Prova di esame dei corsi di Fondamenti di Informatica e Informatica Teorica

04 luglio 2016

Problema 1. Sia Σ un alfabeto finito e siano $L_1, L_2 \subseteq \Sigma^*$ due linguaggi tali che L_1 è decidibile e L_2 è accettabile. Cosa si può dire circa le proprietà di accettabilità e decidibilità del linguaggio $L = L_1 \cap L_2^c$? Dimostrare la propria affermazione.

Problema 2. Dato un grafo $G = (V, E)$, sia $\chi(G) = \langle M, P \rangle$ una sua codifica in cui M è la matrice di adiacenza di G e

$$P = \{\langle V_1, V_2, V_3 \rangle : V_1, V_2, V_3 \subseteq V \ \wedge \ V_1 \cup V_2 \cup V_3 = V \ \wedge \ V_1 \cap V_2 = \emptyset \ \wedge \ V_1 \cap V_3 = \emptyset \ \wedge \ V_2 \cap V_3 = \emptyset\}.$$

Si consideri il seguente problema decisionale: dato un grafo (non orientato) $G = (V, E)$, esiste una partizione $\langle V_1, V_2, V_3 \rangle$ di V tale che, per ogni $i = 1, 2, 3$ e per ogni $u, v \in V_i$, $(u, v) \notin E$?

Dopo aver formalizzato la definizione del suddetto problema mediante la tripla $\langle I, S, \pi \rangle$, descrivere un algoritmo che, prendendo in input $\chi(G)$, decide se $\langle G, k \rangle$ è una istanza sì del problema in tempo polinomiale in $|\chi(G)|$.

Rispondere, infine, alla seguente domanda: l'esistenza di tale algoritmo è sufficiente a dimostrare l'appartenenza alla classe **P** del problema?

Problema 3. Sia $k \in \mathbb{N}$ una costante positiva tale che $k \geq 3$.
Si consideri il problema seguente: dato un grafo non orientato $G = (V, E)$, decidere se G è k-colorabile oppure ha un Vertex Cover di cardinalità k.

Dopo aver formalizzato la definizione del suddetto problema mediante la tripla $\langle I, S, \pi \rangle$, se ne dimostri l'appartenenza alla classe **P** o, in alternativa, la **NP**-completezza.

Soluzione

Problema 1. Poiché L_1 è decidibile, esiste una macchina di Turing T_1 (di tipo riconoscitore) tale che, per ogni $x \in \Sigma^*$, $T_1(x)$ termina ed inoltre

$$o_{T_1}(x) = \begin{cases} q_A^1 & \text{se } x \in L_1. \\ q_R^1 & \text{se } x \notin L_1, \end{cases}$$

dove q_A^1 e q_R^1 sono, rispettivamente, gli stati di accettazione e di rigetto di T_1.
Analogamente, poiché L_2 è accettabile, esiste una macchina di Turing T_2 (di tipo riconoscitore) tale che, per ogni $x \in L_2$, $T_2(x)$ termina ed inoltre $o_{T_2}(x) = q_A^2$ dove q_A^2 è lo stato di accettazione di T_2; osserviamo che nulla si può affermare circa le computazioni $T_2(y)$ con $y \notin L_2$.

Osserviamo, ora, che per poter affermare "$x \in L$" è necessario mostrare che $x \in L_1$ **e** $x \notin L_2$: poiché per affermare $x \notin L_2$ è necessario disporre di una macchina di Turing che accetti L_2^c. Dunque, la sola accettabilità di L_2 non è sufficiente ad assicurare la accettabilità di L.

Osserviamo, infine, che invece è accettabile il linguaggio $L^c = (L_1 \cap L_2^c)^c = L_1^c \cup L_2$: infatti, tale linguaggio è accettato dalla macchina che, con input $x \in \Sigma^*$, esegue i passi seguenti

a) simula la computazione $T_1(x)$: se $o_{T_1}(x) = q_A^1$, allora accetta, altrimenti esegue il passo b);

b) simula la computazione $T_2(x)$: se $o_{T_1}(x) = q_A^2$, allora accetta.

