterizzato dall'insieme delle istanze positive, rispetto all'insieme di tutte le istanze.
- Insiemi in generale infiniti: non è una definizione praticabile
- In genere, definizione di un problema per mezzo di una sua descrizione **finita**
- Caso particolare di descrizione: un algoritmo che **decide** il problema (restituisce Vero per istanze positive e Falso per istanze negative)

- Tutti i problemi possono essere descritti in modo finito?
- Equivalente a dire: per tutti i problemi di decisione esistono algoritmi che li risolvono?
- La risposta è **NO**

# Quanti sono i problemi di decisione?

Consideriamo i linguaggi definiti su un alfabeto dato, ad esempio  $\{0, 1\}$

- Un linguaggio è un insieme (infinito, in generale) di stringhe su  $\{0, 1\}$ , e quindi corrisponde ad una sequenza di 0 (Falso) o 1 (Vero) sulla sequenza di tutte le stringhe ordinate, ad esempio, in modo lessicografico
- Applicando la diagonalizzazione vista per i numeri reali, ne risulta che l'insieme dei linguaggi è non numerabile



# Quante sono le descrizioni/algoritmi?

Consideriamo le descrizioni utilizzando un alfabeto dato, ad esempio ancora  $\{0, 1\}$

- Una descrizione/algoritmo è una stringa su  $\{0, 1\}$
- Dato un qualunque alfabeto finito, l'insieme delle stringhe corrispondenti è numerabile

Conseguenza: ci sono più problemi di decisione (linguaggi) che loro descrizioni finite (algoritmi). Quindi, esistono problemi/linguaggi non descrivibili in modo finiti e non decidibili mediante algoritmi.

# Considerazioni più “fini”

Quanto detto vale nel caso di descrizioni/algoritmi più generali possibili. Che succede se consideriamo delle restrizioni nelle modalità di descrizione?

Perché dovremmo porci la questione?

Descrizioni più limitate possono corrispondere ad algoritmi di decisione più efficienti