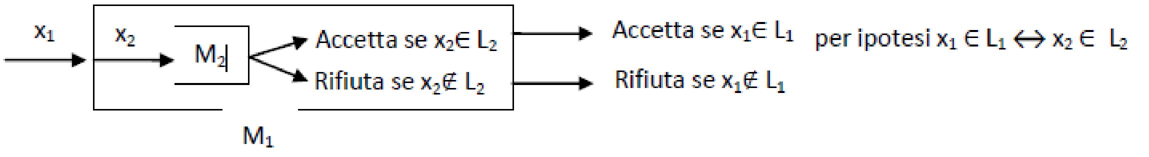
Riduzioni - Teorema Di Rice

Elementi di Teoria della Computazione (Università degli Studi di Salerno)

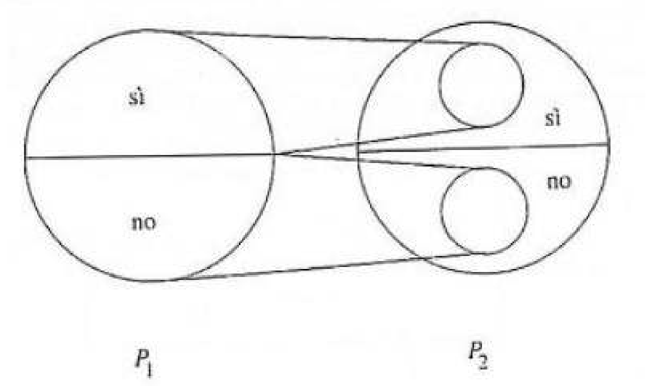
Riduzioni

Diciamo che un linguaggio L_1 può essere ridotto ad uno L_2, ovvero $L_1 \leq_m L_2$, se esiste una funzione f tale che
f: $x1 \rightarrow x2$ = f(x_1) con $x_1 \in L_1$ ed $x_2 \in L_2$
Possiamo definire 2 proprietà:
a) Se $L_1 \leq_m L_2$ e $L_2 \in$ RE, allora $L_1 \in$ RE
b) Se $L_1 \leq_m L_2$ e L_2 è Ricorsivo, allora L_1 è ricorsivo

Dim a)



Dim b)
Se abbiamo un algoritmo per convertire le istanze di un problema P1 in istanze di un problema P2 che hanno la stessa risposta, allora diciamo che P1 si riduce a P2. Possiamo avvalerci di questa dimostrazione per provare che P2 è "difficile" almeno quanto P1. Di conseguenza, se P1 non è ricorsivo, allora P2 non può essere ricorsivo. Se P1 è non RE, allora P2 non può essere RE.



Una riduzione deve trasformare qualsiasi istanza di P1 che ha una risposta affermativa ("sì") in un'istanza di P2 con una risposta affermativa ("si"), e ogni istanza di P1 con una risposta negativa ("no") in un'istanza di P2 con una risposta negativa ("no"). Non è essenziale che ogni istanza di P2 sia l'immagine di una o più istanze di P1: di fatto è normale che solo una piccola frazione di P2 sia l'immagine della riduzione. In termini formali una riduzione da P1 a P2 è una macchina di Turing che riceve un'istanza di P1 scritta sul nastro e si arresta con un'istanza di P2 sul nastro. Nella pratica descriveremo generalmente le riduzioni come se fossero programmi per computer che ricevono in input un'istanza di P1 e producono come output un'istanza di P2.

Teorema 9.7 Se esiste una riduzione da P1 a P2, allora:
1. se P1 è indecidibile, lo è anche P2
2. se P1 è non RE, lo è anche P2.

Il teorema di Rice e le proprietà dei linguaggi RE

Tutte le proprietà non banali dei linguaggi RE sono indecidibili, nel senso che è impossibile riconoscere per mezzo di una macchina di Turing le stringhe binarie che rappresentano codici di una MT il cui linguaggio soddisfa la proprietà. Un esempio di proprietà dei linguaggi RE è "il linguaggio è libero dal contesto". Come caso speciale del principio generale che tutte le proprietà non banali dei linguaggi RE sono indecidibili, il problema se una data MT accetti un linguaggio libero dal contesto è indecidibile.
Una **proprietà** dei linguaggi RE è semplicemente un insieme dei linguaggi. La proprietà di essere vuoto è l'insieme $\{\emptyset\}$, che consiste del solo linguaggio vuoto.

