Come si dimostra l'indecidibilità della logica dei predicati
(**Studiare** il paragrafo 2.6 del libro di logica)

Data una proprietà P, il problema $P(x)$ è decidibile se esiste un procedimento effettivo (algoritmo, programma) che, dato qualunque input x, termina con output `true` se vale $P(x)$ e termina con output `false` se non vale $P(x)$.

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 $S \not\models A$.

L'indecidibilità della logica dei predicati si può dimostrare mostrando come ridurre a un problema di conseguenza logica un problema la cui indecidibilità è nota, ad esempio il

Problema della fermata: dato un programma p e un suo input x, l'esecuzione di p con input x termina?

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

Il problema della fermata è indecidibile: infatti,

supponiamo per assurdo

che esista un algoritmo (programma) `M` che, dato un qualsiasi programma p e un suo input x, termina con output `true` se $p(x)$ si ferma e termina con output `false` se $p(x)$ non si ferma:

`M(`p, x`) = true` se l'esecuzione di p con input x termina, `false` altrimenti.

Possiamo allora scrivere un programma `K` con un unico input:

`K(`p`) = M(`p, p`)`

Quindi `K(`p`) = true` se l'esecuzione di p con input p termina, `false` altrimenti.

Sia ora `loop` un qualsiasi programma che non termina mai (e.g. `while true do done`), e consideriamo il seguente programma `N` con un unico input:

`N(`p`) = if K(`p`) then loop else false`

Questo programma è tale che, per ogni programma `P`, se l'esecuzione di `P` con input `P` termina (`K(P)=true`), allora `N(P)` non si ferma, altrimenti `N(P)` si ferma.

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

```
N(N)  = if K(N) then loop else false
      = if N con input N si ferma then loop else false
```

Quindi se `N(N)` si ferma (`K(N)` riporta `true`), allora `N(N)` non si ferma e se `N(N)` non si ferma (`K(N)` riporta `false`), allora `N(N)` si ferma.

Contraddizione

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 $p(x)$ si ferma e termina con output `false` se $p(x)$ non si ferma.

Riduzione del problema della fermata a un problema di conseguenza logica (sketch): si definisce 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:

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

Se il problema di determinare se $S \models A$ fosse decidibile, sarebbe allora decidibile anche il problema della fermata: dati p e x, si costruiscono i corrispondenti $S_{p,x}$ e $A_{p,x}$ e si decide se $S_{p,x} \models A_{p,x}$.

Quindi la logica dei predicati è indecidibile.