Algoritmi e Principi dell'Informatica
Prima Prova in Itinere (modulo Informatica teorica)

23 Novembre 2015

Il tempo a disposizione è di **1 ora e 45 minuti.**

Esercizio 1 (punti 7 senza la parte c, 8 con la parte c)

Parte a.

Si consideri il gioco della roulette. Un giocatore adotta la seguente strategia:

- inizialmente punta 1€ sul rosso;
- a ciascuna estrazione successiva punterà <u>sempre</u> sul rosso;
- in particolare, se esce il nero, il giocatore perde la posta e al prossimo turno raddoppierà la puntata; altrimenti vince il doppio di quanto ha puntato e al prossimo turno punterà tutta la vincita.

Modellare il comportamento del giocatore con un automa a potenza minima (tra le classi FSA, DPDA, NDPDA e TM) che riceva in ingresso una stringa costruita sull'alfabeto {R,N} (dove R indica che è uscito il rosso e N che è uscito il nero) e che la accetti se e solo se il giocatore che abbia puntato secondo la strategia indicata nelle estrazioni corrispondenti ai simboli in ingresso è in attivo (ossia le vincite superano gli importi complessivamente puntati).

Parte b.

Qual è il linguaggio accettato da tale automa?

Parte c.

Cambierebbe la classe dell'automa richiesto se la strategia fosse modificata in modo che, quando esce il rosso, al turno successivo il giocatore punti solo 1€?

Esercizio 2 (Punti 5)

Il predicato binario *succ* indica che i suoi argomenti sono numeri naturali tali per cui il secondo è il successore del primo. Ad esempio, *succ*(1,2) è vero mentre *succ*(2,6) è falso.

Si specifichi in logica del prim'ordine il predicato ternario *somma*, che indica che il terzo argomento è la somma dei primi due. Ad esempio, *somma*(2,3,5) è vero mentre *somma*(3,6,2) è falso.

Nella specifica del predicato *somma* non si può fare uso di predicati diversi da *succ* e dal predicato di uguaglianza. Inoltre, non si possono usare funzioni aritmetiche (ad esempio il +…). E' invece consentito l'uso di costanti che indicano numeri naturali (0, 1, 2, …).

Esercizio 3 (punti 5)

Si consideri il seguente programma P codificato in uno pseudolinguaggio ispirato al C.

P:

```
{       int x, y, z;
        read (x, y, z);
        while (x != y) {
                x = x - y;
                z = z + y};
        write z; }
```

Sia f(x, y, z) la funzione calcolata da P. Si dica, giustificando brevemente la risposta, se la seguente funzione:

$g(x, y, z) = 1$ se $f(x,y,z) \neq \perp$, 0 altrimenti

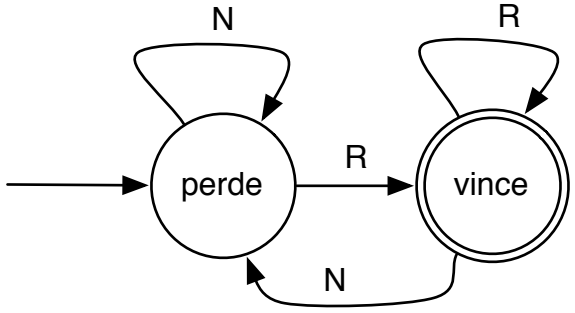
è calcolabile o no.

Tracce di soluzioni

Esercizio 1

Parte a.

La strategia delle puntate è fatta in modo tale che, se il giocatore vince, vince necessariamente almeno 1€ in più di quanto può aver perso complessivamente in precedenza; se perde, perde necessariamente almeno 1€ in più di quanto ha vinto in precedenza. E' sufficiente pertanto un semplice FSA per modellare questo comportamento.



Parte b.

Il linguaggio riconosciuto dall'automa è composto da tutte e sole le stringhe che terminano con una R.

Parte c.

Sì, occorre un formalismo più potente. In questo caso non c'è più la garanzia di essere in perdita dopo l'uscita del nero, in quanto ci possono essere state vincite precedenti che compensano largamente la perdita. Occorrerebbe pertanto tenere conto sia di quanti euro sono stati puntati in precedenza, sia dell'entità della puntata corrente, o comunque di poter ricostruire il bilancio attuale del giocatore (negativo o positivo e arbitrariamente grande), il che eccede il potere espressivo di un automa a stati finiti.

Esercizio 2

$\forall x \forall y \forall z$ (somma(x,y,z) $\leftrightarrow$ ((y=0 $\wedge$ x=z) $\vee$ ($\exists y' \exists z'$ succ (y',y) $\wedge$ succ (z',z) $\wedge$ somma(x,y',z'))))

Esercizio 3

L'esecuzione di P termina se e solo se i dati forniti in ingresso sono tali che x e y siano dello stesso segno, x sia $\geq$ y se entrambi positivi o viceversa se entrambi negativi, e sia, in modulo, un suo multiplo; più precisamente:

g(x, y, z) = 1 se

$x = y$

$\vee$

$((x > y > 0 \vee x < y < 0) \wedge \exists k\ (k > 1 \wedge |x| = k.|y|))$

0 altrimenti

ed è evidentemente calcolabile.