Lezione 17

Arresto ricorsivamente numerabile

Un esempio di insieme che non è ricorsivo, ma è ricorsivamente numerabile è identificato dal problema dell'arresto ristretto, che era così definito:

- nome: $\text{AR}_{\overset{\S}{P}}$
- istanza: $x \in \mathbb{N}$
- domanda: $\varphi_{\overset{\S}{P}}(x) = \varphi_x(x) \downarrow$?

L'insieme $A = \{x : \varphi_x(x) \downarrow\}$ non è ricorsivo, altrimenti sarebbe decidibile. Tuttavia, è **ricorsivamente numerabile**, infatti il seguente programma

$$\begin{aligned} P \equiv\ & \text{input}(x); \\ & U(x, x); \\ & \text{output}(1) \end{aligned}$$

sfrutta il fatto che se $x \in A$, allora $\varphi_x(x) \downarrow$ (quindi l'interprete universale termina) e il programma P restituisce 1, altrimenti non termina. Di conseguenza:

$$\varphi_P(x) = \begin{cases} 1 & \text{se } \varphi_U(x, x) = \varphi_x(x) \downarrow \\ \perp & \text{altrimenti} \end{cases}.$$

Dato che $A = \text{Dom}_{\varphi_P \in \mathcal{P}}$, posso applicare la seconda caratterizzazione per dimostrare che l'insieme dell'arresto è un insieme ricorsivamente numerabile.

Alternativamente, possiamo dire che $A = \{x : \varphi_x(x) \downarrow\} = \left\{x : \exists y : (x, y) \in R_{\overset{\S}{P}}\right\}$ con relazione ricorsiva $R_{\overset{\S}{P}} = \left\{(x, y) : \overset{\S}{P} \text{ su input } x \text{ termina entro } y \text{ passi}\right\}$ e qui possiamo sfruttare la terza caratterizzazione degli insiemi ricorsivamente numerabili.

Ricorsivi vs Ricorsivamente numerabili

Teorema $A \subseteq \mathbb{N}$ ricorsivo $\Rightarrow A$ ricorsivamente numerabile.

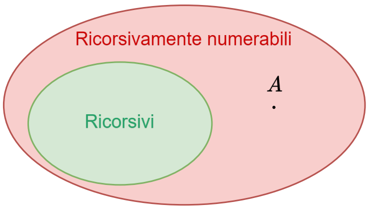
Dimostrazione A ricorsivo implica che esiste un programma che è in grado di riconoscerlo.

$$x \in \mathbb{N} \rightsquigarrow P(x) = \begin{cases} 1 \text{ se } x \in A \\ 0 \text{ se } x \notin A \end{cases}$$

dove P è di questo tipo:

$$\begin{aligned} P \equiv\ & \text{input}(x); \\ & \text{if}\big(P_{A(x)} = 1\big) \\ & \qquad \text{output}(1); \\ & \text{else} \\ & \qquad \text{while}(1 > 0); \end{aligned}$$

Di conseguenza, A è il dominio di una funzione ricorsiva parziale $\Rightarrow A$ è ricorsivamente numerabile per la seconda caratterizzazione. $\square$



$A = \{x \in \mathbb{N} : \varphi_x(x) \downarrow\}$ non è ricorsivo, ma è ricorsivamente numerabile $\Rightarrow$

$$\text{Ricorsivi} \subset \text{Ric. numerabili}$$

Ma esistono insiemi non ricorsivamente numerabili?

Chiusura degli insiemi ricorsivi

Cerchiamo di sfruttare l'operazione di complemento degli insiemi sui ricorsivamente numerabili per vedere di che natura è l'insieme.

$$A^C = \{x \in \mathbb{N} : \varphi_x(x) \uparrow\}??$$

Teorema La classe degli insiemi ricorsivi è un'Algebra di Boole (i.e. chiusa per complemento, intersezione e unione).

Dimostrazione Siano A, B ricorsivi. Allora esistono dei programmi P_A, P_B che li riconoscono (o equivalentemente esistono $\chi_A, \chi_B \in \mathcal{T}$).

