AsmetaV

AValLa (Asmeta Validation Language)

Scenario ASM: primitive Avalla

- **Set**: commando per impostare il valore delle monitorate a un valore specifico. Simula l'ambiente
- **Check**: per verificare il valore delle funzioni dello stato corrente
- **Step**: esegue un passo dell'ASM
- **StepUntil**: esegue l'ASM finchè una condizione risulta vera
- **Invariant**: per verificare che una proprietà è sempre vera durante l'esecuzione dello scenario
- **Exec**: esegue una regola di transizione

NB: nel path NON devono esserci spazi!!!

Sintassi di Avalla

Abstract syntax	Concrete syntax
Scenario	**scenario** name **load** spec_name Invariant* Command*
spec_name is the spec to load; invariants and commands are the script content	
Invariant	**invariant** name ':' expr ';'
expr is a boolean term made of function and domain symbols of the underlying ASM	
Command	(Set | Exec | Step | StepUntil | Check)
Set	**set** loc := value ';'
loc is a location term for a monitored function, and value is a term denoting a possible value for the underlying location	
Exec	**exec** rule ';'
rule is an ASM rule (e.g. a choose/forall rule, a conditional if, a macro call rule, ect.)	
Step	**step** ';'
StepUntil	**step until** cond ';'
cond is a boolean-valued term made of function and domain symbols of the ASM	
Check	**check** expr ';'
expr is a boolean-valued term made of function and domain symbols of the ASM	

Esercizi

Advanced Clock

ATM

Advanced clock

Advanced clock

- Modellare il funzionamento di un Clock che ad ogni passo incrementa di 1 i secondi.

Advanced clock

- Scenario 1:
  - controllare che all'inizio sia mezzanotte (00:00:00);
  - fare un passo di macchina;
  - controllare che l'ora sia 00:00:01;
  - fare un passo di macchina;
  - controllare che l'ora sia 00:00:02.

Advanced clock

- Scenario 2:
  - eseguire la macchina fino a quando la funzione hours non vale 3;
  - controllare che l'ora sia 03:00:00;
  - fare un passo di macchina;
  - controllare che l'ora sia 03:00:01.

Advanced clock

- Scenario 3:
  - impostare un invariante di scenario che afferma che l'ora è sempre minore di 3;
  - impostare una par rule che, con tre update rule, modica l'ora in 02:01:59;
  - fare un passo di macchina;
  - controllare che l'ora sia 02:02:00;
  - impostare una par rule che, con tre update rule, modica l'ora in 00:00:00.
- Scenario 3 modificato:
  - Modificare lo scenario 3 per fare in modo che l'invariante di scenario sia violato.

ATM

ATM

- Design the control for an ATM, and show it to be well functioning, where via a GUI the customer can perform the following operations:
- Op1. Enter the ID. Only one attempt is allowed per session; upon failure the card is withdrawn.
- Op2. Ask for the balance of the account. This operation is allowed only once and only before attempting to withdraw money.
- Op3. Withdraw money from the account. Only one attempt is allowed per session. A warning is issued if the amount required exceeds the balance of the account."

ATM

Dato il modello ASM scrivere i seguenti scenari:

- Scenario 1:
  - un utente inserisce la carta card1 ed il pin corretto;
  - preleva 50 euro;
  - viene disconesso;
  - dopo ogni passo controllare che lo stato atmState sia corretto.

ATM

- Scenario 2:
  - Scrivere uno scenario in cui un utente inserisce il pin errato e gli viene negato l'accesso.
  - Controllare sempre che lo stato atmState sia aggiornato in modo corretto.

ATM

- Scenario 3:
  - impostare il saldo di card1 a 10 euro;
  - autenticare card1;
  - provare a prelevare 50 euro;
  - successivamente provare a prelevare 10 euro.
- Dopo ogni passo controllare che lo stato atmState sia corretto; controllare anche che il saldo di card1 ed i soldi disponibili nel bancomat (moneyLeft) siano aggiornati in modo corretto.

ATM

- Scenario 4:
  - Impostare un invariante di scenario che afferma che la somma disponibile nel bancomat non e mai inferiore a 900 euro;
  - eseguire una serie di prelievi, facendo in modo che l'invariante non sia mai violato.

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

- Scenario 1:
  - Scrivere uno scenario in cui si mostri il ciclo di vita completo della chiusa: FULLYCLOSED - OPENING - FULLYOPENED - CLOSING - FULLYCLOSED.

Sluice gate

- Scenario 2:
  - Modificare lo scenario 1 per fare in modo che il passaggio da uno stato a quello successivo della chiusa avvenga in due passi di macchina. Ogni passaggio di stato della chiusa e condizionato al valore di una determinata locazione monitorata (es. si passa da FULLYCLOSED a OPENING se passed(closedPeriod) e true): se la monitorata vale false la chiusa non cambia stato.

Algoritmo di Euclide

Algoritmo di Euclide

- Creare un'ASM in AsmetaL che implementi l'algoritmo di Euclide nel seguente modo:
- ogni step della macchina deve corrispondere ad un'iterazione del ciclo while;
- i valori di cui bisogna calcolare l'MCD devono essere impostati nello stato iniziale;
- Modificare il modello visto in precedenza per fare in modo che, in simulazione, venga richiesto all'utente il valore dei due numeri su cui eseguire l'MCD.

Algoritmo di Euclide

- Scenario 1:
  - Eseguire la macchina no a quando numA = numB; controllare che il valore di numA sia 13.

Algoritmo di Euclide

- Scenario 1:
  - Scrivere un invariante che afferma che numA e sempre maggiore di numB;
  - modificare i valori di numA e numB, rispettivamente, a 24 e 6;
  - eseguire la macchina fino a quando numA = numB;
  - controllare che il valore di numA sia 6.

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

- Scenario 1:
  - Scrivere uno scenario in cui si mostrino tutti i passi necessari (settaggio della monitorata carry con il comando set ed esecuzione del passo con il comando step) a trasportare tutti gli attori sulla sponda destra.
  - Dopo ogni passo controllare che le posizioni degli attori siano quelle attese (comando check).

Ferryman problem

- Scenario 2:
  - Posizionare il ferryman, la goat ed il wolf sulla sponda destra;
  - eseguire tutti i passi che servono per trasportare tutti gli attori sulla sponda destra.