



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali  
Corso di Laurea in Informatica

Tesi di Laurea

TITOLO ITALIANO

TITOLO INGLESE

NOME CANDIDATO

Relatore: *Relatore*  
Correlatore: *Correlatore*

Anno Accademico 2014-2015

Nome candidato: *Titolo italiano*, Corso di Laurea in Informatica, © Anno  
Accademico 2014-2015

---

## INDICE

---



---

## ELENCO DELLE FIGURE

---



*"Inserire citazione"*  
— *Inserire autore citazione*





---

## ESERCIZI

---

1. Scrivere le possibili evoluzioni del programma

`co X: = X+2 // X: = X+1 oc`

assumendo che ciascun assegnamento è realizzato da tre azioni atomiche che caricano X in un registro (Load R X), incrementano il valore del registro (Add R v) e memorizzano il valore del registro ((Store R X). Per ciascuna delle esecuzioni risultanti dall'interleaving delle azioni atomiche descrivere il contenuto dopo ogni passo della locazione condivisa X e dei registri privati, R<sub>1</sub> del processo che esegue il primo assegnamento ed R<sub>2</sub> per il processo che esegue il secondo assegnamento. Se assuma che il valore iniziale di X sia 50.

2. Si definisca il problema della *barrier synchronization* e si descrivano per sommi capi i differenti approcci alla sua soluzione. Se ne fornisca quindi una soluzione dettagliata utilizzando i semafori.
3. Considerare n api ed un orso che possono avere accesso ad una tazza di miele inizialmente vuota e con una capacità di k porzioni. L'orso dorme finchè la tazza è piena di k-porzioni, quindi mangia tutto il miele e si rimette a dormire. Le api riforniscono in continuazione la tazza con una porzione di miele finchè non si riempie; l'ape che aggiunge la k-esima porzione sveglia l'orso. Fornire una soluzione al problema modellando orso ed api come processi e utilizzando un monitor per gestire le loro operazioni sulla tazza. Prevedere che le api possano eseguire l'operazione *produce-honey* anche concorrentemente.
4. Descrivere le primitive di scambio messaggi send e receive sia sincrone che asincrone ed implementare

## 8 ESERCIZI

- `synch_send(v:int)`
- `send(v:int)`
- `receive(x:int)`

utilizzando le primitive di LINDA.

---

## BIBLIOGRAFIA

---

[1] Autore - *titolo*

[2] Autore - *Titolo* - altre informazioni