Technical University of Denmark

02246 Model Checking

Mandatory Assignment Part 1: Discrete Modelling and Verification

Authors:

Andreas Hallberg Kjeldsen
s092638@student.dtu.dk

Morten Chabert Eskesen
s133304@student.dtu.dk

October 21, 2013

Part A: Introductory Problems

Practical Problems

1.

(a)

$client_1$: $AG(state_1 = 1 \Rightarrow \neg(\neg job_1 = 1 \wedge \neg job_2 = 1))$
$client_2$: $AG(state_2 = 1 \Rightarrow \neg(\neg job_1 = 2 \wedge \neg job_2 = 2))$

(b)

$P >= 1[Gstate1 = 1 => (job1 = 1)|(job2 = 1)]$ - Verified.
$P >= 1[Gstate2 = 1 => (job1 = 2)|(job2 = 2)]$ - Verified.

(c)

$client_1$: $AG(state_1 = 1 \Rightarrow job_1 = true)$
$client_2$: $AG(state_2 = 1 \Rightarrow job_2 = true)$
We require that whenever $state_1 = 1$ then $job_1 = true$ because there should be a job waiting in the queue when the $state_1 = 1$. The same goes for $client_2$.

(d)

$P >= 1[Gstate1 = 1 => job1 = true]$ - Verified.
$P >= 1[Gstate2 = 1 => job2 = true]$ - Verified.

2.

(a)

We added an extra module called $client_3$ with the same commands as the two other clients with the names of the commands corresponding to $client_3$. We changed the finite range, which job_1 and job_2 can take their value over to $0 \ldots 3$. This does not increase the length of the queue because there is still only two jobs allowed in the queue (job_1 and job_2). We also added commands create3, serve3 and finish3 and only changed the values according to the number of $client_3$.

(b)

In the new model there are 214 reachable states.

(c)

A client cannot create a job when the queue is full. This is because all the modules synchronize over all action names that appear syntactically in the modules. The commands create1, create2 and create3 are also in scheduler with a guard that specifies that the $job_2 = 0$ for creation of a job to be possible, and $job_2 = 0$ is only true if the queue is empty.

(d)

Yes the properties previously verified still hold in the new model. They do because the clients' states will only be 1 if the job is in the scheduler since the modules are synchronized.

3.

(a)

We added another job to the queue called job_3 which will hold the third job of the queue. We also changed the create commands to have the guard $job_3 = 0$ because now this is the last job in the queue, so when it is 0 there is a place for one more job. Furthermore we added another method for shifting the queue when there is an empty slot. The old command stays in place, but there is now another command with the guard $job_2 = 0$ & $job_3 > 0$ that shifts job_3 to job_2 so it is moved up in the queue. Since the commands have no action names the commands can always occur *independently* of what any other modules in the systems are doing - just so long as its guard is true.

(b)

In the new model there are 1459 reachable states.

(c)

The properties does not hold in the new model, because the queue is now of length 3. Which means that a job created by a client could be in scheduled as the last job, i.e. in job_3. Example: this would cause (for $client_1$) to have $state_1 = 1$ while $job_3 = 1$ because the job is at the end of the queue.

(d)

??????????????????????????????

4.

(a)

$AG\ \phi$ specifies that from all the paths from this state ϕ should hold. Whereas property ϕ should only hold in that state.

(b)

The semantics are different in the version of PRISM we use. If the property ϕ should hold in all reachable states it should be written $AG\ \phi$. Because if only ϕ has been written as the property - this version of PRISM will only check if the property ϕ holds in the *initial* state.

(c)

(d)

Theoretical Problems

1.

(a)

(b)

2.

(a)

(b)

(c)

(d)

(e)

3.

(a)

(b)

(c)

(d)

4.

Part B: Intermediate Problems

Practical Problems

1.

(a)

(b)

(c)

(d)

(e)

2.

(a)

(b)

(c)

(d)

(e)

Theoretical Problems

1.

(a)

(b)

(c)

(d)

(e)

(f)

2.

(a)

(b)

(c)

(d)

Part C: Advanced Problems

Only one of these sections need to be answered.

Practical Problems

Theoretical Problems