

# Realizzazione di un sistema semaforico intelligente

Installazione e uso

Andrea Buda

## Installazione

Per prima cosa occorre decomprimere la cartella compressa *progetto.zip* in una directory a propria scelta (*dir*).

Nella directory *bin* sono inserite le librerie *tuProlog* e *TuCSon*, le quali permettono rispettivamente di implementare agenti logici e di creare degli spazi di tuple necessari per la comunicazione e la coordinazione tra i vari agenti.

Per la creazione di un nodo *TuCSon*, occorre aprire una finestra di prompt, poi dalla directory *dir* digitare:

```
java -cp bin\tucson.jar alice.tucson.runtime.Node
```

Per la creazione di un ispettore *TuCSon* digitare (sempre dalla directory *dir*):

```
java -cp bin\tucson.jar alice.tucson.ide.Inspector
```

## Creazione e test del sistema semaforico

Per prima cosa occorre creare con le istruzioni sopra citate, il nodo *TuCSon* e l'ispettore per analizzare il centro di tuple e verificare il corretto funzionamento del sistema semaforico.

Si consiglia di creare 2 ispettori, uno che monitora il centro di tuple visualizzando tutte le tuple senza alcun filtro, l'altro per visualizzare solamente il numero di utenti per ogni fila, in modo da avere una visione diretta e di facile lettura sull'andamento del traffico sulle file del semaforo.

Per far questo, cliccare la cartella *view* nella finestra *tuples* dell'ispettore, e inserire la maschera

```
numeroMacchineInFila(fila(_,_),_)
```

abilitando la funzione del filtro.

Per far partire il sistema semaforico, con l'inizializzazione del Tuple Centre e la creazione di tutti gli agenti citati nella relazione di progetto, occorre aprire una finestra di prompt, e, dalla directory *dir*, digitare:

```
java -cp bin;bin\2p.jar;bin\tucson.jar SimulaIncroci
```

Per verificare il corretto funzionamento del sistema in tutti i casi possibili, sono state create sette sequenze di test, che simulano l'arrivo di un numero di utenti in modo tale da creare le diverse condizioni di funzionamento.

Di seguito sono riportati i casi che le varie sequenze riproducono:

- Test0 – verifica che in condizioni di traffico scarso gli utenti ricevono l'abilitazione ad attraversare nel minor tempo possibile, evitando sprechi di tempo e cicli di verde inutili.
- Test1 – riproduce le condizioni di traffico casuale senza il verificarsi dell'onda verde.
- Test2 – riproduce l'arrivo di utenti al semaforo 1 tale da provocare l'onda verde.
- Test3 – simula un arrivo di utenti al semaforo 2 tale da provocare l'onda verde.

- Test4 – riproduce l'arrivo di un mezzo di emergenza al semaforo 1 in condizioni di normale funzionamento, cioè senza onda verde.
- Test5 – riproduce l'arrivo di un mezzo di emergenza al semaforo 1 in condizioni di onda verde.
- Test6 – riproduce l'arrivo di un mezzo di emergenza al semaforo 1 e istantaneo arrivo di utenti al semaforo 2 tale da far scattare l'onda verde.

Per lanciare una delle qualsiasi sequenze di test, occorre aprire una finestra di prompt, e, dalla directory *dir*, digitare:

```
java -cp bin;bin\2p.jar;bin\tucson.jar TestX
```

con X=numero del test da lanciare.

Ogni volta effettuato un test, per farne uno nuovo occorre chiudere l'applicazione della sequenza precedente, in quanto la successiva simulazione sarebbe eseguita a scatti, per un carico eccessivo della CPU (almeno per calcolatori non troppo grandi).

Quindi occorre chiudere l'applicazione *SimulaIncroci*, cancellare tutte le tuple presenti nel tuple centre, e cliccare il tasto *apply* della finestra *tuples-action* dell'ispettore.

In seguito lanciare l'applicazione *SimulaIncroci* e l'applicazione di *Test* desiderata.

Tutte le frasi utilizzate per lanciare le varie applicazioni sono riportate nel file *stringheDaLanciare.txt*.