Autore: Gabriel Rovesti

Esercizio 1: Definizione formale di enumeratore

Definizione

Un **enumeratore** è una macchina di Turing a due nastri $E = (Q, \Sigma, \Gamma, \delta, q_0, q\_accept, q\_reject)$ dove:

- Q è l'insieme finito degli stati
- Σ è l'alfabeto di input
- Γ è l'alfabeto del nastro ($\Gamma \supseteq \Sigma \cup \{\square, \#\}$)
- $\delta: Q \times \Gamma^2 \rightarrow Q \times \Gamma^2 \times \{L, R, S\}^2$ è la funzione di transizione
- $q_0 \in Q$ è lo stato iniziale
- $q\_accept, q\_reject \in Q$ sono gli stati di accettazione e rifiuto

Il primo nastro funziona come nastro di lavoro, mentre il secondo nastro serve come stampante e viene utilizzato in modalità di sola scrittura con movimento unidirezionale verso destra.

L'enumeratore E inizia con entrambi i nastri vuoti (contenenti solo simboli blank □). Durante la computazione, E può scrivere stringhe sul nastro di output, separate da un simbolo speciale # (delimitatore).

Linguaggio enumerato

Il linguaggio L(E) enumerato da E è definito come:

$L(E) = \{w \in \Sigma^* \mid \text{esiste una computazione di E che scrive w sul nastro di output}\}$

Formalmente, se indichiamo con E(t) la stringa presente sul secondo nastro dopo t passi di computazione (ignorando i simboli blank), allora:

$L(E) = \{w \in \Sigma \mid \exists t, \exists u, v \in \Sigma \text{ tale che } E(t) = u\#w\#v\}$

Questo significa che una stringa appartiene al linguaggio enumerato se e solo se l'enumeratore la scrive sul nastro di output durante la sua computazione (eventualmente preceduta e seguita da altre stringhe e delimitatori).

Esercizio 2: Equivalenza tra linguaggi Turing-riconoscibili ed enumerabili

Teorema: Un linguaggio L è Turing-riconoscibile se e solo se esiste un enumeratore che lo enumera.

Dimostrazione (⇒): Se L è Turing-riconoscibile, allora L è enumerabile

Sia M una macchina di Turing che riconosce L. Costruiamo un enumeratore E per L come segue:

1. E genera sistematicamente tutte le stringhe s_1, s_2, s_3, ... in Σ* seguendo l'ordinamento standard (prima per lunghezza, poi lessicograficamente)
2. Per ogni stringa s_i generata, E simula M su s_i per i passi
3. Se la simulazione di M su s_i si ferma in uno stato di accettazione entro i passi, E scrive s_i sul nastro di output

Formalmente, l'algoritmo di E è:

```
i = 1
while true:
    for each string s in Σ* with |s| ≤ i (in ordine standard):
        Simula M su s per i passi
        Se M accetta s entro i passi:
            Scrivi s sul nastro di output seguito da #
    i = i + 1
```

Questo algoritmo garantisce che:

- Ogni stringa in L sarà eventualmente enumerata (poiché M accetta ogni stringa in L)
- Solo stringhe in L saranno enumerate (poiché M accetta solo stringhe in L)

Dimostrazione (⇐): Se L è enumerabile, allora L è Turing-riconoscibile

Sia E un enumeratore per L. Costruiamo una macchina di Turing M che riconosce L come segue:

Su input w, M opera così:

1. M simula l'enumeratore E
2. Ogni volta che E scrive una stringa s sul nastro di output, M confronta s con w
3. Se s = w, allora M accetta
4. Se s ≠ w, M continua la simulazione di E

Formalmente, l'algoritmo di M è:

```
Su input w:
    Simula E partendo con nastri vuoti
```

```
    Per ogni stringa s scritta da E sul nastro di output:
        Se s = w:
            Accetta
        Altrimenti:
            Continua la simulazione
```

La macchina M accetta l'input w se e solo se E enumera w, il che avviene se e solo se w ∈ L. Pertanto, M riconosce esattamente L, dimostrando che L è Turing-riconoscibile.

Conclusione

Le due direzioni della dimostrazione stabiliscono che un linguaggio è Turing-riconoscibile se e solo se esiste un enumeratore che lo enumera.