Informatica Teorica 2022/2023 - Esercitazione 3

29 Marzo 2023

melissa.antonelli2@unibo.it

Notazione. Usiamo $\langle \cdot \rangle$ per indicare una codifica ragionevole di uno o piú oggetti in stringhe, senza specificare ulteriormente il metodo di codifica.

Sessione 1

Nozioni Richieste. Complessitá di Tempo; Notazione Asintotica.
Problema 1. Dato il linguaggio:

$$A = \{\mathtt{0}^k \mathtt{1}^k \mid k \geq 0\}$$

considera la (multi-tape) TM M che decide A e programmata come segue:

1. M legge l'input e rigetta se trova uno `0` a destra di un `1`.
2. M legge gli `0`i sul nastro 1 e li copia sul nastro 2.
3. M legge gli `1`i sul nastro 1 e per ogni `1` sul nastro 1 cancella uno `0` sul nastro 2. Se tutti gli `0`i sono cancellati prima che tutti gli `1`i siano letti, M rigetta.
4. Se tutti gli `0`i sono cancellati, M accetta; se qualche `0` resta sul nastro 2, M rigetta.

(a.) Qual'é la *time complexity* di M? (b.) Ed espressa in notazione asintotica?

$$\textit{Tempo previsto}: \ 10/15 \ \textit{minuti}$$

Nozioni Richieste. Caratterizzazion*i* di **NP**.
Problema 2. Considera il problema che dato un insieme di numeri $x_1, \dots, x_k$ e un target t, determina se l'insieme dato contiene un sottoinsieme la cui somma di elementi ha valore t:

$$SSUM = \{\langle S, t\rangle \mid S = \{x_1, \dots, x_k\} \ \& \ \textit{per qualche} \ \{y_1, \dots, y_l\} \subseteq \{x_1, \dots, x_k\}, \sum y_i = t\}.$$

Esempio. $\langle \{4, 11, 12, 21, 28, 50\}, 25\rangle \in$ SSUM, in quanto 4 + 21=25.
Dimostra SSUM $\in$ **NP** (a.) tramite (poly-time) NTM e (b.) tramite verificatore.

$$\textit{Tempo previsto}: \ 15/20 \ \textit{minuti}$$

Nozioni Richieste: Poly-Riduzione; **NP**-Completezza.
Problema 3. Se L é **NP**-completo, $L' \in \mathbf{NP}$ e $L \leq_p L'$, allora L' é **NP**-completo.
Suggerimento. Ricorda che durante la passata esercitazione abbiamo dimostrato che la m-riduzione é transitiva.

$$Tempo\ previsto:\ 15/20\ minuti.$$

Nozioni Richieste: Formule Booleane; Formula Soddisfacibile; Letterale; Clausole; CNF; SAT; 3SAT; Teorema di Cook-Levin.
Problema 4.1. Sapendo che 3-SAT $\in$ **NP**, mostra (ad alto livello che) 3-SAT essere **NP**-completo (utilizza il teorema di Cook-Levin).

$$Tempo\ previsto: 15\ minuti$$

Nozioni Richieste: Space Complexity; PSPACE; TQBF.
Problema 5.1. In generale, un *game* é una competizione tra giocatori opponenti che tentano di raggiungere un obiettivo seguendo date regole. Consideriamo il *formula game* (fg). Sia F una QBF (*quantified Boolean formula*) in PNF (*prenex normal form*):

$$F = \exists \mathbf{X_1} \forall \mathbf{X_2} \exists \mathbf{X_3} \dots Q \mathbf{X_k} G$$

con $Q \in \{\forall, \exists\}$. I due giocatori – diciamo `A` ed `E` – selezionano *a turno* i valori di veritá da attribuire a (gruppi) di variabili. In particolare, `A` assegna i valori alle variabili vincolate da $\forall$ ed `E` a quelle vincolate da $\exists$. A partire dai valori dati si decide chi vince: se $G = 1$, vince `E`; se $G = 0$, vince `A`.

(a.) Consideriamo un semplice esempio.

$$F_1 = \exists x_1 \forall x_2 \exists x_3 ((x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3))$$

Se `E` assegna $x_1 = 1$; poi `A` assegna $x_2 = 0$; `E` assegna $x_3 = 1$,chi vince?

