Problemi decidibili, semidecidibili, indecidibili

(Paragrafo 3.5 delle dispense)

Il problema di determinare se una formula A della logica **proposizionale** sia valida o no può essere risolto mediante un **procedimento automatico** o un **algoritmo**:
il numero di interpretazioni di A è finito, e il metodo per determinare se A è vera in ciascuna interpretazione è **meccanico**.

Cos'è un procedimento automatico, cosa significa che un metodo è meccanico o automatizzabile?

Più in generale: cosa significa che una **funzione** è **calcolabile** in modo meccanico/automatico, o i cui valori si possono calcolare mediante un algoritmo?

Funzioni calcolabili

Anni 30: caratterizzazioni formali della **calcolabilità**, cioè definizione formale di classi di funzioni **effettivamente calcolabili**.

funzioni ricorsive
funzioni calcolabili da una macchina di Turing
funzioni definibili nel λ-calcolo
...

Le diverse caratterizzazioni sono state dimostrate equivalenti

Tesi di Church: le funzioni effettivamente calcolabili sono esattamente quelle caratterizzate da questi sistemi formali

Linguaggi di programmazione **computazionalmente completi**: vi si possono definire tutte le funzioni ricorsive

Se si accetta la tesi di Church: funzioni calcolabili = funzioni che possono essere definite in un linguaggio di programmazione computazionalmente completo

Problemi di decisione

Problemi la cui soluzione è SI o NO

"dato un oggetto x di un insieme S, determinare se x ha la proprietà P oppure no"

$$\Pi_{S,A}\text{: dato } x \in S \text{ e } A \subseteq S\text{, determinare se } x \in A$$

La soluzione di un problema di decisione equivale al calcolo della

Funzione caratteristica

f_A dell'insieme A, tale che per ogni $x \in S$:

$$f_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{altrimenti} \end{cases}$$

Problemi decidibili e indecidibili

Procedura di decisione per $\Pi_{S,A}$: procedimento automatico che, dato $x \in S$ termina con risultato TRUE se $x \in A$, FALSE altrimenti

Una procedura di decisione per $\Pi_{S,A}$ equivale al calcolo della funzione caratteristica di A.

Un problema (di decisione) $\Pi_{S,A}$ è **decidibile** se per esso esiste una procedura di decisione, cioè se la funzione caratteristica di A è calcolabile

Un problema $\Pi_{S,A}$ è **indecidibile** se la funzione caratteristica di A non è effettivamente calcolabile

Problemi indecidibili: **è dimostrata l'impossibilità** di soluzione automatica, cioè è impossibile (assurdo) che esista un programma che li risolva

Problemi semi-decidibili

Un problema $\Pi_{S,A}$ è semidecidibile se esiste una **procedura di semi-decisione** per $\Pi_{S,A}$:

> procedimento automatico che, dato qualunque elemento $x \in S$,
>
> - se $x \in A$ termina riportando TRUE.
> - Ma se $x \notin A$ il procedimento potrebbe non terminare.

Esempio di procedura di semi-decisione: macchina M che, dato x, genera ad uno ad uno tutti gli elementi di A. Se incontra x, si ferma e riporta *TRUE*. Se $x \notin A$ e A è infinito, M non si ferma mai.

Un problema $\Pi_{S,A}$ è semidecidibile sse A è **ricorsivamente enumerabile** (enumerabile in modo automatico), cioè se è l'immagine di una funzione calcolabile

Il problema della fermata

Dato un qualsiasi programma p e un suo input x, determinare se l'esecuzione di p con input x termina oppure no.

Indichiamo con $p(x)$ l'esecuzione di p con input x.

Il problema della fermata è decidibile sse esiste un programma M (in un linguaggio computazionalmente completo) tale che per ogni programma p e input x per p:

$$M(p, x) = \begin{cases} \mathit{TRUE} & \text{se } p(x) \text{ termina} \\ \mathit{FALSE} & \text{se } p(x) \text{ non termina} \end{cases}$$

Il problema della fermata è indecidibile (I)

Supponiamo, per assurdo, che esista un programma M tale che per ogni programma p e input x per p:

$$M(p,x) = \begin{cases} \mathit{TRUE} & \text{se } p(x) \text{ termina} \\ \mathit{FALSE} & \text{se } p(x) \text{ non termina} \end{cases}$$

Scriviamo il programma K con un unico input:

$$K(p) = M(p,p)$$

$$K(p) = \begin{cases} \mathit{TRUE} & \text{se } p(p) \text{ termina} \\ \mathit{FALSE} & \text{se } p(p) \text{ non termina} \end{cases}$$

Il problema della fermata è indecidibile (II)

Sia **loop** un qualsiasi programma che non termina mai, e definiamo il programma N:

$$N(p) \textbf{ = if } K(p) = \mathit{TRUE} \textbf{ then loop else return } \mathit{FALSE}$$

$$N(p) \begin{cases} \text{non termina} & \text{se } p(p) \text{ termina } (K(p) = \mathit{TRUE}) \\ \text{termina} & \text{se } p(p) \text{ non termina } (K(p) = \mathit{FALSE}) \end{cases}$$

Cosa succede quando il programma N viene eseguito con input N stesso?

$$N(N) \begin{cases} \text{non termina} & \text{se } N(N) \text{ termina} \\ \text{termina} & \text{se } N(N) \text{ non termina} \end{cases}$$

Contraddizione: un tale programma N non può esistere

Quindi è assurda l'ipotesi iniziale che esista un algoritmo M che, dato un qualsiasi programma p e un suo input x, termina con output *TRUE* se $p(x)$ termina e termina con output *FALSE* se $p(x)$ non termina.

Decidibilità e indecidibilità di una logica

Una logica è decidibile se il problema di determinare se $S \models A$ è decidibile, cioè se esiste un algoritmo che, dato qualunque insieme S di formule e una formula A, termina con output *TRUE* se $S \models A$ e termina con output *FALSE* se $S \not\models A$.

La logica proposizionale è decidibile

La logica dei predicati è indecidibile: il problema della fermata si può ridurre al problema di determinare se $S \models A$, per S e A opportuni:

È possibile definire un procedimento automatico che, dato qualsiasi programma p e input x, costruisce un insieme di formule $S_{p,x}$ e una formula $A_{p,x}$ tali che:

$S_{p,x} \models A_{p,x}$ se e solo se l'esecuzione di p con input x termina.

Sistemi assiomatici e semidecidibilità

Un **sistema assiomatico** $\mathcal{I}$ è un sistema di inferenza, in cui la proprietà di essere un assioma di $\mathcal{I}$ e la relazione di derivabilità mediante ciascuna regola di inferenza di $\mathcal{I}$ sono nozioni decidibili.

Una logica $\mathcal{L}$ è **assiomatizzabile** se esiste un sistema assiomatico corretto e completo per $\mathcal{L}$.

> Se una logica $\mathcal{L}$ è assiomatizzabile e il suo linguaggio è numerabile, allora $\mathcal{L}$ è semidecidibile

Dati S e A:

- Enumeriamo tutte le sequenze finite di simboli del linguaggio
- Per ciascuna di esse, determiniamo se è una derivazione di A da S in $\mathcal{I}$.

Se $S \models A$ allora $S \vdash_{\mathcal{I}} A$ quindi prima o poi si troverà una derivazione di A da S e il procedimento termina.