Problema 2. Il problema decisionale considerato, che chiameremo PARTIZIONE IN 3 INSIEMI INDIPENDENTI (in breve, $P3II$), può essere formalizzato come di seguito descritto:

- $I_{P3II} = \{G = (V,E) : G \text{ è un grafo non orientato } \}$;
- $S_{P3II}(G) = \{\langle V_1, V_2, V_3 \rangle : \subseteq V_1, V_2, V_3 \subseteq V \ \}$;
- $\pi_{P3II}(G, S_{P3II}(G)) = \exists \langle V_1, V_2, V_3 \rangle \in S_{P3II}(G) : \ V_1 \cup V_2 \cup V_3 = V \ \wedge \ V_1 \cap V_2 = \emptyset \ \wedge \ V_1 \cap V_3 = \emptyset \ \wedge$
  $V_2 \cap V_3 = \emptyset \ \wedge \ \forall i = 1,2,3 \ \forall u, v \in V_i \ [\ (u,v) \notin E \ ]$.

Osserviamo che l'insieme P nella codifica di un grafo G è un sottoinsieme di $S_{PII}(G)$. In particolare, il predicato π_{P3II} del problema può essere espresso nella maniera seguente:

$$\exists \langle V_1, V_2, V_3 \rangle \in P : \forall i = 1,2,3 \ \forall u, v \in V_i \ [\ (u,v) \notin E \ ],$$

Quindi, l'algoritmo richiesto nel testo, che riceve in input la matrice di adiacenza M di un grafo $G = (V,E)$ e l'insieme P di tutte le partizioni di V in insiemi indipendenti in G, è descritto nel seguente frammento di codice, che restituisce `vero` se G è una istanza sì di $P3II$:

trovato $\leftarrow$ `falso`;
while ($P \neq \emptyset \wedge$ trovato$=$ **falso**) **do begin**
    estrai un elemento $\langle V_1, V_2, V_3 \rangle$ da P:
    trovato $\leftarrow$ `vero`;
    for ($i \leftarrow 1$; $i \leq 3$; $i \leftarrow i+1$) **do**
        for ($u \in V_i$) **do**
            for ($v \in V_i$) **do**

8 **if** $(M[u,v] = 1)$ **then** trovato $\leftarrow$ `falso`;
9 **end**;
10 **Output**: trovato.

Analizziamo, ora, la complessità del frammento di codice appena descritto.
Poiché accedere ad un elemento della matrice M ha costo costante, l'istruzione **if** alla linea 8 ha costo costante; pertanto, poiché, per ogni $\langle V_1, V_2, V_3 \rangle \in P$, la cardinalità di V_1, $|V_2|$ e $|V_3|$ è al più $|V|$, il doppio ciclo **for** alle linee 6 e 7 ha costo $\mathbf{O}(|V|^2)$.
Il numero di iterazioni del ciclo **for** alla linea 5 è costante, il numero di iterazioni del ciclo **while** alla linea 2 è $|P|$, e, quindi, il costo del frammento di codice è $\mathbf{O}(|P| \cdot |V|^2)$, ossia, è polinomiale *nella dimensione dell'input* o, in altri termini, è polinomiale in $|\chi(G)|$.

Osserviamo, infine, che la codifica $\chi(G)$ non è una codifica ragionevole di G: infatti, $|P| = 3^{|V|}$ e, quindi, la codifica di G mediante la sola matrice di adiacenza (che codifica tutte le informazioni necessarie ad individuare un grafo e che ha dimensione $|V|^2$) è esponenzialmente più corta di $\chi(G)$. Poiché un problema è in **P** se esiste un algoritmo deterministico che richiede tempo polinomiale nella dimensione di una *codifica ragionevole* delle sue istanze, il frammento di codice non permette di affermare che il problema A è in **P** (e, in effetti, esso coincide con COLORABILITÀ ed è, quindi, **NP**-completo).

Problema 3. Il problema in esame, che indicheremo, in breve, con l'acronimo $3COL \vee kVC$, può essere formalizzato come di seguito descritto:

- $I_{3COL \vee kVC} = \{G = (V,E) : G \text{ è un grafo non orientato } \}$;
- $S_{3COL \vee kVC}(G) = \{ \langle c, V' \rangle : \; c : V \rightarrow \{1,2,3\} \; \wedge \; V' \subseteq V \} \}$;
- $\pi_{3COL \vee kVC}(G, S_{3COL \vee kVC}(G)) = \exists \langle c, V' \rangle \in S_{3COL \vee kVC}(G) : \; \{ \forall (u,v) \in E \; [\; c(u) \neq c(v) \;] \; \} \vee$
  $\{ \; |V'| \leq k \wedge \forall (u,v) \in E \; [u \in V' \; \vee \; v \in V' \;] \; \}$.