È facile dimostrare che le operazioni di unione, intersezione e complemento sono facilmente implementabili da programmi che terminano sempre. Di conseguenza $A \cup B$, $A \cap B$, A^C sono ricorsive.

Ecco tre esempi di programmi che calcolano le tre funzioni insiemistiche:

1. Complemento:

$$\begin{aligned} P_{A^C} \equiv\ & \text{input}(x) \\ & \text{output}(1 \dot{-} P_A(x)) \end{aligned}$$

1. Intersezione:

$$\begin{aligned} P_{A \cap B} \equiv\ & \text{input}(x) \\ & \text{output}(\text{MIN}(P_A(x), P_B(x))) \end{aligned}$$

1. Unione:

$$\begin{aligned} P_{A \cup B} \equiv\ & \text{input}(x) \\ & \text{output}(\max(P_A(x), P_B(x))) \end{aligned}$$

Allo stesso modo possiamo trovare le funzioni caratteristiche delle tre operazioni:

1. $\chi_{A^C}(x) = 1 \dot{-} \chi_A(x)$
2. $\chi_{A \cap B} = \chi_A(x) \cdot \chi_B(x)$
3. $\chi_{A \cup B} = 1 \dot{-} (1 \dot{-} \chi_A(x))(1 \dot{-} \chi_B(x))$

Tutte queste funzioni sono ricorsive totali $\Rightarrow$ le funzioni A^C, $A \cap B$, $A \cup B$ sono ricorsive. □

Ora, però, vediamo un risultato molto importante riguardante nello specifico il complemento dell'insieme dell'arresto $A^C = \{x \in \mathbb{N} : \varphi_x(x) \downarrow\}$.

Teorema A^C non è ricorsivo.

Dimostrazione Se A^C fosse ricorsivo, per la proprietà di chiusura dimostrata nel teorema precedente, avremmo $\left(A^C\right)^C = A$ ricorsivo, il che è assurdo. □

Ricapitolando abbiamo:
- $A = \{x : \varphi_x(x) \downarrow\}$ ricorsivamente numerabile, ma non ricorsivo;
- $A^C = \{x : \varphi_x(x) \uparrow\}$ non ricorsivo. *Potrebbe essere ricorsivamente numerabile?*

Teorema (A ricorsivamente numerabile, A^C ricorsivamente numerabile) $\Rightarrow A$ ricorsivo.

Dimostrazione
INFORMALE:

Essendo A, A^C ricorsivamente numerabili, esiste un libro con infinite pagine su ognuna delle quali compare un elemento di A ed esiste un libro analogo per A^C.

Per decidere l'appartenenza ad A, possiamo utilizzare il seguente procedimento:
1. input(x);
2. Apriamo i due libri alla prima pagina;
3. Se x compare nel libro di A, stampa 1,
   Se x compare nel libro di A^C, stampa 0,
   Se x non compare su nessuna delle due pagine, voltiamo la pagina di ogni libro e rieseguiamo 3.

Da notare che questo algoritmo termina sempre dato che x o sta in A o sta in A^C, quindi prima o poi verrà trovato su uno dei due libri.

Dunque, l'algoritmo riconosce $A \Rightarrow A$ è ricorsivo

FORMALE:

Essendo A, A^C ricorsivamente numerabili, esistono $f, g \in \mathcal{T}$ tali che $A = \mathrm{Im}_f$, $A^C = \mathrm{Im}_g$. Sia f implementata dal programma F e g dal programma G. Il seguente programma riconosce A:

$$
\begin{aligned}
P \equiv\ & \mathrm{input}(x) \\
& i := 0; \\
& \mathrm{while}(\mathrm{true}) \\
& \quad \mathrm{if}(F(i) = x)\ \mathrm{output}(1); \\
& \quad \mathrm{if}(G(i) = x)\ \mathrm{output}(0); \\
& \quad i := i + 1;
\end{aligned}
$$

