

Domanda

Ciao a tutti. Una domanda: nella nostra definizione di enumeratore c'è il concetto di terminazione?

A me non risulta, quindi ho un problema. In particolare si trova nella dimostrazione di "un linguaggio è decidibile se e solo se esiste un enumeratore che lo enumera seguendo l'ordinamento standard delle stringhe", nella parte in cui presupponiamo esista un enumeratore e costruiamo la relativa TM che decide il linguaggio.

Il fatto è che il linguaggio enumerato potrebbe essere finito e contenere solo parole minori dell'input della TM. In questo caso, la TM aspetterebbe "all'infinito" la stampa di una parola da parte dell'enumeratore.

Allego un esempio per capirci meglio. Nell'esempio, l'enumeratore enumera il linguaggio composto da 1 e 11.

Alla prima stampa, la TM vede che 1 è minore di 111 e quindi continua.

Alla secondo, la TM vede che 11 è minore di 111 e quindi continua.

Non essendoci poi più stampe, è come se la TM andasse in loop aspettando nuove parole, non può rifiutare!

Risposta

Domanda superlecita e che richiede un commento passo per passo. Il problema non è nel teorema stesso, ma nell'incompletezza della dimostrazione quando non si esplicita il comportamento dell'enumeratore su linguaggi finiti.

Il problema che hai individuato è reale solo se si usa una definizione di enumeratore che non prevede l'halting sui linguaggi finiti. Esistono due approcci equivalenti:

- Convenzione: enumeratore termina su linguaggi finiti
- Modifica algoritmica: decisore gestisce esplicitamente la non-terminazione

La soluzione standard prevede una convenzione implicita sugli enumeratori.

Per linguaggi finiti, l'enumeratore deve terminare immediatamente dopo l'ultima stringa - solito Sipser che ce lo dice: an enumerator for a finite language will eventually halt. Qualsiasi enumeratore che, terminato L, restasse in loop può essere "wrappato" affinché stampi un sentinella più grande di tutte le stringhe e poi halti, quindi l'ipotesi non è restrittiva.

La costruzione corretta sarebbe una roba del tipo:

D su input w:

1. Simula E
2. Per ogni stringa x prodotta da E:

- Se $x = w \rightarrow$ accetta
- Se $x > w \rightarrow$ rifiuta

3. Se E termina \rightarrow rifiuta

Analizzando quindi la terminazione:

- $w \in L$: E produce w in tempo finito (accetta)
- $w \notin L$, L infinito: E produce $x > w$ in tempo finito (rifiuta)
- $w \notin L$, L finito: E termina, D rileva la terminazione e rifiuta (rifiuta)

In (quasi) tutta la letteratura, però, l'enumeratore deve terminare dopo l'ultima stringa se il linguaggio è finito; in alternativa si modifica di un rigo la costruzione del decisore per gestire esplicitamente l'halting. In entrambi i casi il teorema resta valido.