Una proprietà è **banale** se è vuota (ossia se non viene soddisfatta da nessun linguaggio) o comprende tutti i linguaggi RE.
- ✓ P=RE è una proprietà banale (tutti i linguaggi la verificano)
- ✓ P=$\emptyset$ è una proprietà banale (nessun linguaggio la soddisfa)

Altrimenti è **non banale**:
- Osserviamo che la proprietà vuota, $\emptyset$, è diversa dalla proprietà di essere un linguaggio vuoto $\{\emptyset\}$.
- P=$\{\emptyset\}$ è non banale

Non possiamo riconoscere un insieme di linguaggi come i linguaggi stessi. La ragione è che il tipico linguaggio, essendo infinito, non può essere espresso come una stringa di lunghezza finita che possa essere l'input di una MT. Dobbiamo piuttosto riconoscere le macchine di Turing che accettano quei linguaggi; il codice della MT è finito anche se il linguaggio che accetta è infinito. Di conseguenza, se *P* è una proprietà dei linguaggi RE, il linguaggio L_P è l'insieme delle macchine di Turing M_i tali che $L(M_i)$ è un linguaggio in *P*. Quando parliamo di decidibilità di una proprietà *P*, intendiamo la decidibilità del linguaggio L_P.

Teorema di Rice

TH. Ogni proprietà non banale dei linguaggi RE è indecidibile.

Dim.
Sia *P* una proprietà non banale dei linguaggi RE. Per cominciare, supponiamo che $\emptyset$, il linguaggio vuoto, non sia in *P*; ci occuperemo più tardi del caso opposto. Dato che *P* è non banale, deve esistere un linguaggio non vuoto L che sia in *P*. Sia M_L un MT che accetta L.
Ridurremo L_u a L_P, dimostrando in questo modo che L_P è indecidibile, dal momento che L_u è indecidibile. L'algoritmo per la riduzione riceve in input una coppia (M,w) e produce una MT M'. Lo schema di M' è illustrato nella figura 9.10; L(M') è $\emptyset$ se M non accetta w, e L(M') = L se M accetta w.

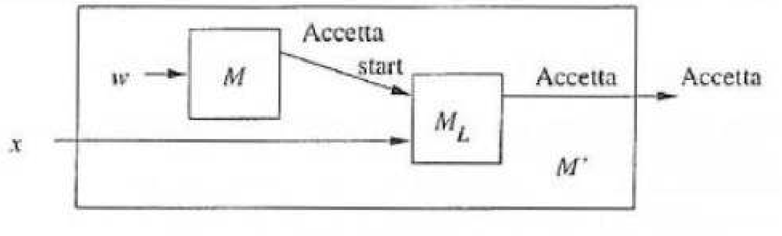


Figura 9.10 Costruzione di M' per la dimostrazione del teorema di Rice.

M' è una MT a due nastri. Un nastro è usato per simulare M su w. Ricordiamo che l'algoritmo che effettua la riduzione riceve come input M e w, e li usa nella definizione delle transizioni di M'. Di conseguenza la simulazione di M su w è incorporata in M'; la seconda MT non deve leggere le transizioni di M sui suoi nastri.

Se necessario, l'altro nastro di M' viene usato per simulare M_L sull'input x di M'. Anche qui le transizioni di M_L sono note all'algoritmo di riduzione e possono essere incorporate nelle transizioni di M'. La MT M' è costruita per operare nel modo seguente.

1. Simula M su input w. Osserviamo che w non è l'input di M'; M' scrive invece M e w su uno dei due nastri e simula la MT universale U su tale coppia.
2. Se M non accetta w, allora M' non fa nient'altro. M' non accetta mai il proprio input x, per cui L(M') = ∅. Poiché assumiamo che ∅ non sia nella proprietà *P*, ciò significa che il codice di M' non è in L_P.
3. Se M accetta w, allora M' comincia a simulare M_L sul proprio input x. Perciò M' accetterà esattamente il linguaggio L. poiché L è in *P* il codice di M' è in L_P.

Si può notare che la costruzione di M' da M e w può essere realizzata da un algoritmo. Dato che tale algoritmo trasforma (M,w) in una M' che è in L_P se e solo se (M,w) è in L_U, esso è una riduzione di L_u a L_P, e dimostra che la proprietà *P* è indecidibile.
Ci resta da considerare il caso in cui ∅ è in *P*. Esaminiamo allora la proprietà complemento P , l'insieme dei linguaggi RE che non hanno la proprietà *P*. Per quanto abbiamo visto sopra, P è indecidibile. Ma poiché ogni MT accetta un linguaggio RE, L_p l'insieme di (codici per) macchine di Turing che non accettano un linguaggio in *P* è uguale a L_P , l'insieme di MT che accettano un linguaggio in complemento di P . Supponiamo che L_P sia decidibile. Allora lo sarebbe anche L_P , perché il complemento di un linguaggio ricorsivo è ricorsivo. L_P quindi non è ricorsivo.