

# Semaforo Senso Unico

C'è un impianto semaforico che regola un senso unico. Ai due estremi di una strada ci sono due semafori (0 e 1) che possono essere comandati da una centralina di cui dobbiamo scrivere il software.

Ogni semaforo può essere messo a verde (0), giallo (1) o rosso (2).

Un semaforo può essere messo a verde solo se esso è rosso e anche l'altro è rosso. Può essere messo a giallo solo se esso stesso è verde e può essere messo a rosso solo se esso è giallo.

Inizialmente i due semafori sono entrambi rossi.

## Compito

In un file word annota tutto quello che fai. Aggiungi anche gli screenshot per i punti principali (ad esempio copertura ...). Per effettuare gli screenshots usa lo strumento di cattura di windows. Anche la documentazione sarà valutata per ogni singolo progetto (circa del 15%).

Sviluppa un progetto in eclipse per ognuna delle domande. Ogni progetto dovrà avere come nome il tuo COGNOME\_ES (con ES il tipo dell'esercizio – vedi titoli degli esercizi). Puoi anche andare in parallelo (ad esempio modello NUSMV e codice Java). Metti i progetti e il documento word in una directory con nome COGNOME\_NOME (che puoi usare come workspace). Alla fine crea un file zip e consegna quello.

Commenta per bene tutto il codice che scrivi.

## Java

Rappresenta lo stato dei due semafori con dei numeri interi.

Definisci la classe SemaforoSensoUnico con questi metodi

`boolean changecolor(int sem, int color)`

che cambia stato del semaforo “sem” settando il colore a color (vedi sopra). Il metodo restituisce TRUE solo se il semaforo ha cambiato stato.

`public String toString()`

restituisce una string con 2 caratteri che rappresentano lo stato dei due semafori: R per rosso, G per giallo, V per verde. Ad esempio “RR” vuol dire che entrambi sono rossi.

## 1. ES = TESTING

Scrivere l'implementazione e i casi di test con Junit. Scrivi una classe di test **DueCicli** con un caso di test che mostri uno scenario ognuno dei due semafori diventa verde e poi rosso. Nei casi di test controlla che effettivamente lo stato dell'altro semaforo non cambi (puoi usare toString se vuoi).

Esegui il controllo della copertura (istruzioni, branch, decisioni, MCDC). Se il caso sopra non è sufficiente ad avere una copertura soddisfacente, aggiungi i casi di test necessari (in una classe separata per ogni copertura).

Se ritieni che qualche decisione non sia copribile giustificalo.

Metti la directory test con i test separata (oltre src).

Prova con Randoop a generare casi di test (con pochi casi di test, tipo 10) e prova a valutarne la copertura.

## 2. ES = JML

Scrivi i contratti JML. Cerca di scrivere sia le precondizioni, che le postcondizioni dei metodi.

Cerca di spostare nelle precondizioni alcuni controlli che facevi all'inizio dei metodi nel precedente esercizio.

Cerca di scrivere anche invarianti.

Prova i contratti JML con una classe main in cui chiami i diversi metodi.

Prova anche a modificare il codice e controlla che i contratti siano violati. Documenta bene le violazioni e le loro cause in commenti e nel file di documento.

## 3. ES = KEY Verifica dei programmi

CONSIGLIO. Inizia da una classe con contratti **vuoti** (true) e **aggiungi** mano a mano i contratti però sempre dimostrandone la loro correttezza – cioè le prove devono essere sempre chiuse **vere**. La valutazione dipenderà dal massimo contratto che riuscite a dimostrare vero (contratti non provati non verranno valutati).

Se la tua classe contiene dei metodi delle librerie (tipo `system.out.println`) eliminali. Anche `Random` va eliminata se l'hai usata.

## 4. ES = NUMSV

Specificare `SemaforoSensoUnico` in NuSMV e le sue proprietà usando LTL e/o CTL. Puoi assumere che ad ogni passo almeno un semaforo cambi stato sempre.

Tra le proprietà desiderate (alcune potrebbero essere false) c'è:

- P1: i due semafori non possono mai diventare entrambi verdi
- P2: se un semaforo è verde allora l'altro è sempre rosso

Provare se sono vere oppure no. Se sono false presentare un contro esempio (se viene prodotto).

Aggiungi anche altre proprietà che ti sembrano importanti, cerca di provarle e commentale/spiegale per bene (valutazione).

Scrivi almeno una proprietà con operatore EX o AX

Per ogni proprietà aggiungi una descrizione in Italiano di quello che vuol dire.

## 5. ES = FSM (MBT con le FSM)

Scrivi una FSM usando ModelJunit (non connetterla con la sut, usa ModelJunit solo per costruire la EFSM).

Prova sia a fare online testing che offline testing.

Salva il disegno dell'impianto semaforico come immagine.

## 6. ES = COMB (Combinatorial testing)

Rappresenta in combinatorial testing lo stato dei due semafori. Introduci anche gli opportuni vincoli.

Applica il combinatorial testing.

Prova a generare la copertura pairwise.