Calcolabilitá e Complessitá A.A. 2009/2010

Universitá degli Studi di Napoli 'Federico II'

Laurea in Informatica

Docenti: Prof. A. Murano, Dr. F. Mogavero

18 febbraio 2011

Nome e Cognome: ______________________ Matricola: ______________________

Spazio riservato alla correzione.

Domanda	1	2	3	4	Totale
Punteggio	4	8	10	8	30
Voto					

Domanda	5	6	Totale
Punteggio	16	14	30
Voto			

(4) 1. Dare un esempio di un linguaggio per (se non é possibile dire formalmente perché):

(a) in PTIME ma non regolare:
Soluzione: $\{0^n1^n \mid n > 0\}$.

(b) in NPTIME ma non NPTIME-COMPLETE:
Soluzione: $\emptyset$. Altrimenti PTIME = NPTIME.

(c) decidibile ma non in NPTIME:
Soluzione: Ogni linguaggio EXPTIME-COMPLETE o superiore.

(d) Turing riconoscibile ma non decidable:
Soluzione: Esistono moltissimi problemi con questa caratteristica. Per esempio, A_{TM} é uno di questi.

(8) 2. Rispondere formalmente alle seguenti domande:

(a) Si supponga che si sia formalmente provato che $P \neq NP$. Cosa si può dire circa il tempo richiesto per risolvere il problema del VERTEX-COVER?
Soluzione: Se PTIME e NPTIME sono differenti, allora non puøesistere un algoritmo polinomiale che risolve il problema del VERTEX-COVER, visto che quest'ultimo é NPTIME-COMPLETE.

(b) Si mostri che A_{TM} non è mapping riducibile a E_{TM}.
Soluzione: Per la prova si veda l'ultima slide della lezione 10: mapping reducibility.

(c) É vero che ogni linguaggio co-riconoscibile non puó essere decidibile?
Soluzione: Falso. Ogni linguaggio decidibile é per definizione sia riconoscibile che co-riconoscibile. Quindi quelli che oltre ad essere co-riconoscibili sono anche riconoscibili sono sicuramente decidibili.

(d) Si mostri che se un problema é NPTIME-COMPLETE, allora il suo complemento é CONPTIME-COMPLETE.
Soluzione: Sia X un problema NPTIME-COMPLETEe Y il suo complemento. Siccome X é in NPTIME, per definizione di CONPTIME, Y é in CONPTIME. Sia L un generico problema in CONPTIME, e M il suo complemento. Per definizione, M é in NPTIME. Siccome X é NPTIME-COMPLETE, ogni problem in NPTIME si riduce ad X in tempo polinomiale. Allo stesso modo, dato che M é in NPTIME, esiste una funzione computabile in tempo polinomiale f tale che $x \in M$ se e solo se $f(x) \in X$. Ma siccome L é il complemento di M e Y é il complemento di X, allora ne consegue che $x \notin L$ se e solo se $f(x) \notin Y$. Ovviamente possiamo negare entrambi i lati del se e solo se e ottenere $x \in L$ se e solo se $f(x) \in Y$. Dunque, la funzione f si riduce da L é permette di ridurre L a Y. Siccome L é un problema CONPTIME arbitrario, ogni problema in CONPTIME si riduce a Y in tempo polinomiale.

(10) 3. (a) Si consideri il *Post correspondence problem* (PCP) con la seguente semplificazione (SPCP). Una istanza P di LPCP é $P = \{\frac{t_1}{b_1}, \frac{t_2}{b_2}, \ldots, \frac{t_k}{b_k}\}$ dove $t_i, b_i \in \Sigma^*$ e $\|t_i\| = \|b_i\|$, per ogni $1 \leq i \leq k$. In altre parole, in ogni domino, la parte superiore ha la stessa lunghezza della parte inferiore. Questa variante del PCP é ancora indecidibile?
Soluzione: il problema SPCP é decidibile e questo segue dalla seguente osservazione: una istanza di SPCP ha un match se e solo se esiste un domino $\frac{t_i}{b_i}$ tale che $t_i = b_i$. Infatti (–>) se una istanza di SPCP ha un match questa deve cominciare con un domino. Siccome la lunghezza del top e del bottom coincide in tutti i domino, il primo domino deve necessariamente avere un top e un bottom coincidente. Di contro, («–) se c'é un domino con lo stesso top e bottom, allora il singolo domino forma un match banale per SPCP. Dunque, per dimostrare che SPCP é decidibile, basta mostrare che é decidibile il verificare che nella collezione dei domino effettivamente ne esista uno che abbia top e bottom uguali.

