

Teorema 9.7. Se esiste una riduzione da P_1 a P_2 , allora:

- a) se P_1 è indecidibile, lo è anche P_2
- b) se P_1 è non RE, lo è anche P_2 .

DIMOSTRAZIONE Supponiamo dapprima che P_1 sia indecidibile. Se è possibile decidere P_2 , allora possiamo combinare la riduzione da P_1 a P_2 con l'algoritmo che decide P_2 per costruire un algoritmo che decide P_1 . L'idea è illustrata graficamente nella Figura 8.7. Più in dettaglio, supponiamo di avere un'istanza w di P_1 . Applichiamo a w l'algoritmo che la converte in un'istanza x di P_2 . Applichiamo poi a x l'algoritmo che decide P_2 . Se l'algoritmo dice "sì", allora x è in P_2 . Poiché abbiamo ridotto P_1 a P_2 , sappiamo che la risposta a w per P_1 è "sì", ossia w è in P_1 . Analogamente, se x non è in P_2 , allora w non è in P_1 , e qualunque risposta sia data alla domanda " x è in P_2 ?" è anche la risposta corretta a " w è in P_1 ?".

Abbiamo così contraddetto l'ipotesi che P_1 sia indecidibile. Concludiamo che se P_1 è indecidibile, allora anche P_2 è indecidibile.

Consideriamo ora la parte (b). Supponiamo che P_1 sia non RE e P_2 sia RE. Disponiamo di un algoritmo che riduce P_1 a P_2 , ma abbiamo solo una procedura per riconoscere P_2 ; in altre parole esiste una TM che dice "sì" se il suo input è in P_2 , ma può non arrestarsi se il suo input non è in P_2 . Come per (a), tramite l'algoritmo di riduzione convertiamo un'istanza w di P_1 in un'istanza x di P_2 . Applichiamo poi a x la TM per P_2 . Se x è accettato, accettiamo w .

Questa procedura descrive una TM (che può anche non arrestarsi) il cui linguaggio è P_1 . Se w è in P_1 , allora x è in P_2 , per cui questa TM accetterà w . Se w non è in P_1 , allora x non è in P_2 . La TM può arrestarsi o no, ma di certo non accetterà w . Poiché abbiamo presupposto che non esista nessuna TM per P_1 , abbiamo dimostrato per assurdo che non esiste nessuna TM neanche per P_2 ; in altre parole, se P_1 è non RE, allora anche P_2 è non RE. \square