

# AsmetaSMV

Un model checker per modelli AsmetaL

# Proprietà CTL/LTL in AsmetaL

- Le proprietà CTL devono essere dichiarate dopo la sezione degli invarianti. La sintassi di una proprietà CTL è:

**CTLSPEC  $p$**

dove  $p$  è un'espressione booleana.

- Le proprietà LTL devono essere dichiarate dopo le proprietà CTL. La sintassi di una proprietà LTL è:

**LTLSPEC  $q$**

dove  $q$  è un'espressione booleana.

# CTL/LTL Library

- Per usare le funzioni CTL in un modello AsmetaL, dobbiamo importare la libreria

**CTLlibrary.asm**

- Per usare le funzioni LTL in un modello AsmetaL, dobbiamo importare la libreria

**LTLlibrary.asm**

# Proprietà LTL

Connettore temporale	Significato
$X\varphi \rightarrow$ (Next)	$\varphi$ è vera in $s_t$ se e solo se $\varphi$ è vera nello stato $s_{t+1}$
$F\varphi \rightarrow$ (Future)	$\varphi$ è vera in $s_t$ se esiste $t' > t$ tale che $\varphi$ è vera in $s_{t'}$
$G\varphi \rightarrow$ (Globally)	$\varphi$ è vera in $s_t$ se per ogni $t' > t$ $\varphi$ è vera in $s_{t'}$
$\varphi_1 U \varphi_2 \rightarrow$ (Until)	È vera se esiste $t_n > t$ tale che $\varphi_2$ è vera in $s_{t_n}$ e per ogni $t \leq t_i \leq t_{n-1}$ $\varphi_1$ è vera in $s_{t_i}$

# Proprietà CTL

Connettore temporale		
A ("along All paths" (inevitably))	X	E ("along at least (there Exists) one path" (possibly))
	F	
	G	
	U	

# NuSMV

Symbolic model checker

# NuSMV

- <http://nusmv.fbk.eu/>
- Estrai i file
- Aggiungi NuSMV\bin nel path
- Riavvia il PC

# Esercizi

CoffeeVendingMachine

Tic-Tac-Toe

Sluice gate

CoffeeVendingMachine

# CoffeeVendingMachine

- Una macchinetta automatica dispensa caffè, tè e latte. La macchinetta accetta solo monete da 50 centesimi e da 1 euro. Se viene inserita una moneta da 50 centesimi, la macchinetta dispensa latte (se disponibile); se viene inserita una moneta da 1 euro, invece, la macchinetta decide in modo casuale di dispensare caffè o tè (se disponibili). Se viene dispensata una bevanda, la sua disponibilità viene decrementata e la moneta viene conservata nella macchinetta. Nel modello ASM ogni passo di macchina corrisponde all'inserimento di una moneta e all'erogazione di una bevanda corrispondente. La macchina all'inizio contiene 10 unità per ogni bevanda; l'atto di erogazione di una bevanda corrisponde alla diminuzione di un'unità della disponibilità della stessa e alla conservazione della moneta (nelle monete conservate, non bisogna distinguere tra monete da 50 centesimi ed 1 euro). Se la bevanda non è disponibile, non viene erogata e la moneta non viene conservata. La macchina può contenere al massimo 25 monete; quando la macchina è piena di monete, non accetta altre monete e, quindi, non eroga più alcuna bevanda. All'inizio la macchinetta non contiene alcuna moneta. L'utente del sistema decide ad ogni passo di simulazione il tipo di moneta da inserire.

# CoffeeVendingMachine

Scrivere le seguenti proprietà:

- una volta che un prodotto e' terminato non puo' essere disponibile nuovamente in futuro
- nella macchinetta ci sono sempre almeno  $k$  unita' di prodotto. Quanto vale questo  $k$ ?

$$\text{totaleB}=10+10+10$$

$$\text{MaxCoins}=25$$

$$K=30-25=5$$

- esiste uno stato in cui il latte e il te' sono terminati e ci sono ancora 9 caffe'

Tic-Tac-Toe

# Tic-Tac-Toe

- Il gioco del tris funziona nel modo seguente:
- l'utente (croce) deve giocare contro il computer (cerchio);
- ad ogni passo (in modo alternato) o l'utente o il computer devono eseguire una mossa;
- se l'utente esegue una mossa errata (inserisce la croce in una cella occupata), la mossa non deve essere eseguita e la macchina deve chiedere nuovamente la mossa all'utente;
- le mosse del computer sono scelte in modo casuale tra le mosse corrette;
- la mossa iniziale è dell'utente;

# Tic-Tac-Toe

Scrivere le seguenti proprietà:

- l'utente può vincere;
- il computer può vincere;
- l'utente e il computer non possono vincere insieme;
- il gioco può terminare senza che ci sia un vincitore.

Sluice gate

# Sluice gate

- A small sluice, with a rising and falling gate, is used in a simple irrigation system. A computer system is needed to control the sluice gate: the requirement is that the gate should be held in the fully open position for ten minutes in every three hours and otherwise kept in the fully closed position.
- The gate is opened and closed by rotating vertical screws. The screws are driven by a small motor, which can be controlled by clockwise, anticlockwise, on and off pulses.

# Sluice gate

Scrivere le seguenti proprietà:

- Se lo stato è FULLYCLOSED può rimanere in FULLYCLOSED o diventa OPENING
- Se lo stato è OPENING può rimanere OPENING o diventa FULLYOPENED
- Se lo stato è FULLYOPENED può rimanere FULLYOPENED o diventa CLOSING
- Se lo stato è CLOSING può rimanere CLOSING o diventa FULLYCLOSED
- Se lo stato è FULLYCLOSED il motore è spento
- Se lo stato è FULLYOPENED il motore è spento
- Se lo stato è OPENING il motore è acceso
- Se lo stato è CLOSING il motore è acceso
- Se lo stato è FULLYOPENED in futuro sarà FULLYCLOSED
- Se lo stato è FULLYCLOSED in futuro sarà FULLYOPENED

Ferryman problem

# Ferryman problem

- Un ferryman deve portare sull'altra sponda di un fiume un wolf, una goat ed un cabbage. Puo' trasportarne solo uno alla volta.
- Ci sono due situazioni di pericolo:
- il wolf mangia la goat se il ferryman non e' presente a controllare;
- la goat mangia il cabbage se il ferryman non e' presente a controllare.
- All'inizio sono tutti sulla sponda LEFT.
  
- In simulazione, ad ogni passo, bisogna decidere chi deve essere trasportato sull'altra sponda dal FERRYMAN: la GOAT, il CABBAGE, il WOLF oppure fare attraversare il fiume con nessuno a bordo (NONE).

# Ferryman problem

Scrivere le seguenti proprietà:

- Se su una sponda ci sono goat e cabbage deve esserci anche il ferryman
- Se su una sponda ci sono wolf e goat ci deve essere anche ferryman
- Esiste uno stato in cui tutti si trovano sulla sponda di destra
- E' possibile che goat sia da solo sulla sponda di sinistra
- E' possibile che il cabbage sia da solo sulla sponda di destra