(b) Si consideri il *Binary Post correspondence problem* (BPCP) variante del PCP e cosí definito. Una istanza P di BPCP is $P = \{\frac{t_1}{b_1}, \frac{t_2}{b_2}, \ldots, \frac{t_k}{b_k}\}$, dove $\{0,1\}^* \in \Sigma^*$, per ogni $1 \leq i \leq k$. In altre parole, ogni istanza del BPCP è come una istanza del PCP, dove il top e il bottom é codificato in binario. Si mostri che questa variante del PCP é indecidibile usando una riduzione dal PCP classico.
Soluzione: Per contraddizione, si assuma che esista un decisore R per BPCP. Costruiamo da questo un

decisore S per PCP come segue. Ogni istanza di P é creata su un alfabeto $\Sigma = \{a_1, a_2, \ldots, a_n\}$ ovvero ogni domino di P avrá come top e bottom una stinga di Σ. Il decisore R su P costruisce una nuova istanza P' di BPCP tale che P ha un match se e solo se P' ha un match. Per produrre P' da P, operiamo come segue: sostituiamo ogni a_i in P con la sequenza di simboli $011\ldots1$, con i ripetizioni di 1. Quindi, per esempio, al domino in P con top a_2a_3, corrisponde il domino in P' con top 0110111 in P'. L'idea é che lo zero funzioni da separatore e la sequenza di uno rappresenti la codifica unaria dell'indice del simbolo a_i. É facile adesso mostrare che P ha un match se e solo se P' ha un match (la prova qui é omessa ma va inserita in fase di verifica). Siccome abbiamo assunto che esiste un decisore R per BPCP, allora possiamo usare R su P' cosí che ci dirá se P ha un match oppure no. Un decisore S per PCP semplicemente prenderá la risposta sul match per P da R. Dunque abbiamo costruito un decisore S per il PCP ma noi sappiamo che un tale decisore per PCP non puó esistere perché PCP é indecidibile, dunque l'assurdo.

(8) 4. (a) Sia U un insieme di taglia n e $S = \{S_1, \ldots, S_m\}$ una collezione di sottoinsiemi di U tale che $\cup_{1 \le i \le m} S_i = U$. Un set cover di taglia k di U é un sottoinsieme di indici $C = \{1, \ldots, m\}$ di taglia k, tale che $\cup_{i \in C} S_i = U$. Sia SET-COVER = $\{(U, S, k)$ tale che U ha un set cover di taglia $k\}$. Si provi che SET-COVER é NP-completo.
(Hint: si usi una riduzione da VERTEX-COVER)
Idea della riduzione per l'hardness: data una istanza (G, k) di VERTEX-COVER si costruisce U come l'insieme degli archi di G e per ogni nodo di G un insieme S_i formato dagli archi ad esso adiacenti.

5. Si consideri l'automa alternante di Co-Büchi $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_0, \mathrm{F} \rangle$, dove $\Sigma = \{0,1\}$, $\mathrm{Q} = \{q_0, q_1\}$, $\mathrm{F} = \{q_0\}$ e la funzione di transizione δ è così definita: $\delta(q_0, 0) = q_0$, $\delta(q_0, 1) = q_1$, $\delta(q_1, 0) = \mathsf{t}$ (true), e $\delta(q_1, 1) = q_0 \wedge q_1$.

(5) (a) Determinare l'espressione ω-regolare del linguaggio accettato da $\mathcal{A}$.

(2) (b) Determinare l'automa alternante $\mathcal{A}'$ duale di $\mathcal{A}$.

(6) (c) Costruire l'automa nondeterministico di Büchi $\mathcal{N} = \langle \Sigma, \mathrm{Q}', \delta', Q_0', \mathrm{F}' \rangle$ equivalente ad $\mathcal{A}'$ eseguendo la procedura di nondeterminizzazione (breakpoint construction, anche detta costruzione di 'Miyano-Hayashi').

(3) (d) Sia $\mathcal{N}' = \langle \Sigma, \mathrm{Q}', \delta', Q_0', \mathrm{F}' \rangle$ l'automa nondeterministico di Co-Büchi topologicamente uguale ad $\mathcal{N}$ (la differenza tra $\mathcal{N}$ e $\mathcal{N}'$ risiede unicamente nella semantica dell'accettazione). Determinare l'espressione ω-regolare del linguaggio accettato da $\mathcal{N}'$.

6. Si consideri la formula LTL $\psi = p \wedge \mathsf{G}\,(p \rightarrow \mathsf{X}\,\mathsf{X}\,p)$ sull'insieme di proposizioni atomiche $\mathrm{AP} = \{p, q\}$.

(4) (a) Determinare l'espressione ω-regolare del linguaggio $\mathrm{L}(\psi)$ sull'alfabeto $\Sigma = 2^{\mathrm{AP}}$.

(4) (b) Costruire l'automa alternante di Büchi $\mathcal{A}$ con $\mathrm{L}(\mathcal{A}) = \mathrm{L}(\psi)$.

(6) (c) Costruire l'automa nondeterministico generalizzato di Büchi $\mathcal{N}$ con $\mathrm{L}(\mathcal{N}) = \mathrm{L}(\psi)$.