Modelli Computazionali per Sistemi Complessi

Corso di Laurea in Informatica
Facoltà di Scienze Matematiche Fisiche e Naturali
Università della Calabria

Esercitazione 1 – aggiornato al 2008-05-08 13:00

Argomenti fatti a lezione:

1. Discussione sull'algoritmo di minimizzazione LINK AI LUCIDI MOSTRATI IN LABORATORIO
2. Progetto Camelot Lava

Argomenti suggeriti per casa: LINK AL MATERIALE ONLINE

3. Progetto Camelot Life
4. Progetto Camelot Minimizzazione

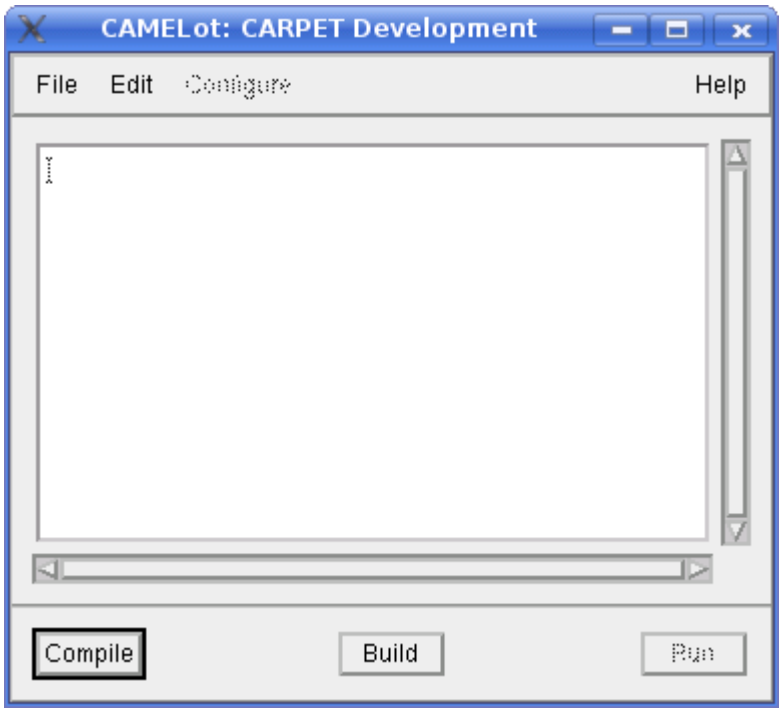
---

1. Introduzione Camelot

Dopo essersi loggati nel sistema operativo linux di un pc del laboratorio aprire una shell e scrivere il comando

```
camelot &
```

questo ci mosta l'ambiente camelot



questo ambiente se pur molto potente ha dei bug nell'editor integrato quindi per scrivere i progetti in seguito useremo *gedit* oppure *kwrite*

e poi useremo la voce FILE-OPEN del menu di camelot come mostrato nella figura 1 per caricare il progetto scritto. Per il momento chiudiamo camelot: FILE-EXIT.

---

2. Progetto Camelot Mod2

Per prima cosa, usando la shell aperta prima (utilizzabile grazie all'uso della &), creeremo una cartella progetti e dentro questa un'altra cartella dal nome mod2 con il comando

```
mkdir progetti
cd progetti
mkdir mod2
cd mod2
```

a questo punto siamo nella cartella mod2 e dobbiamo scaricare dal sito del dott. Spataro dei files che ci servono con il comando

```
wget http://www.mat.unical.it/spataro/teaching/modelli/istruzioni.txt
wget http://www.mat.unical.it/spataro/teaching/modelli/macrocell.c
wget http://www.mat.unical.it/spataro/teaching/modelli/modelli.cmt
```