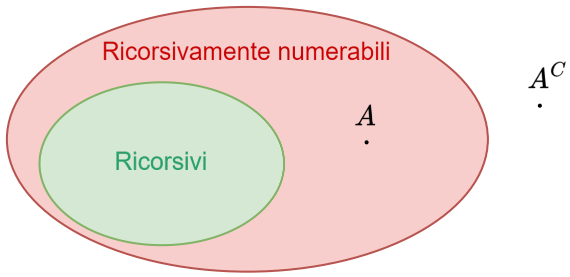
Questo algoritmo termina per ogni input, in quanto $x \in A$ o $x \in A^C$. Possiamo concludere che l'insieme A è ricorsivo. □

Possiamo concludere immediatamente che $A^C = \{x : \varphi_x(x) \uparrow\}$ non può essere ricorsivamente numerabile.

In generale, questo teorema ci fornisce uno strumento molto interessante per studiare le caratteristiche della riconoscibilità di un insieme A:
- se A non è ricorsivo, potrebbe essere ricorsivamente numerabile;
- se non riesco a mostrarlo, provo a studiare A^C;

- se A^C è ricorsivamente numerabile, allora per il teorema possiamo concludere che A non è ricorsivamente numerabile.

Situazione finale



Chiusura degli insiemi ricorsivamente numerabili

Teorema La classe degli insiemi ricorsivamente numerabili è chiusa per unione e intersezione, ma non per complemento.

Dimostrazione Per complemento, abbiamo mostrato che $A = \{x : \varphi_x(x) \downarrow\}$ è ricorsivamente numerabile, mentre $A^C = \{x : \varphi_x(x) \uparrow\}$ non lo è.

Siano A, B ricorsivamente numerabili. Esistono, perciò, $f, g \in \mathcal{T} : A = \mathrm{Im}_f$ e $B = \mathrm{Im}_g$. Sia f implementata da F e g implementata da G. Siano

$$
\begin{aligned}
P' \equiv\ & \mathrm{input}(x); \\
& i := 0; \\
& \mathrm{while}(F(i) \neq x) i++; \\
& i := 0; \\
& \mathrm{while}(G(i) \neq x) i++; \\
& \mathrm{output}(1);
\end{aligned}
\qquad
\begin{aligned}
P'' \equiv\ & \mathrm{input}(x); \\
& i := 0; \\
& \mathrm{while}(\mathrm{true}) \\
& \quad \mathrm{if}(F(i) = x)\ \mathrm{output}(1); \\
& \quad \mathrm{if}(G(i) = x)\ \mathrm{output}(x); \\
& \quad i++;
\end{aligned}
$$

i due programmi che calcolano rispettivamente $A \cap B$ e $A \cup B$. Le loro semantiche sono

$$
\varphi_{P'} = \begin{cases} 1 & \text{se } x \in A \cap B \\ \perp & \text{altrimenti} \end{cases}
\qquad
\varphi_{P''} = \begin{cases} 1 & \text{se } x \in A \cup B \\ \perp & \text{altrimenti} \end{cases}
$$

da cui ricaviamo che

$$
A \cap B = \mathrm{Dom}_{\varphi_{P'} \in \mathcal{P}} \qquad A \cup B = \mathrm{Dom}_{\varphi_{P''} \in \mathcal{P}}
$$

Entrambe le funzioni sono, dunque, ricorsive numerabili per la seconda caratterizzazione. □

Teorema di Rice

Il teorema di Rice è un potente strumento per mostrare che gli insiemi appartenenti a una certa classe non sono ricorsivi.

Sia $\{\varphi_i\}$ un spa.
Insiemi che rispettano le funzioni $\to I \subseteq \mathbb{N}$ (insieme di programmi) rispetta le funzioni sse $(a \in I \land \varphi_a = \varphi_b) \Rightarrow b \in I$.

In sostanza, I rispetta le funzioni sse data una funzione calcolata da un programma in I, allora I contiene tutti i programmi che calcolano quella funzione.