Diciamo che un giocatore ha una *winning strategy* se, seguendo tale strategia, vince per ogni scelta dell'altro giocatore. (b.) Consideriamo

$$F_2 = \exists x \forall y (x \vee \neg y).$$

Il giocatore `E` ha una *winning strategy* su F_2? Se sì, definiscila.
Considerando anche

$$F_3 = \exists x_1 \forall x_2 \exists x_3 ((x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (x_2 \vee \neg x_3)).$$

Il giocatore `A` ha una *winning strategy* per F_3? Se sí, definiscila.

$$Tempo\ previsto:\ 5/10\ minuti$$

Problema 5.2. Consideriamo il problema di determinare quale giocatore abbia una *winning strategy* in un *formula game* associato a una data QBF F:

$$FG = \{\langle F \rangle \mid \texttt{E} \text{ } ha\ winning\ strategy\ nel\ fg\ associato\ a\ F\}$$

Mostra (anche informalmente) che FG é PSPACE-completo.

$$Tempo\ previsto: \ 10/15\ minuti$$

Sessione II

Problema 6.1. Dato il linguaggio

$$U = \{w \in \Sigma^* \mid w\ contiene\ eguale\ numero\ di\ \texttt{0}\ e\ \texttt{1}\},$$

considera la TM M su alfabeto $\Sigma = \{\texttt{0}, \texttt{1}\}$ in grado di "contrassegnare" i simboli dell'alfabeto e tale che, su input w:

1. Scansiona il nastro fino al primo bit (non segato) `0` o `1`

   Se non ne trova alcuno, accetta;

   Altrimenti continua a scansionare fino all primo bit diverso (`1` e `0` resp.)
2. Se non ne trova, rigetta; altrimenti segna i due simboli e ripete la procedura

(Per la finalitá dell'esercizio, consideriamo che essendola macchina considerata un'astrazione, l'estensione dell'alfabeto a "bit contrassegnato" non infici il calcolo della complessitá di tempo, che riferiamo qui alla nostra specifica M.) Qual'é la *time complexity* di M in notazione asintotica?

Problema 6.2. Dato il linguaggio

$$U = \{w \in \Sigma^* \mid w\ ha\ uguali\ \texttt{0}\ e\ \texttt{1}\},$$

considera la multi-tape TM M' su alfabeto $\Sigma = \{\texttt{0}, \texttt{1}\}$, tale che su input w:

1. Scansiona il nastro e copia tutti gli `1`i presenti sul nastro secondario
2. Muove entrambe le testine all'inizio dei rispettivi nastri
3. Fino al raggiungimento della fine dell'input:
   - Scansiona il nastro in input fino a ciascuno `0`
   - Associa `0` a `1` del nastro secondario
4. Se non si ha corrispondenza uno-a-uno, rigetta; Se ciascuno `0` é associato al corrispondente `1`, accetta.

Qual'é la complessitá in tempo di M' espressa in notazione asintotica?

$$Tempo\ previsto:\ 10\ minuti$$

Problema 7. Dimostra PATH $\in$ **P** (senza consultare le slide).
Un algoritmo brute-force per PATH potrebbe fare al caso nostro? Motiva.
Suggerimento. Ricorda che, dato un grafo diretto G con nodi s e t, il problema PATH consiste nel determinare se esiste un percorso diretto da s a t:

$$PATH = \{\langle G, s, t\rangle \mid G\ grafo\ diretto\ con\ percorso\ diretto\ da\ s\ a\ t\}.$$

$$Tempo\ Previsto:\ 15\ minuti$$

Problema 4.2. Sia $F = \bigwedge_{j\in\{1,\dots,n\}} l_j$

$$F \in SAT \quad sse \quad f(F) \in 3SAT.$$

Suggerimento. Considera la funzione ausiliaria g tale che, per ogni letterale l, $g(l) = (l \vee x_1 \vee x_2) \wedge (l \vee x_1 \vee \overline{x_2}) \wedge (l \vee \overline{x_1} \vee x_2) \wedge (l \vee \overline{x_1} \vee \overline{x_2})$.

Problema 8. Dimostra che **NP** $\subseteq$ PSPACE (senza consultare le slide).
Suggerimento. Dimostra che (1) SAT $\in$ PSPACE e (2) per ogni A, se $A \in$ **NP** allora $A \in$ PSPACE.