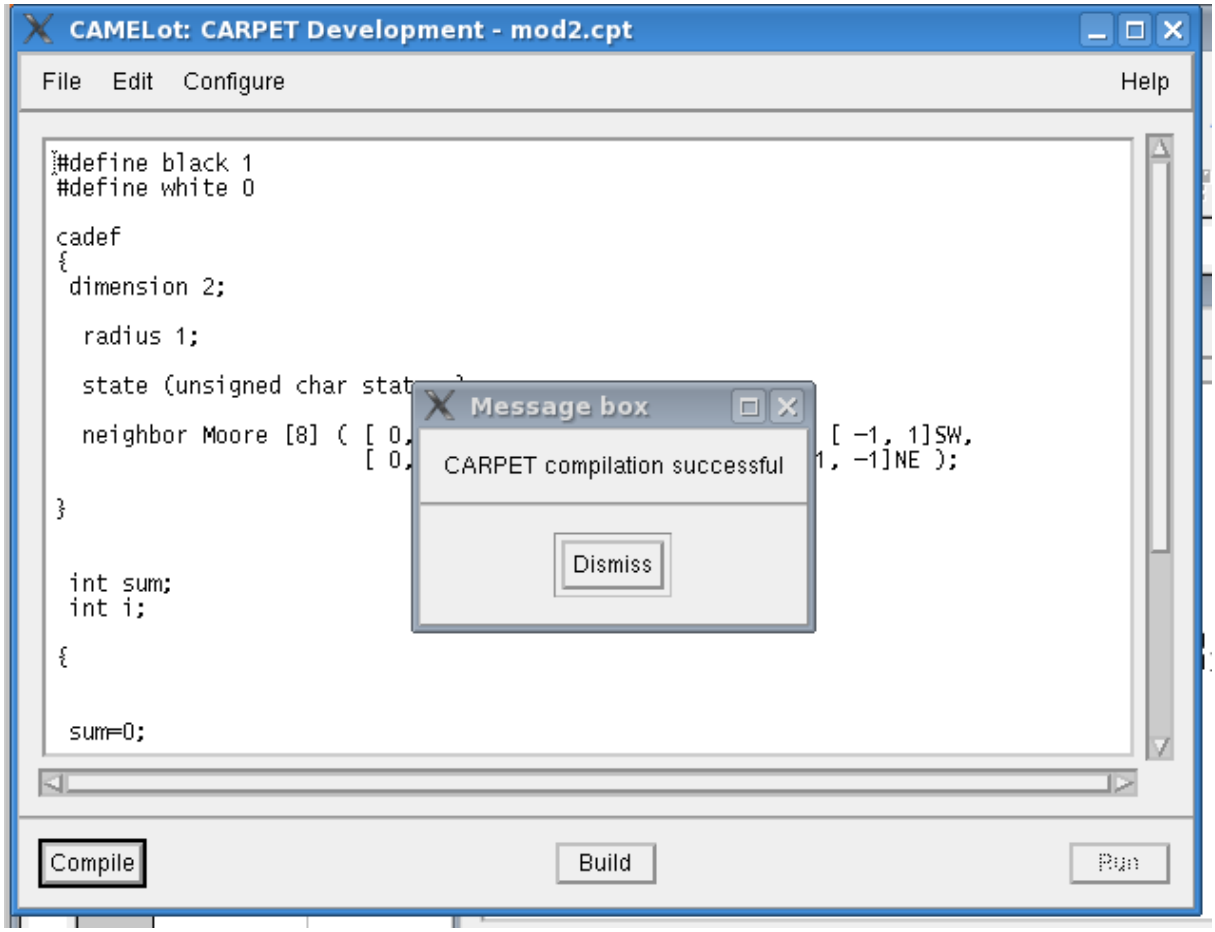
il primo file contiene le istruzioni per settare le variabili interne di camelot, il secondo è un file che va sempre incluso in qualsiasi progetto vogliamo fare, il terzo è una configurazione iniziale per l'unico sottostato che è contenuto in questo progetto.

A questo punto apriamo il nostro editor di testo preferito, scriviamo il programma e lo salviamo col nome *mod2.cpt* nella cartella mod2

A questo punto possiamo aprire il nostro progetto mod2 con camelot con il comando:

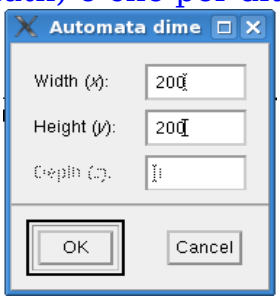
```
camelot mod2.cpt
```

fatto ciò controlliamo che non ci siano errori sintattici nel programma: usiamo il bottone Compile di camelot.



Una schermata del genere ci da l'ok a proseguire.

Fatto ciò il menu CONFIGURE di camelot diverrà attivo e lo utilizzeremo per configurare alcune variabili interne del nostro progetto (nello specifico andremo a indicare la dimensione dell'automa cellulare e i flag con cui compilare il progetto); usando il menu CONFIGURE-AUTOMATA_DIMENSIONS andremo a specificare sia per larghezza (width) e che per altezza (height) il valore 200.



