TESTING E VERIFICA DEL SOFTWARE
ESERCITAZIONE ASMETA
SIMULATORE E VALIDATORE

INDICAZIONI PER UTILIZZARE Asmeta

Scarica il file

https://foselab.unibg.it/asmeta/eclipse_asmeta_smv_2023_05.zip

da

https://github.com/asmeta/asmeta

oppure scarica lo zip con tutti i tool dall'area restricted

https://cs.unibg.it/gargantini/didattica/swtestandver/restricted/tools_x_tvsw_2023.zip

Entrambi contengono sia eclipse con asmeta che NuSMV

Esegui il file eclipse_nusmv.bat che setta il path con anche NuSMV

Utenti macOS e linux:

Che usa macOS dovrà scaricare eclipse per macOS, installare il plugin di asmeta dall'update site

https://raw.githubusercontent.com/asmeta/asmeta_update_site

Scaricare il model checker NuSMV:

https://nusmv.fbk.eu/NuSMV/download/getting_bin-v2.html

e aggiungerlo al PATH

Compito:

Prova a scrivere il modello asmeta per ognuno dei seguenti esercizi, prova a simularlo e animarlo per assicurarti che sia corretto.

Esercizio 1 Algoritmo di Euclide

```
Function MCD (a,b)
        while a!=b
                if a>b
                        a:=a-b
                else
                        b:=b-a
        return a
```

Creare un'ASM in AsmetaL che implementi l'algoritmo di Euclide nel seguente modo:

- I valori dei due numeri numA e numB rappresentano i valori dei due numeri su cui calcolare l'MCD
    - Sono valori controllati dalla macchina
    - Devono essere inizializzati nello stato iniziale a due valori a tua scelta, tra cui vuoi calcolare l'MCD (quindi non domandiamo nulla all'utente)
- Ogni step della macchina deve corrispondere ad un'iterazione del ciclo while;
- Il modello deve essere eseguito con l'opzione "run until empty".
- (Variante) 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

Suggerimenti:

Usa due controlled function di tipo intero per rappresentare a e b

Coffee Machine

- Una macchinetta automatica dispensa caffe, te 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.
- Ogni passo di ASM deve corrispondere all'inserimento di una moneta e all'eventuale erogazione di una bevanda corrispondente.
- L'utente del sistema decide, ad ogni passo di simulazione, il tipo di moneta da inserire.

- La macchina all'inizio contiene 10 unita per ogni bevanda; l'atto di erogazione di una bevanda corrisponde alla diminuzione di un'unita 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 e disponibile, non viene erogata e la moneta non viene conservata.
- La macchina può contenere al massimo 25 monete. Quando la macchina e piena di monete, non accetta altre monete e, quindi, non eroga più alcuna bevanda.
- All'inizio la macchinetta non contiene alcuna moneta.

Suggerimenti:

La segnatura della macchina proposta è:

signature:
    enum domain CoinType = {HALF | ONE}
    enum domain Product = *…caffè e altri tipi….*
    domain QuantityDomain subsetof Integer
    domain CoinDomain subsetof Integer
    controlled available: *… funzione che per ogni prodotto dice la quantità*
    controlled coins: CoinDomain
    monitored insertedCoin: CoinType
definitions:
    domain QuantityDomain = …
    rule r_serveProduct($p in Product) = …
    main rule r_Main = …

default init s0:
….

Morra Cinese

- Lo scopo del gioco e quello di sconfiggere l'avversario scegliendo un segno (arma) in grado di battere quello dell'altro, secondo le seguenti regole:
    - Il sasso spezza le forbici (vince il sasso)
    - Le forbici tagliano la carta (vincono le forbici)
    - La carta avvolge il sasso (vince la carta)
    - Se i due giocatori scelgono la stessa arma, il gioco è pari e si gioca di nuovo.
- Creare un modello ASM che permetta di giocare a Morra cinese contro il computer. Il modello deve avere le seguenti caratteristiche:
    - ogni step della macchina corrisponde ad una singola giocata;
    - l'utente deve avere la possibilità di scegliere uno dei tre simboli;
    - il computer deve scegliere in modo casuale uno dei tre simboli;

    - la macchina deve valutare la giocata e segnalare il vincitore.

- Tre possibili raffinamenti successivi del modello Asm possono essere:

1. Memorizzare il numero di partite vinte dall'utente e dal computer;
2. Impostare il numero massimo di partite che devono essere giocate;
3. Se, dopo aver giocato il numero massimo di partite, il computer e l'utente sono in parità, permettere che continuino a giocare fino a quando uno dei due non vince.

Suggerimenti:

- Un dominio per il simbolo della giocata
- Un dominio per il risultato
- Una funzione monitorata che rappresenta la scelta dell'utente
- Una funzione controllata che rappresenta la scelta non deterministica del computer (casuale)
- Puoi fare anche una string controllata che rappresenta il messaggio se uno ha vinto oppure no
- Puoi fare anche una funzione che confronta due segni dà il risultato
  static playResult: Prod(Sign, Sign) -> Result //data una coppia di giocate, ritorna il risultato