Automi e Linguaggi Formali – 15/7/2024
Secondo appello – Seconda parte

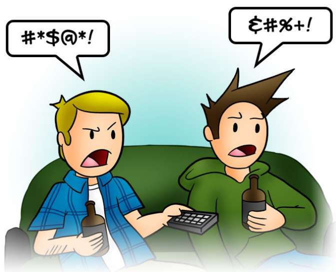
1. (12 punti) Le macchine di Turing con un solo stato sono definite "stateless". Queste macchine rimangono nello stesso stato per tutta la durata della computazione. L'unico modo in cui possono ricordare qualcosa è scriverlo sul nastro. Consideriamo una variante di Turing Machine, chiamata JTM, che è stateless, deterministica e a nastro singolo. Le JTM differiscono dalle TM convenzionali nel modo seguente:

- la testina si estende su tre celle consecutive del nastro, e può leggere/scrivere una stringa di tre simboli del nastro tutti insieme;
- il movimento della testina non è in termini di "blocchi di tre celle": ad ogni transizione la testina non salta al blocco di tre celle adiacenti, ma si sposta solo di una cella a destra o a sinistra;
- se la macchina scrive la stringa di tre simboli $\mathbb{YEA}$ sul nastro la computazione termina con accettazione;
- se la macchina scrive la stringa di tre simboli $\mathbb{NAY}$ sul nastro la computazione termina con rifiuto;
- i simboli $\mathbb{A}, \mathbb{E}, \mathbb{N}, \mathbb{Y}$ sono simboli speciali sempre inclusi nell'alfabeto del nastro;
- la macchina ha un solo stato e rimane in quello stato per sempre. Ciò significa che la nozione di "cambio di stato" (e di conseguenza la nozione stessa di "stati") diventa inutile nel contesto di un JTM.

(a) Dai una definizione formale della funzione di transizione di una JTM.

(b) Dimostra che le JTM riconoscono la classe dei linguaggi Turing-riconoscibili. Usa una descrizione a livello implementativo per definire le macchine di Turing.

2. (12 punti) I *grawlix* sono sequenze di simboli senza senso che sostituiscono le parolacce nei fumetti.



Un linguaggio è *volgare* se contiene almeno un grawlix. Considera il problema di determinare se il linguaggio di una TM è volgare.

(a) Formula questo problema come un linguaggio $GROSS_{TM}$.

(b) Dimostra che il linguaggio $GROSS_{TM}$ è indecidibile.

3. (12 punti) In una delle storie delle Mille e una notte, Alì Babà, mentre viaggiava con il suo asino, trovò la grotta in cui i 40 ladroni avevano nascosto il loro bottino. Come cittadino rispettoso della legge, denunciò il fatto alla polizia, ma solo dopo aver tenuto il più possibile per sé. Il problema è che c'è troppo bottino e l'asino non può portarlo tutto: c'è un limite M al peso che l'asino può trasportare. Supponiamo che ognuno degli N oggetti rubati abbia un prezzo $P[i]$ e un peso $W[i]$. Alì Babà può caricare sull'asino un numero sufficiente di oggetti in modo che il prezzo totale sia almeno L?

Formalmente, possiamo rappresentare il problema che Alì Babà deve risolvere con il linguaggio

$$ALIBABA = \left\{ \langle N, P, W, M, L \rangle \;\middle|\; \text{esiste } B \subseteq \{1, \dots, N\} \text{ tale che } \sum_{j \in B} W[j] \leq M \text{ e } \sum_{j \in B} P[j] \geq L \right\}.$$

(a) Dimostra che $ALIBABA$ è un problema NP.

(b) Sappiamo che il linguaggio

$$SUBSET\text{-}SUM = \left\{ \langle S, t \rangle \;\middle|\; S \text{ insieme di naturali, ed esiste } S' \subseteq S \text{ tale che } \sum_{x \in S'} x = t \right\}$$

è NP-completo. Dimostra che $ALIBABA$ è NP-hard, usando $SUBSET\text{-}SUM$ come problema NP-hard di riferimento.