A questo punto salviamo la configurazione fin qui creata: FILE-SAVE_CONFIGURATION...
Così facendo è stato creato un file dal nome mod2.cnf che contiene i le informazioni riguardo il progetto. Ora chiudiamo camelot: FILE_EXIT.

Per evitare di usare la simulazione del terzo tasto del mouse andiamo a modificare a mano il file *mod2.cnf* e lo rendiamo uguale alle seguenti 11 righe:

```
200 200 1
1
1
0
0
$MPIR_ROOT/bin/mpicc
-O -DCPT_INCLUDE_FILE=\"%s\" -DNO_XDR_READ -DNO_XDR_WRITE -I/usr/local/camelot -L/usr/local/camelot/linux -I/usr/local/mpich-1.2.5/include -L/usr/local/mpich-1.2.5/lib
-lm
$MPIR_ROOT/bin/mpirun -np %d
```

il carattere nanoscopico con cui è scritto è dettato dal fatto che se la riga più lunga fosse spalmata su due righe, tutto sarebbe interpretato in modo errato da camelot. Ad ogni modo ecco la versione leggibile:

200 200 1
1
1
0
0
$MPIR_ROOT/bin/mpicc
-O -DCPT_INCLUDE_FILE=\"%s\" -DNO_XDR_READ -DNO_XDR_WRITE -I/usr/local/camelot -L/usr/local/camelot/linux
-I/usr/local/mpich-1.2.5/include -L/usr/local/mpich-1.2.5/lib
-lm
$MPIR_ROOT/bin/mpirun -np %d

è importante che il file sia costituito esattamente da 11 righe e che sia utilizzato per i pc del laboratorio.

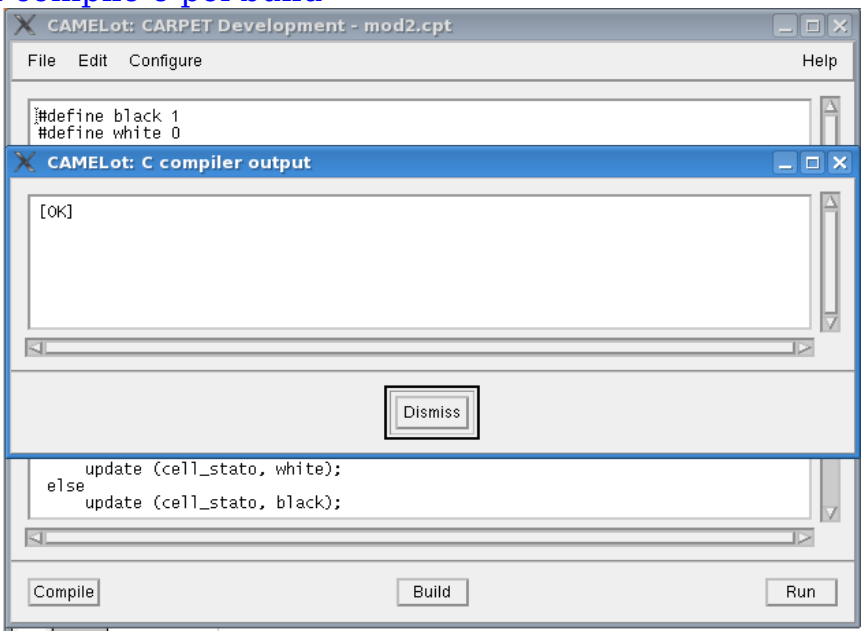
NOTA: SOLO per chi ha installato linux sul proprio portatile seguendo la guida pubblicata sul sito, il file di configurazione di 11 righe è il seguente

```
200 200 1
1
1
0
0
/home/modcomp/usr/local/mpich-1.2.5/bin/mpicc
-O -DCPT_INCLUDE_FILE=\"%s\" -DNO_XDR_READ -DNO_XDR_WRITE -I/home/modcomp/usr/local/camelot -L/home/modcomp/usr/local/camelot/linux -I/home/modcomp/usr/local/mpich-1.2.5/include -L/home/modcomp/usr/local/mpich-1.2.5/lib
-lm
/home/modcomp/usr/local/mpich-1.2.5/bin/mpirun -np %d
```

Fatto ciò lanciamo il nostro progetto in camelot col comando:

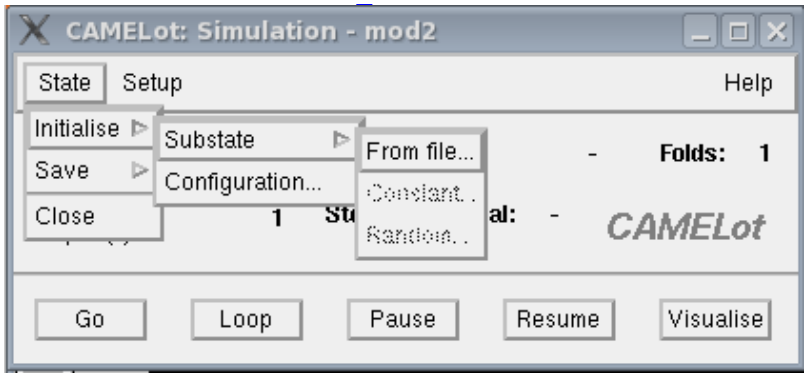
```
camelot mod2.cpt &
```

così facendo il file di configurazione prima editato è caricato in automatico; ora pigiamo prima compile e poi build

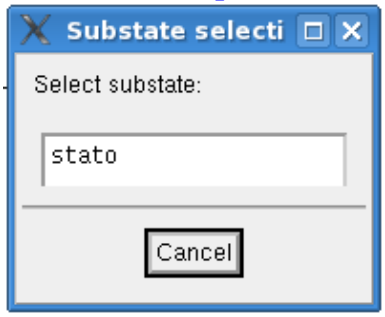


questo è quello che otteniamo con Build se tutto è andato bene (il risultato di Compile è quello di figura 2).

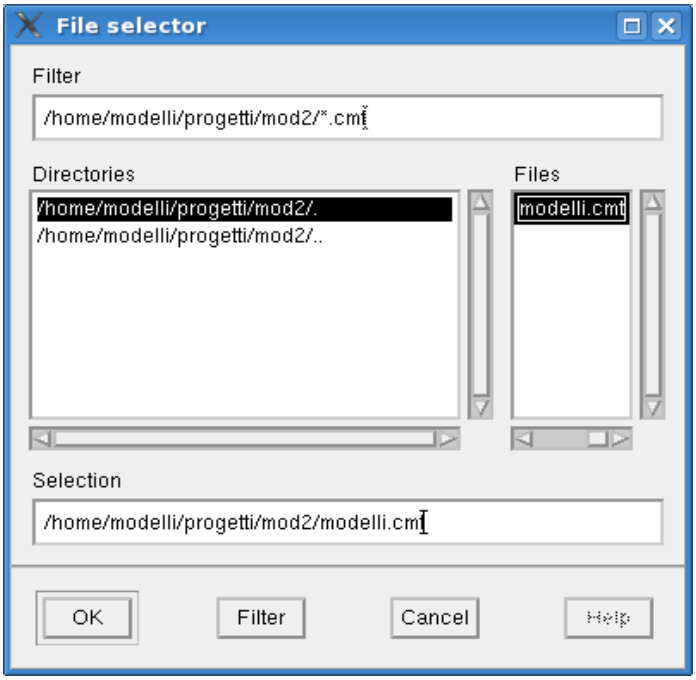
Siamo finalmente pronti per caricare la configurazione iniziale per il sottostato chiamato “stato” e procedere alla simulazione: premiamo Run in camelot e ci appare una nuova maschera, da questa scegliamo il menu:
STATE-INITIALIZE-SUBSTATE-FROM_FILE...



e selezioniamo stato nella finesta che compare;



in fine facciamo doppio click sul file modelli.cmq che compare in una nuova maschera.



Siamo pronti finalmente a visualizzare il nostro progetto!!!
Nella finesta dal titolo

“CAMELot: Simulation – mod2”

scegliamo il bottone Visualise,
scriviamo 1 nel box che ci chiede ogni quanti step visualizzare,

