Riduzione fra da problema Fermata a problema Fermata_input_0

Problema Fermata_input_0
Input: una stringa che descrive una MdT M
Output: 1 se M si ferma con input 0

Apparentemente Fermata_input_0
sembra più semplice di Fermata
(dobbiamo risolvere solo per un input)

Riduzione da problema Fermata a problema Fermata_input_0

Algoritmo: (obiettivo risolvere: Fermata assumendo che esista alg. $A(M)$ che risolve Fermata_input_0 per ogni M)

Input: M,x ; Output Fermata(M,x)

1. Sia $N_{M,x}$ la MdT che risolve il seguente: "su input $z \in \{0,1\}^*$ valuta M su input x e ritorna il risultato" (Nota: la macchina $N_{M,x}$ ignora z; inoltre x è nella sua codifica)
2. Ritorna $y = A(N_{M,x})$ (dice se $N_{M,x}$ termina su input $z = 0$, che verrà ignorato da $N_{M,x}$).

Riduzione da problema Fermata a problema Fermata_input_0

Algoritmo: (obiettivo risolvere il problema della Fermata assumendo che esista alg. A(M) che risolve Fermata_input_0 per ogni M)

La prova è per assurdo e controintuitiva:

Ragionamento del tipo se "I maiali fischiano (probl. fermata decidibile) allora i cavalli volano (probl. fermata su input zero decidibile)"

Poiché so che i maiali non fischiano (problema fermata non decidibile) allora deduco che i cavalli non volano (fermata con input zero non decidibile)

Riduzione da problema Fermata a problema Fermata_input_0

Prova formale: idea (difficoltà principale) è pensare che in $N_{M,x}$ la stringa x non è input di M ma è una nuova costante inserita nel programma che definisce una macchina di Turing diversa da M.

L'algoritmo NON esegue $N_{M,x}$ ma semplicemente scrive la descrizione di $N_{M,x}$ come una stringa e fornisce questa stringa in input a A (che risolve il prob. Fermata_input_0)

Lemma: data la stringa M,x,z la macchina $N_{M,x}$ si ferma su input z se e solo se M si ferma con input x

Riduzione da problema Fermata a problema Fermata_input_0

Lemma: data la stringa M,x,z la macchina $N_{M,x}$ si ferma su input z se e solo se M si ferma con input x

Prova: $N_{M,x}$ ignora input z (fa la stessa cosa qualunque sia z); $N_{M,x}$ si ferma se e solo se M si ferma su input x

In altre parole

$$\text{Fermata}(M,x) = \text{Fermata\_input\_0}(N_{M,x})$$

NB: la descrizione di M e del suo input sono stringhe di bit; x in questo caso NON e' input di una MdT ma parte della descrizione di $N_{M,x}$

IMPORTANTE: per dimostrare indecidibilità la riduzione va da problema A che so indecidbile a problema B che voglio dimostrare indecidibile

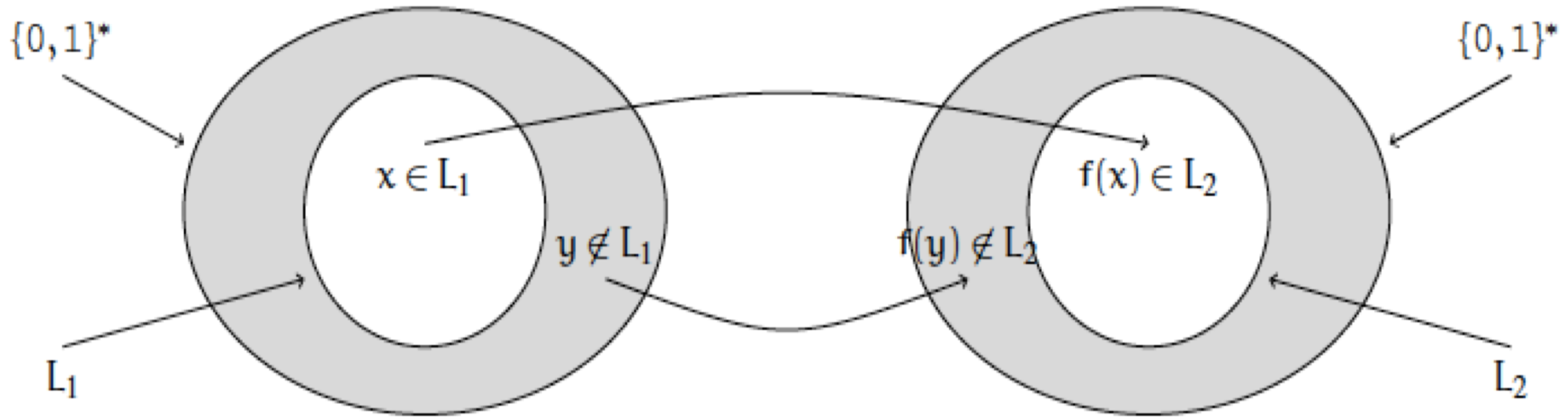
La riduzione mostra che se sapessi risolvere B allora risolvo anche problema A

Quindi B non è più facile di A

La riduzione nella direzione opposta non fornisce nulla (mi dice che A non è più facile di B ma non dice nulla su quanto sia difficile B)

Un linguaggio L1 è riducibile a un linguaggio L2 se esiste una funzione totale calcolabile $f : \{0,1\}^* \rightarrow \{0,1\}^*$, detta riduzione, tale che, per ogni stringa binaria x, $x \in L1$ se e solo se $f(x) \in L2$ (x in L1 se e solo se f(x) in L2)

Figura 3.4: riducibilità tra linguaggi



Siano L1 e L2 due linguaggi tali che L1 è riducibile a L2.

1- Se L2 è decidibile, allora L1 è decidibile.

2- Se L1 non è decidibile, allora L2 non è decidibile.

Intuizione se L1 riducibile a L2 allora

- L1 non è più difficile di L2 (vedi 1 sopra)
- L2 è almeno tanto difficile quanto L1 (2 sopra)

Risultato_sempre_0: data MdT M output 1 se e solo se M da in output 0 per ogni input

Linguaggio associato:

$L_{sempre\ 0}$={codifica di M tale che output di M è 0 per ogni input }

Teorema: Risultato_sempre_0 è indecidibile

Prova: Riduzione da Fermata_input_0 a problema Risultato_sempre_0

NB: la riduzione va da problema che so indecidibile a problema che voglio dimostrare indecidibile