Un certificato per una istanza $\langle X, f, k \rangle$ di $3COL \vee kVC$ è una coppia $\langle c, V' \rangle \in S_{3COL \vee kVC}(G)$, e, dunque, poiché $|c| \in \mathbf{O}(|V|)$ e $|V'| \in \mathbf{O}(|V|)$, ha lunghezza $\mathbf{O}(|V|)$. Per verificare un certificato è necessario verificare che c sia una colorazione valida per G (ossia, che colori con colori diversi nodi adiacenti) o che V' sia un Vertex Cover di G di dimensione non superiore a k: poiché sia il problema COLORABILITÀ che il problema VERTEX COVER sono in **NP**, sappiamo che tale verifica richiede tempo polinomiale in $|V|$ e $|E|$). Questo prova che il problema è in **NP**.

Dimostriamo, ora, la completezza del problema per la classe **NP** mediante una riduzione polinomiale dal problema 3COL.
Prima di procedere, cerchiamo di capire perché riduciamo da 3COL invece che da VC. A questo scopo, osserviamo che, poiché k è costante (e, infatti, non è dichiarata nella definizione dell'insieme $I_{3COL \vee kVC}$) la versione di VERTEX COVER di interesse in questo problema è in **P**. Infatti, dato un grafo G, per verificare se G ha un Vertex Cover di k nodi è sufficiente verificare se uno degli $\mathbf{O}(|V|^k)$ sottoinsiemi di V contenenti k nodi sia un Vertex Cover per G: poiché la verifica richiede tempo polinomiale in $|V|$ e poicé k è costante, questo prova che il problema è in **P**. Quindi, ridurre da un problema in **P** non sarebbe di alcuna utilità per mostrare la **NP**-completezza di $3COL \vee kVC$.
Sia, dunque, G una istanza di 3COL; l'istanza corrispondente di $3COL \vee kVC$ è il grafo $\overline{G} = (\overline{V}, \overline{E})$ ottenuto aggiungendo a G $k+1$ nuovi grafi, che non hanno nodi in comune con G, ciascuno dei quali costituito da un singolo arco. Più in dettaglio: $\overline{G} = G \cup G_1 \cup G_2 \cup \ldots \cup G_{k+1}$, dove, per ogni $i = 1, \ldots, k+1$, $G_i = (V_i, E_i)$ e

- $V_i = \{u_i, v_i\}$, e
- $E_i = \{(u_i, v_i)\}$.

Dunque, $\overline{V} = V \cup V_1 \cup \ldots \cup V_{k+1}$ e $\overline{E} = V \cup E_1 \cup \ldots \cup E_{k+1}$.
Se $G = (V,E)$ è una istanza sì di 3COL, allora esiste una colorazione $c : V \to V$ dei nodi di G che non assegna lo stesso colore a due nodi adiacenti (ossia, collegati da un arco); allora la colorazione $\overline{c}$ tale che, per ogni $x \in \overline{V}$

$$\overline{c}(x) = \begin{cases} c(x) & \text{se } x \in V, \\ 1 & \text{se } x \in V_i \text{ e } x = u_i, \\ 2 & \text{se } x \in V_i \text{ e } x = v_i, \end{cases}$$

è una colorazione valida per $\overline{G}$, ossia assegna colori diversi a nodi adiacenti in $\overline{G}$.
Se, invece, $G = (V,e)$ è una istanza no di 3COL, allora non esiste alcuna colorazione di V con 3 colori che non assegni lo stesso colore ad almeno una coppia di nodi adiacenti in G; allora, poiché $\overline{G}$ contiene G, non esiste nemmeno alcuna colorazione di $\overline{V}$ con 3 colori che non assegni lo stesso colore ad almeno una coppia di nodi adiacenti in $\overline{G}$. D'altra parte, poiché ogni Vertex Cover di $\overline{G}$ deve almeno contenere un nodo in ciascun V_i, per $i = 1, \ldots, k+1$, ogni Vertex Cover di $\overline{G}$ ha cardinalità almeno $k+1$. Dunque, $\overline{G}$ è una istanza no di $3COL \vee kVC$.
Poiché costruire $\overline{G}$ a partire da G richiede tempo lineare in $|V|$ e $|E|$, questo dimostra che il problema $3COL \vee kVC$ è **NP**-completo.