Prova scritta dell'esame di ammissione al Dottorato di Ricerca in Informatica - Università di Pisa 10 Novembre 2005

1 Il candidato svolga due a scelta tra i tre esercizi seguenti.

Esercizio 1

Dati $n>3$ filosofi (numerati da 1 ad n) si consideri la seguente soluzione del problema dei filosofi a cena:

```
Semaforo bacchetta[n];

Filosofo i (con i pari):
Loop {
      pensa
      wait( bacchetta[i] )        // prende la bacchetta alla sua destra
      wait( bacchetta[((i+1) mod n) +1] ) // prende la bacchetta alla sua sinistra
      mangia
      signal( bacchetta[((i+1) mod n) +1] ) // rilascia la bacchetta alla sua sinistra
      signal( bacchetta[i] ) // rilascia la bacchetta alla sua destra
}

Filosofo j (con j dispari):
Loop {
      pensa
      wait( bacchetta[((j+1) mod n) +1] ) // prende la bacchetta alla sua sinistra
      wait( bacchetta[j] ) // prende la bacchetta alla sua destra
      mangia
      signal( bacchetta[j] ) // rilascia la bacchetta alla sua destra
      signal( bacchetta[((j+1) mod n) +1] ) // rilascia la bacchetta alla sua sinistra
}
```

Dire (motivando la risposta) se:

1) nella soluzione proposta si può verificare il deadlock
2) nella soluzione proposta si può verificare l'attesa indefinita
3) la soluzione è ottima dal punto di vista del parallelismo (numero di filosofi che possono mangiare contemporaneamente)

Esercizio 2

Dato l'alfabeto A = {a,b,c}, il candidato fornisca:

- un linguaggio libero (context-free) non regolare su A*
- un linguaggio dipendente dal contesto e non libero su A*

motivando formalmente le risposte.

Esercizio n. 3

Si consideri un semplice linguaggio imperativo con la seguente sintassi.

```
Prog ::=      program ConstVarDecl ; ProcDecl ; Com.

ConstVarDecl ::=  null | const Ide = ConstValue | var Ide : Type |
                  ConstVarDecl ; ConstVarDecl

Type ::=      int | bool

ProcDecl ::= procedure Ide (Par) ConstVarDecl ; Com
             | ProcDecl ; ProcDecl

Par ::=       Type Ide

Com ::=       Ide := Exp
              | Com ; Com
              | if Exp then Com else Com endif
              | while Exp do Com endw
              | read (Ide) | write (Exp)
              | Ide (Exp)
```

Le categorie sintattiche `Ide`, `Exp` e `ConstValue` vengono omesse per semplicità e hanno l'ovvio significato. L'unica modalità di passaggio dei parametri prevista è quella *per valore*. Inoltre le procedure possono essere ricorsive e possono far riferimento a variabili globali.
Si definisca la semantica formale della dichiarazione di procedura e della chiamata di procedura, assumendo data la semantica degli altri costrutti, sia nel caso di *scoping statico* che nel caso di *scoping dinamico*. Si dia un esempio di programma la cui semantica di input/output differisce nei due casi, motivando la risposta.
Si ricorda che, nel regime di scoping statico, l'ambiente globale utilizzato al momento della chiamata di procedura è quello relativo alla dichiarazione della procedura stessa. In regime di scoping dinamico, invece, l'ambiente globale è quello presente al momento della chiamata.

2 Il candidato descriva un'area di ricerca innovativa in informatica, inquadrandola nel settore di riferimento e delinenandone i possibili sviluppi futuri che ritiene più promettenti.