Numerabilità e Turing calcolabilità

Ogni linguaggio numerabile è Turing riconoscibile? Si motivi la risposta.

Risposta. NO. Le TM sono codificabili in binario, cosi' some le parole sull'alfabeto di input.
Poichè {0,1}* è numerabile, il complemento di A_{TM} = {<T,w> | T è una TM e w è in L(T)} è un sottoinsieme infinito di un insieme numerabile e in quanto tale numerabile.
Ma si è dimostrato che il complemento di A_{TM} non è Turing riconoscibile.

Turing riconoscibilità

Se un linguaggio L <u>non</u> è Turing riconoscibile, il suo complemento può essere
1.decidibile?
2.Turing-riconoscibile?

decidibile? NO

Turing-riconoscibile? SI

Enumerabilità

Il linguaggio $\neg HALT_{TM}$ = {<T,w> | T è una TM e T non si ferma su w} è enumerabile?

(Sugg. Si supponga per assurdo che $\neg HALT_{TM}$ sia enumerabile e si costruisca una TM che riconosce $\neg HALT_{TM}$, utilizzando la TM che calcola la funzione di numerazione)

Enumerabilità e Turing calcolabilità

Il linguaggio $\neg HALT_{TM}$ = {<T,w> | T è una TM e T non si ferma su w} è enumerabile?

Risposta:

Sappiamo che $\neg HALT_{TM}$ è numerabile e quindi che esiste una funzione biettiva f : N → ¬HALTM. Tale funzione non può essere Turing calcolabile altrimenti potrei utilizzarla per costruire una TM che riconosce ¬HALTM.
Supponiamo per assurdo che sia T_f la TM che calcola f e costruiamo una TM M a 3 nastri che riconosce ¬HALTM:
M = Input <T,w>
1. inizializza il II nastro a 1
2. sia k il numero sul II nastro, esegui T_f su k utilizzando il III nastro.
3. confronta f(k) con <T,w>, se le due codifiche sono uguali allora accetta altrimenti incrementa k sul II nastro e torna al punto 2.

M si ferma e accetta se T non si ferma su w e non si ferma altrimenti, quindi M Turing riconosce ¬HALTM, cosa che sappiamo impossibile. La contraddizione nasce dall'aver supposto f calcolabile.

Decidibilità

Si dimostri che il linguaggio L = {<T,x,n> | T accetta x in n passi} è decidibile.

Soluzione:
Utilizziamo una variante della UTM, M:

input <T,x,n>
passo1. inzializziamo un contatore m a 1
passo2. esegui un passo di T su x
passo 3. se T accetta, accetta
altrimenti se m=n rifiuta, altrimenti incrementa m e torna al passo 2

T accetta x in n passi sse M accetta <T,x,n>