Esempio:
$I = \{x \in \mathbb{N} : \varphi_3 = 5\}$ rispetta le funzioni. Infatti:

$$\underbrace{a \in I}_{\varphi_a(3)=5} \wedge \underbrace{\varphi_a = \varphi_b}_{\varphi_b(3)=5 \Rightarrow b \in I}$$

Teorema *(Teorema di Rice)* Sia $I \subseteq \mathbb{N}$ un insieme che rispetta le funzioni. Allora I è ricorsivo solo se $I = \varnothing$ oppure $I = \mathbb{N}$.

Dimostrazione Sia I che rispetta le funzioni con $I \neq \varnothing$ e $I \neq \mathbb{N}$.

Per assurdo, assumiamo che I sia ricorsivo.

Dato che $I \neq \varnothing$, esiste almeno un elemento $a \in I$. Inoltre, dato che $I \neq \mathbb{N}$, esiste almeno un elemento $\bar{a} \notin I$.

Definiamo la funzione $t : \mathbb{N} \to \mathbb{N}$ come:

$$t(n) = \begin{cases} \bar{a} & \text{se } n \in I \\ a & \text{se } n \notin I \end{cases}$$

Sappiamo che $t \in \mathcal{T}$, dato che è calcolabile dal seguente programma

$$\begin{aligned} P \equiv\ & \text{input}(x); \\ & \text{if}(P_I(n) = 1)\ \text{output}(\bar{a}); \\ & \text{else output}(a) \end{aligned}$$

Notiamo che P si arresta sempre e calcola $t(n) \Rightarrow t \in \mathcal{T}$.
Il **teorema di Ricorsione** assicura in un spa $\{\varphi_i\}$ l'esistenza di un $d \in \mathbb{N}$ tale che

$$\varphi_d = \varphi_{t(d)}$$

Per tale d, ci sono solo due possibilità rispetto a I:
- $d \in I$:
  dato che I rispetta le funzioni e $\varphi_d = \varphi_{t(d)}$, allora $t(d)$ devve essere in I. Ma $t(d) = \bar{a} \notin I \Rightarrow$ **contraddizione**;
- $d \notin I$:
  $t(d) = a \in I$ e I rispetta le funzioni. Dato che $\varphi_d = \varphi_{t(d)}$, devve essere che $d \in I \Rightarrow$ **contraddizione**.

Assumere I ricorsivo ha portato ad un assurdo. □

Mostrare che un insieme non è ricorsivo

Il teorema di Rice suggerisce un approccio per stabilire se un insieme $A \subseteq \mathbb{N}$ **non è ricorsivo**:
1. Mostrare che A rispetta le funzioni;
2. Mostrare che $A \neq \varnothing$ e $A \neq \mathbb{N}$;
3. A non è ricorsivo per Rice.

Limiti verifica automatica del software

Definiamo:

- **specifiche** = descrizione di un problema e richiesta per i programmi che devono risolverlo automaticamente. Un programma è *corretto* se risponde alle specifiche;
- **problema** = *Posso scrivere un programma V che testa automaticamente se un programma sia corretto o meno?*

$$P \rightsquigarrow V(P) = \begin{cases} 1 & \text{se } P \text{ è corretto} \\ 0 & \text{se } P \text{ è errato} \end{cases}$$

Chiamiamo $\mathrm{PC} = \{P : P \text{ è corretto}\}$. Osserviamo che esso rispetta le funzioni

$$\underbrace{P \in \mathrm{PC}}_{P \text{ è corretto}} \wedge \underbrace{\varphi_P = \varphi_{P'}}_{P' \text{ è corretto}} \Rightarrow P' \in \mathrm{PC} \Rightarrow \mathrm{PC} \text{ non è ricorsivo}$$

Dato che PC non è ricorsivo, la correttezza dei programmi non può essere testata automaticamente.

Esistono, però, dei casi limite in cui è possibile costruire dei test automatici:

- specifiche = "nessun programma è corretto" $\Rightarrow \mathrm{PC} = \emptyset$
- specifiche = "tutti i programmi sono corretto" $\Rightarrow \mathrm{PC} = \mathbb{N}$

entrambi i PC sono ovviamente ricorsivi e quindi possono essere testati automaticamente.