Formal techniques for software and hardware verification

Lecturers:

Vladimir Zakharov

Vladislav Podymov

e-mail:

valdus@yandex.ru

2020, fall semester

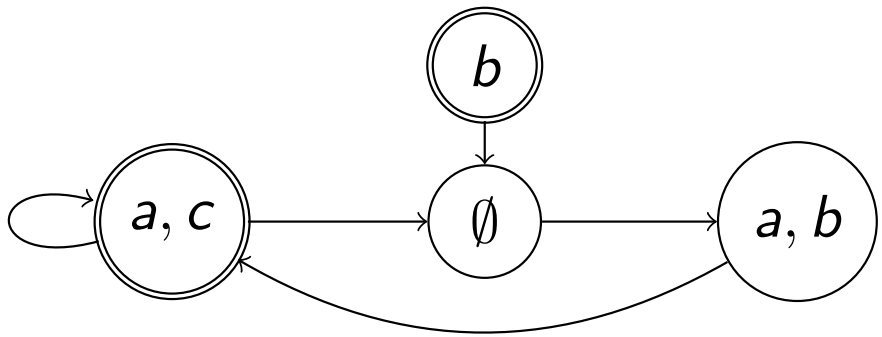
Seminar 8–9

Spin

exercises

Exercise 1: Kripke structures

Find out (*using Spin*) whether the following structure satisfies the following formulae



- $\mathbf{G}(a \rightarrow b \vee c)$
- $\mathbf{GF}a$
- $\mathbf{FG}a$
- $\mathbf{GF}\neg c \rightarrow \mathbf{F}(a \wedge b)$
- $\mathbf{GF}\neg c \rightarrow \mathbf{G}(\neg b \mathbf{U} a \wedge b)$

Exercise 2: programs

The following three programs execute in parallel, each one infinitely in a loop:

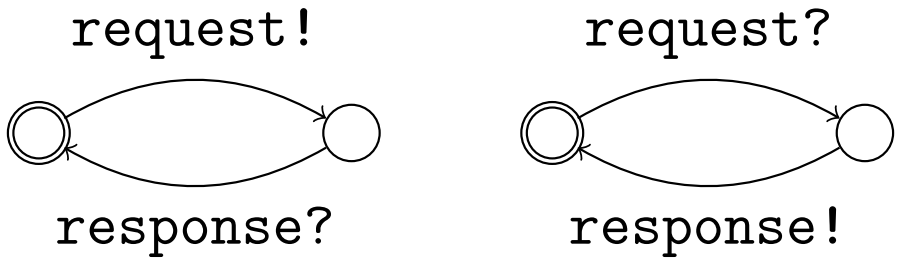
1. `if(x < 10) x = x + 1;`
2. `if(x > 0) x = x - 1;`
3. `if(x == 10) x = 0;`

An initial value of `x` is 0

Find out whether all possible values of `x` belong to the interval (a) `[0,10]`, and (b) `[-1,11]`, assuming that:

1. each mentioned conditional statement is a sequence of two atomic actions: a test, and an assignment
2. the whole conditional statement is atomic

Exercise 3: messages



The system consists of the client (the left automaton), the server (the right automaton), and a two-way communication channel

1. either synchronous,
2. or asynchronous with a queue of capacity 1 for each way

"`m!`"/"`m?`" means that the message `m` is being sent to/received from the channel during a transition

Check the following properties:

- After sending a `request`, the client will eventually receive a `response`
- The client and the server cannot wait for messages simultaneously

Exercise 4: synchronization tricks

```
bool b1, b2;
active proctype P() {
  do
  :: b1 = !b1; b2 = !b2;
  od
}
```

Extend the code above so that it is possible to check the following property:
"At the beginning of every loop iteration the values of `b1` and `b2` are equal"

Check the property

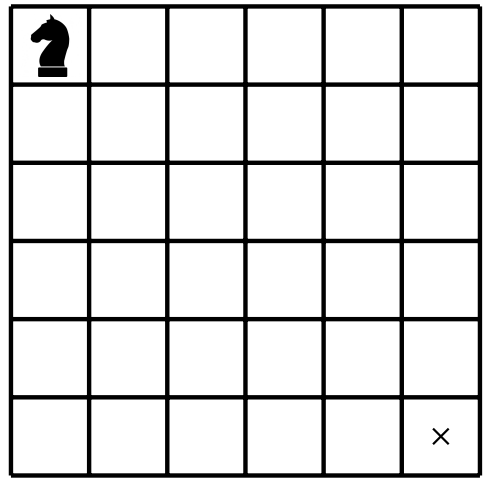
Exercise 4: synchronization tricks

```
bool b1, b2;
active proctype P1() {
  do :: b1 = !b1; od
}
active proctype P2() {
  do :: b2 = !b2; od
}
```

Extend the code above so that if one of the assignments (A) is executed more times than another (B), then B is to be executed before the next execution of A

Check the following property for the extended code:
"When the numbers of executions of A and B are equal, then the values of `b1` and `b2` are equal"

Exercise 5: trivial chess puzzle



Imagine a 6x6 chess board, and recall how a knight moves in chess

Can a knight reach the bottom right square
from the top left square?

Is it true that a knight placed in **any** square
can reach the bottom right square?