Cognome .... Nome ....Matricola ....

Quinta prova intermedia di Informatica Teorica – 25/01/2019 – Tempo a disposizione 60 minuti.
Regole del gioco: Libri e quaderni chiusi, vietato scambiare informazioni con altri; indicare su tutti i fogli, con chiarezza, nome e numero di matricola; *consegnare solo i fogli con le domande (questi)*.

Esercizio 1. Una rete sociale, modellata come un grafo G=(V,E), è disponibile su un repository remoto consultabile dal sistema locale tramite query online. Il problema SixDegreesOfSeparation(G,u,v) consiste nel determinare se esiste un cammino in G tra i due nodi u e v di lunghezza al massimo 6.

1.1 Mostra come il problema SixDegreesOfSeparation(G,u,v) può essere risolto in spazio logaritmico da un sistema locale non deterministico.

Il problema è molto simile a STCONN. Per risolverlo con una MTND in spazio logaritmico è sufficiente un puntatore al nodo corrente ed un contatore binario del numero di passi. Ad ogni passo mi muovo non deterministicamente su un vicino del nodo corrente incrementando il contatore. Se il contatore è > 6 rifiuto. Se raggiungo v accetto.

1.2 Quanto spazio sarebbe necessario, invece, se il sistema locale fosse deterministico? Quel è il miglior limite superiore asintotico che sapresti proporre e perché?

Poiché s(n) >= log(n) posso applicare il teorema di Savitch esteso: su una MT deterministica è risolvibile in spazio $\log^2(n)$

Esercizio 2. A quali delle tre classi PSPACE, NL o L appartengono i seguenti linguaggi e perché?

2.1 Linguaggio delle stringhe palindrome sull'alfabeto {a,b}.	$\in$ L perché sono sufficienti due puntatori all'inizio e alla fine della stringa.

2.2 Linguaggio <G,k> dove G è un grafo che contiene una clique di dimensione almeno k.	$\in$ PSPACE, perché il problema è chiaramente in NP ed NP $\subseteq$ PSPACE. (le proposte di soluzione che memorizzano un sottoinsieme di dimensione k non dimostrano l'appartenenza ad NL perché k $\in$ O(n))

2.3 Linguaggio L= $\{a^i b^j c^k d^h$ dove i=0 oppure j=k=h}	$\in$ L perché sono sufficienti quattro contatori.

2.4 Linguaggio <G> dove G è un grafo connesso (esiste un cammino da un nodo a tutti gli altri nodi).	$\in$ NL. E' sufficiente applicare STCONN iterativamente tra il primo nodo e tutti gli altri, uno alla volta.

Cognome .... Nome ....Matricola ....

<u>Esercizio 3</u> – Ricorda che i problemi SAT a TQBF sono definiti come segue:

Istanza SAT: una formula booleana in forma normale congiuntiva. Es: $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee \neg x_4)$

Predicato SAT: esiste un'assegnazione di verità alle variabili per cui la formula è vera.

Istanza TQBF: una formula booleana generica preceduta da un quantificatore esistenziale o universale per ogni variabile. Es: $\forall x_1\, \exists x_2 \forall x_3\, \exists x_4$: $(\neg x_1 \vee (x_4 \wedge x_2)) \vee ((\neg x_2 \wedge x_3) \vee \neg x_4)$

Predicato TQBF: la formula booleana è vera.

3.1 Mostra una riduzione polinomiale da SAT a TQBF.

Data un'istanza di SAT è sufficiente premettere $\exists x_1 \exists x_2\, \exists x_3$ …. per ottenere la corrispondente formula TQBF.

3.2 Dimostra che le istanze Yes di SAT vengono trasformate in istanze Yes di TQBF.

Banale: se l'istanza SAT di partenza è un'istanza Yes, allora esiste un'assegnazione alla variabili x_1, x_2, x_3, …. che soddisfa la formula SAT, ma questo è esattamente ciò che è richiesto dalla formula TQBF.

3.3 Dimostra che le istanze No di SAT vengono trasformate in istanze No di TQBF.

Se l'istanza TQBF è un'istanza Yes, vuol dire che esistono valori di verità per le variabili $\exists x_1 \exists x_2\, \exists x_3$ … tali che la formula è soddisfatta, ma questo implica che quei valori di verità soddisfano SAT, che dunque è un'istanza Yes.

<u>Esercizio n</u>. Se non rispondi a questo esercizio o rispondi in modo errato non ottieni nessuna penalizzazione. Se rispondi in modo corretto allora ottieni un **+1** sul voto complessivo del compito.

Considera il linguaggio L={<M>| M è una MT che, con ogni possibile input, scrive una "a" sul nastro nei primi 10 passi della sua computazione}. Il linguaggio L è decidibile o no?

Il linguaggio è decidibile. Basta cosruire una macchina M' che simula M su ogni possibile input di lunghezza 10 e verifica che M scriva (o non scriva) una "a" sul nastro. Siccome nei primi 10 passi M può esplorare solo 10 caselle del nastro non è necessario provare M su tutti i possibili input.