

Esercizi per il corso di Tecnica di Specifica e di Verifica (consegnato il 02/05/2003)

Modellare le 5 seguenti proposte di soluzione al problema della mutua esclusione con NuSMV. Per ciascun modello NuSMV verificare le seguenti proprietà, dopo averle formalizzate in LTL:

1. **Mutua esclusione.**
2. **Assenza di starvation** [ciascun processo, qualora inizi la procedura di ingresso in sezione critica, prima o poi eseguirà la sua sezione critica]
3. È possibile, ma non necessario, che un processo entri anche due volte di seguito nella sezione critica [**assenza di ordinamento stretto tra i processi**].

Nota:

- è ammesso l'uso di **FAIRNESS** constraints ove ritenuto opportuno.
- La "possibilità" (come in proprietà 3) di una proprietà corrisponde a dire che esiste un percorso che soddisfa tale proprietà. Questo, come abbiamo visto, non è direttamente esprimibile in LTL (la soddisfacibilità di una formula LTL è infatti definita come la soddisfacibilità rispetto a tutte le computazioni del modello). Un modo per ovviare a tale inconveniente è quello di utilizzare una seconda proprietà in LTL (quindi che si richiede valga su tutte le esecuzioni) che però **non** soddisfa la proprietà desiderata. Se, infatti, fosse possibile che la prima proprietà valesse, allora NuSMV risponderebbe che la seconda proprietà è falsa, mostrando inoltre un contro-esempio alla seconda proprietà. Tale contro-esempio dimostra proprio l'esistenza di una computazione che soddisfa la proprietà desiderata.

Soluzione 1

PROCESS A	PROCESS B
<pre>repeat non-critical code /* entry_protocol; */ flag1 := true; while flag2 do skip; /* end entry_protocol; */ critical section; /* exit_protocol; */ flag1 := false; /* end exit_protocol; */ non-critical code forever;</pre>	<pre>repeat non-critical code /* entry_protocol; */ flag2 := true; while flag1 do skip; /* end entry_protocol; */ critical section; /* exit_protocol; */ flag2 := false; /* end exit_protocol; */ non-critical code forever;</pre>

Soluzione 2

PROCESS A	PROCESS B
<pre>repeat non-critical code /* entry_protocol; */ while flag2 do skip; flag1 := true; /* end entry_protocol; */ critical section; /* exit_protocol; */ flag1 := false; /* end exit_protocol; */ non-critical code forever;</pre>	<pre>repeat non-critical code /* entry_protocol; */ while flag1 do skip; flag2 := true; /* end entry_protocol; */ critical section; /* exit_protocol; */ flag2 := false; /* end exit_protocol; */ non-critical code forever;</pre>

Soluzione 3

PROCESS A	PROCESS B
<pre>repeat non-critical code /* entry_protocol; */ while (turn = 2) do skip; /* end entry_protocol; */ critical section; /* exit_protocol; */ turn := 2; /* end exit_protocol; */ non-critical code forever;</pre>	<pre>repeat non-critical code /* entry_protocol; */ while (turn = 1) do skip; /* end entry_protocol; */ critical section; /* exit_protocol; */ turn := 1; /* end exit_protocol; */ non-critical code forever;</pre>

Soluzione 4

<p>PROCESS A</p> <pre>repeat non-critical code /* entry_protocol; */ flag1 := true; while flag2 do if (turn = 2) then flag1 := false; while (turn = 2) do skip; flag1 := true; /* end entry_protocol; */ critical section; /* exit_protocol; */ turn := 2; flag1 := false; /* end exit_protocol; */ non-critical code forever;</pre>	<p>PROCESS B</p> <pre>repeat non-critical code /* entry_protocol; */ flag2 := true; while flag1 do if (turn = 1) then flag2 := false; while (turn = 1) do skip; flag2 := true /* end entry_protocol; */ critical section; /* exit_protocol; */ turn := 1; flag2 := false; /* end exit_protocol; */ non-critical code forever;</pre>
--	---

Soluzione 5

<p>PROCESS A</p> <pre>repeat non-critical code /* entry_protocol; */ flag1 := true; turn := 2; while flag2 and (turn = 2) do skip; /* end entry_protocol; */ critical section; /* exit_protocol; */ flag1 := false; /* end exit_protocol; */ non-critical code forever;</pre>	<p>PROCESS B</p> <pre>repeat non-critical code /* entry_protocol; */ flag2 := true; turn := 1; while flag1 and (turn = 1) do skip; /* end entry_protocol; */ critical section; /* exit_protocol; */ flag2 := false; /* end exit_protocol; */ non-critical code forever;</pre>
---	---