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

Tesi di Laurea

ESTENSIONE DEL LINGUAGGIO FACPL PER ESPRIMERE POLITICHE DI ACCESSO ALLE RISORSE DI UN SISTEMA DI CALCOLO BASATE SUL COMPORTAMENTO PASSATO

EXTENTION OF LANGUAGE FACPL TO USE ACCESS CONTROL POLICIES BASED ON THE PAST ACTIONS

FILIPPO MAMELI

Relatore: *Relatore*
Correlatore: *Correlatore*

Anno Accademico 2015-2016

INDICE

ELENCO DELLE FIGURE

"Inserire citazione"
— Inserire autore citazione

1

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 dallÕinterleaving delle azioni atomiche descrivere il contenuto dopo ogni passo della locazione condivisa X e dei registri privati, R_1 del processo che esegue il primo assegnamento ed R_2 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

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

utilizzando le primitive di LINDA.

BIBLIOGRAFIA

[1] Autore - *titolo*

[2] Autore - *Titolo* - altre informazioni