Informatica Teorica

Seconda prova in itinere - 3 Luglio 2007

Avvisi importanti

- Il tempo disponibile per lo svolgimento della prova è **1 ora e 30 minuti**.
- Gli studenti laureandi **devono indicare esplicitamente la loro situazione nella testata del loro elaborato**.
- Come in altre occasioni i risultati e le successive comunicazioni (in particolare le modalità di accettazione/rifiuto del voto proposto) saranno pubblicati nella pagina personale del docente sul sito ufficiale del Dipartimento. Verrà data precedenza alla correzione dei compiti dei laureandi (che risultino tali dall'archivio ufficiale del CEDA)

Esercizio 1 (punti 4/17)

La successione $2^{2^n}+1$, per *n* numero naturale, è detta dei *numeri di Fermat*. Il matematico Pierre de Fermat fu il primo a notare come i primi termini della successione (per n=0, 1, 2, 3) siano primi, dunque congetturò che lo siano tutti.

Si risponda alle seguenti domande, fornendo delle brevi, ma esaurienti, spiegazioni:

1.1. E' decidibile il problema "tutti i numeri di Fermat sono primi?"?

1.2. E' decidibile il problema "dato un numero naturale *n*, il numero $2^{2^n}+1$ è primo?"?

1.3. (nel caso si sia risposto positivamente alla domanda 1.2) E' decidibile l'insieme delle macchine di Turing che risolve il problema della domanda 1.2?

Esercizio 2 (punti 6/17)

2.1

Si descriva un algoritmo che, dato un numero naturale *n*, determini se l'n-esimo numero di Fermat è primo.

Si noti che:

- non è necessario che l'algoritmo sia efficiente o elegante, l'importante è che sia semplice;
- per descrivere l'algoritmo è possibile usare pseudocodice oppure un linguaggio ad alto livello.

2.2.

Se l'algoritmo definito al punto 2.1 viene eseguito su una macchina RAM, quale è il criterio di costo più opportuno per valutare la complessità di tale macchina RAM? Si motivi la risposta.

2.3.

Si usi il criterio individuato al punto 2.2 per valutare la complessità spaziale e temporale di una macchina RAM che esegue l'algoritmo definito.

Esercizio 3 (punti 9/17)

Si vuole usare la logica del prim'ordine (con il predicato di uguaglianza) per descrivere giocatori e squadre di uno sport a squadre (per esempio il calcio).

Si assuma quanto segue:

- le variabili (x, y, w, ...) denotano giocatori;
- viene introdotto il predicato SameTeam(x,y) il quale è vero se i giocatori x e y sono nella stessa squadra (laddove "essere nella stessa squadra" è una relazione di equivalenza).

3.1.

Si scrivano, usando *esclusivamente* i predicati di uguaglianza (=), disuguaglianza ($\neq$) e SameTeam, delle formule logiche che formalizzano i seguenti vincoli:

a) tutte le squadre hanno almeno 3 giocatori;

b) esistono almeno 3 squadre.

3.2.

Si consideri una relazione di "rivalità personale" tra alcuni dei giocatori. A questo scopo si introduca il predicato PersRivalry(x,y), il quale rappresenta il fatto che i giocatori x e y sono rivali (si assuma pure che PersRivalry sia simmetrica).

Si scriva una formula che formalizzi il fatto che non ci possono essere rivalità all'interno di una stessa squadra.

3.3.

Si assuma che periodicamente (per esempio ogni mese) si svolga un torneo tra le squadre, e che sia le squadre che le rivalità personali possano cambiare da un torneo all'altro.

Si assuma inoltre quanto segue:

- le variabili t, t', t'' denotano i momenti di svolgimento dei tornei, rappresentati come numeri interi.
- per descrivere squadre e rivalità si usano delle varianti temporizzate dei predicati introdotti in precedenza, SameTeam(x,y,t) e PersRivalry(x,y,t), che indicano appartenenza alla stessa squadra / rivalità di x e y al momento del torneo t.

Si scrivano delle formule che formalizzino i seguenti vincoli:

a) due giocatori sono nella stessa squadra durante un torneo solo se non erano rivali in alcuno dei precedenti k tornei;

b) se, per un certo torneo, due giocatori della stessa squadra sono rivali, essi non saranno nella stessa squadra per i prossimi h tornei.

Soluzioni

Esercizio 1

1.1. Sì, si tratta di una domanda “chiusa” (la risposta è sì oppure no). Tra l’altro si conosce anche la risposta: Eulero dimostrò che per n=5 si ottiene un numero non primo.

1.2. Sì, basta scrivere una semplice procedura che, dato n, calcoli $2^{2^n}+1$ e poi ne controlli la primalità.

1.3. No, è una conseguenza del teorema di Rice (l'insieme delle MT che risolvono il problema 1.2 non è l'insieme vuoto, né tantomeno quello universo).

Esercizio 2

2.1

```
boolean primeNthFermat(int n){
1. x = 2
2. for i from 1 to n do x = x*x
3. x = x+1
4. for i from 2 to squareRootOf(x) do
5.   if (x/i)*i==x then return false
6. return true
```

2.2-2.3.

Per il calcolo del numero di Fermat (righe 1-3) è necessario usare il criterio di costo logaritmico, per ottenere una valutazione di complessità realistica. Quindi si ottiene:

$\sum_{k=1, \dots, n} 2^k = \Theta(2^n)$ per il tempo e $\Theta(2^n)$ anche per lo spazio.

Il fattore di complessità dominante è però il secondo ciclo (righe 4-5): la complessità temporale che si ottiene per esso (naturalmente sempre a costo logaritmico) è:

$\Theta(\boxed{2^n \cdot \sqrt{2^{2^n}}}) = \Theta(2^n \cdot 2^{2^{n-1}}) = \Theta(2^{2^{n-1}+n})$.

Infatti la singola iterazione costa log (2^{2n}) ed il calcolo della radice quadrata non impatta sul comportamento asintotico della complessità.

Esercizio 3

3.1.

a) $\forall x \exists y \exists z(x \neq y \wedge y \neq z \wedge x \neq z \wedge SameTeam(x,y) \wedge SameTeam(y,z))$

b) $\exists x \exists y \exists z(x \neq y \wedge y \neq z \wedge x \neq z \wedge \neg SameTeam(x,y) \wedge \neg SameTeam(y,z) \wedge \neg SameTeam(x,z))$

3.2.

$\forall x \forall y(x \neq y \wedge SameTeam(x,y) \rightarrow \neg PersRivalry(x,y))$

3.3.

a) $\forall x \forall y \forall t(SameTeam(x,y,t) \rightarrow \forall t'(t-k \leq t' < t \rightarrow \neg PersRivalry(x,y,t')))$

b) $\forall x \forall y \forall t(PersRivalry(x,y,t) \wedge SameTeam(x,y,t) \rightarrow \forall t'(t < t' \leq t+h \rightarrow \neg SameTeam(x,y,t')))$