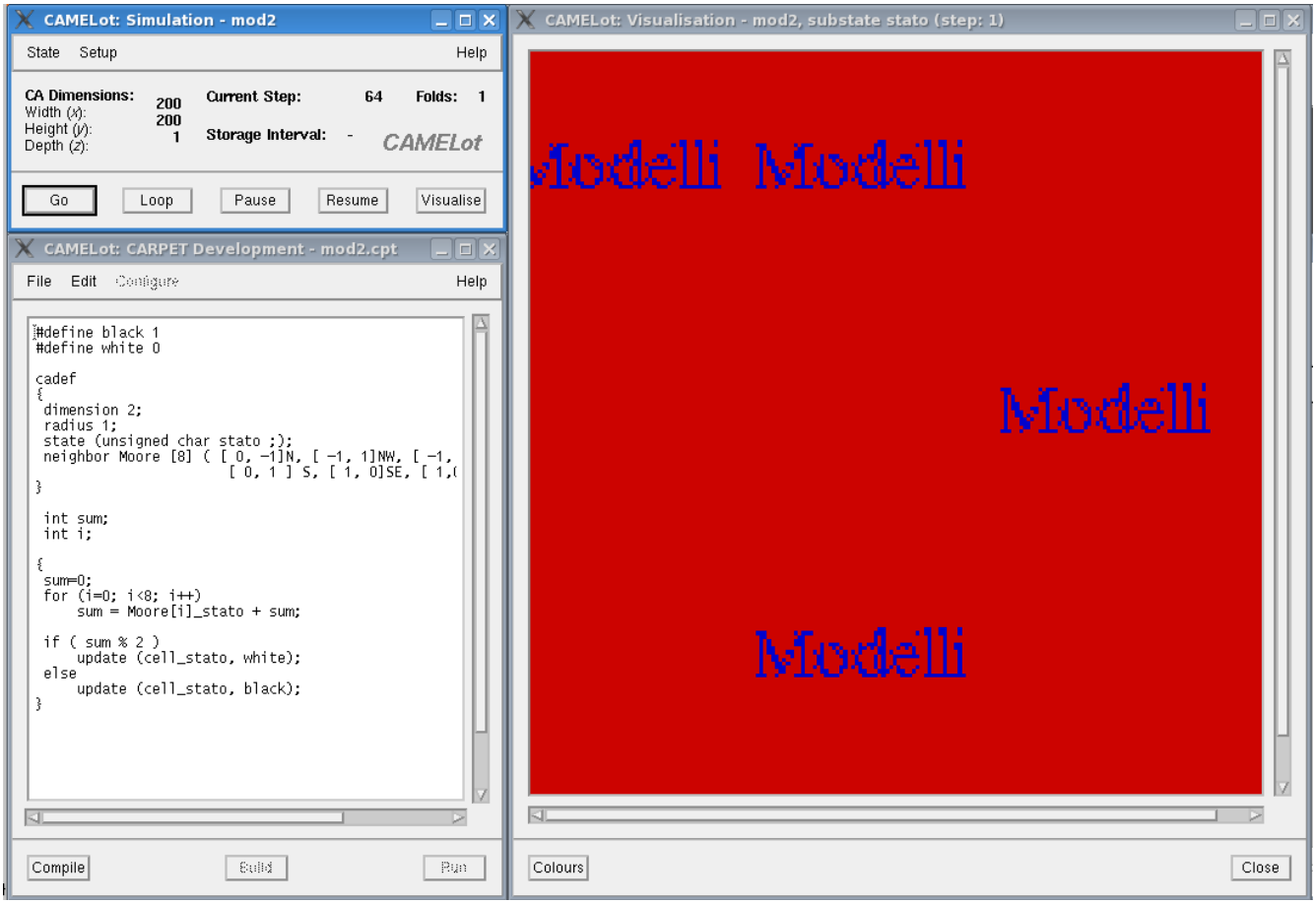
segliamo il sottostato “stato”;

a questo punto affianchiamo le finestre e premendo Go

vedremo su sfondo rosso come il nostro automa cellulare passa

dalla configurare iniziale a delle nuove configurazioni molto particolari...

vedere per credere.



3. Progetto Camelot Life: **pdf_source_codes/life.cpt.pdf**

```
#define alive 1
#define dead 0

cadef
  {
    dimension 2;

    radius 1;

    state ( int life; );

    neighbor Moore[8] ( [ 0,-1]N, [-1,-1]NW, [-1, 0]W, [-1,  1]SW,
                        [ 0, 1]S, [ 1, 1]SE, [ 1, 0]E, [ 1, -1]NE  );
    deterministic;

  }
    register int i;
    register int sum;
  {
    sum = 0;
    for( i=0 ; i<8; i++)
        sum = Moore[i]_life + sum;

    if ( sum == 3  || ( sum == 2 && cell_life == 1) )
        update (cell_life, alive);
    else
        update (cell_life, dead);
  }
```

4. Progetto Camelot Minimizzazione

un esempio di ciò è il programma per la simulazione delle lave etnee realizzato da Gino Mirocle Crisci, Salvatore Di Gregorio, Rocco Rongo, William Spataro e Donato D'Ambrosio nel 2000: **pdf_source_codes/sciara.cpt.pdf**