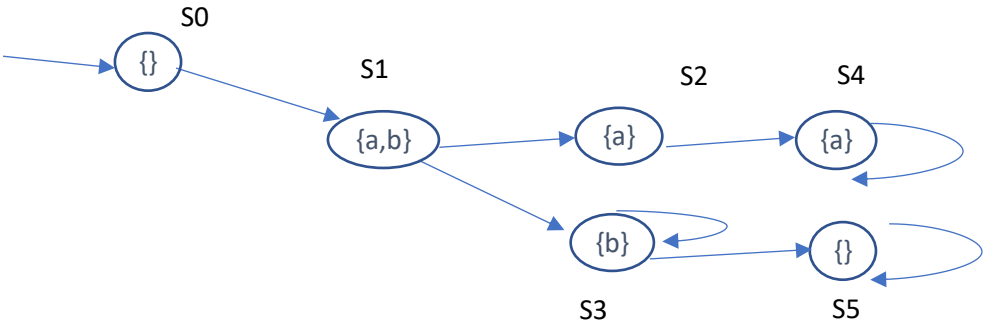
TEMA D'ESAME 20 GIUGNO 2023

COGNOME _____

Durata 3.30h

1. algoritmo di model checking

Data la seguente macchina M con AP = {a,b}



Mediante l'algoritmo di model checking, dire in quali stati s valgono le proprietà:

1. M,s |= AG(a)
2. M,s |= AF(a)
3. M,s |= EF(not b)
4. M,s |= (a and b) implies EG(a)
5. M,s |= b implies EX(a)
6. M,s |= E(b U not b)
7. M,s |= A((a or b) U (not b or a))

Nota che proprietà CTL potrebbero aver bisogno di essere trasformate per applicare l'algoritmo di model checking.

Commenta il perché alcune proprietà valgono in alcuni stati e in altri no come ti aspetteresti.

2. combinatorial testing

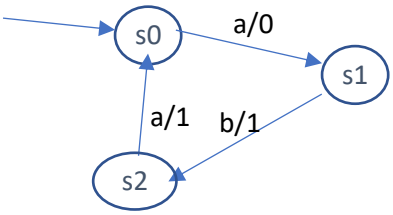
Date 3 variabili con i loro domini x: 1,2 y: a,b z: F,G

Costruisci la test suite combinatoriale pairwise usando l'algoritmo IPO (cerca di minimizzare la dimensione della test suite).

Se ci fosse il vincolo **x=1 implies y!=a** dovresti modificare la test suite? Se sì come, se no perché?

3. conformance testing

Data la seguente FSM con 4 stati, due input a e b e due output 1 e 0



La macchina è correttamente definita? (giustifica la risposta – se non lo è, correggi la macchina aggiungendo almeno solo le transizioni necessarie per correggere la macchina – non usare solo loop)

Scrivi due test sequences, una per la copertura degli stati e una per la copertura delle transizioni.

Fa due esempi di errori (di quelli visti) e controlla se riesci a scoprirli con i test che hai trovato tu (può succedere che tu non riesca a scoprirli?) Giustifica la risposta.

Indica bene le test sequences come test di input e anche gli output attesi.

ESERCIZI CON ECLIPSE

Per questi esercizi prepara un file word in cui descrivi la tua soluzione.

4. Testing di un programma

Data la seguente classe in Java che conta da 0 a 5 con un dato step (sotto certe condizioni):

```
public class Car {

  private int speed = 0;

  // ferma la macchina (decrementa speed di x tra 1 e 10) restituisce true
  // solo se la macchina è ferma
  boolean int stop(int decel) {
    if (decel >= 1 && decel <= 10 && speed > 0 && speed - decel <= 0)){
                speed -= decel;
                if (speed < 0) speed = 0;
                return true;
          }
     if (decel >= 1 && decel <= 10 && speed >= decel)
                 speed -= decel;
    return false;
  }
   void acc(int step) {
    if (step >= 1 && step <= 4) speed += step;
   }
}
```

Scrivere i casi di test JUNIT per ottenere la copertura

- Delle istruzioni
- Dei branch (solo quelli in più rispetto alle istruzion)
- MCDC

ESERCIZIO per asmeta e JML TALKSHOW

5. Asmeta: progetto talk show moderato (talkshow.asm)

Nella stanza ci sono 6 persone ed una moderatrice, inizialmente le persone sono nello stato di ascolto e la moderatrice ha la parola. Ogni persona può essere in ascolto, in attesa di parlare o parlare.

Ad ogni passo di simulazione:

- vengono considerate tutte le persone:
  - se la persona sta parlando e non ha esaurito il tempo a disposizione per parlare continua a parlare e viene diminuito il tempo a disposizione. Quando il tempo termina torna nello stato di ascolto. Il tempo a disposizione è di 5 minuti.
  - se sono in attesa di parlare viene diminuito il tempo per cui sono disposte ad attendere prima di parlare, nel caso il tempo si esaurisca decidono di non restare più in attesa e tornano in semplice ascolto. Il tempo massimo di attesa di parlare è di 7 minuti.
  - se sono in ascolto, continuano ad ascoltare
  - se non c'è nessuna persona che vuole parlare e nessuna parla, la moderatrice continua a parlare lei.
- una nuova persona decide di mettersi in attesa per parlare con una probabilità del 30% (simula la probabilità con un intero da 1 a 10 ad esempio) soltanto se si trova nello stato di ascolto. La persona può decidere in qualsiasi istante (anche mentre la moderatrice sta parlando) di voler parlare. Per gli scenari usa step until oppure introduci delle variabili monitorate (puoi fare diverse versioni)
- la moderatrice gestisce le persone: se una persona ha finito il tempo, la moderatrice sceglie la prossima persona che parli tra quelle che sono in attesa di parlare

- ogni persona ha un microfono e anche la moderatrice, però il microfono di una sola persona (quella che parla) deve essere attivato. Quando la moderatrice dà la parola ad una persona, attiva il suo microfono.
- inizialmente la moderatrice ha il microfono acceso e tutti gli altri sono spenti
- ogni passo di macchina corrisponde ad 1 minuto nella realtà

Anima la specifica e fai un paio di screenshot e copia nel tuo file word.

Scrivi uno scenario in cui una persona parla e poi smette di parlare e parla qualcun altro.

Scrivi le seguenti proprietà e dimostrale (o prova che sono false)

Proprietà P1:

ogni persona prima o poi può parlare

Proprietà P2:

solo una persona alla volta parla

Proprietà P3:

solo la persona che parla ha il microfono attivato

Scrivi inoltre almeno una proprietà che è giustamente falsa e controlla che il model checker trovi il contro esempio atteso (puoi partire anche dalle proprietà sopra).

Riporta ogni proprietà nel file word e spiegale.

6. JML progetto TalkShow.java

Scrivi il codice in Java del sistema e modella pre/post e invarianti con JML.

Testa i contratti JML con una classe main in cui chiami i diversi metodi e controlli runtime la verifica dei contratti - prova anche a modifcare il codice per vedere che i contratti (di diverso tipi) violati siano identificati (copia degli screenshot). Documenta bene le violazioni e le loro cause in commenti.

Prova a dimostrare la correttezza con ESC.

7. MODEL BASED TESTING

Traduci lo scenario avalla nel caso di test JUNIT per il codice java che hai scritto nel progetto
COGNOME_TALKSHOW_JML

Prova ad introdurre un errore nel codice del TalkShow e controlla che il caso di test lo catturi. Documenta il tutto con commenti nel codice e nel codice.