Prof. Dr. Andreas Podelski
Tanja Schindler

Hand in until November 10th, 2017
11:59 via the post boxes
Discussion: November 13th, 2017

Tutorial for Cyber-Physical Systems - Discrete Models
Exercise Sheet 3

Exercise 1: Program Graph – Atomic Expressions

In the following there are two examples of parallel systems shown, which seem to do semantically the same. The initial value of x is 1 for both programs and each line of the pseudocodes reflects an atomic expression which is executed at once.

(a) The first example is given in a high level language.

/* PROCESS 1 */	/* PROCESS 2 */
`1: x := x+1;`	`1: x := 3*x;`

Draw the program graphs PG_1 and PG_2 for each process with $Var_1 = Var_2 = \{x\}$. The domain of x is $\{-128, -127, \dots, -1, 0, 1, \dots, 127\}$ (signed 8-bit integer).

Declare the effect functions Effect_1 and Effect_2.

Draw the interleaving of the program graphs $PG_1 ||| PG_2$ and provide the effect function.

Which values can be finally stored in x?

(b) The second example is given in assembler.

/* PROCESS 1 */	/* PROCESS 2 */
`1: LOAD x;`	`1: LOAD x;`
`2: ADDI 1;`	`2: MULTI 3;`
`3: STORE x;`	`3: STORE x;`

Draw the program graphs PG_1 and PG_2 for each process with $Var_1 = \{x, ACC_1\}$ and $Var_1 = \{x, ACC_2\}$. The domain of x and ACC_i is $\{-128, -127, \dots, -1, 0, 1, \dots, 127\}$ (signed 8-bit integer).

Semantics of the assembler commands:

- `LOAD i` : Load the content of address i of the memory into an accumulator register ACC.
- `MULTI i` : Multiplies the content of register ACC with i and stores the result in ACC.
- `STORE i` : Store the content of register ACC at memory address i.

Note: A variable is here represented by a memory address which is a typical abstraction.

Declare the effect functions Effect_1 and Effect_2.

Draw the interleaving of the program graphs $PG_1|||PG_2$ and provide the effect function.

Which values can be finally stored in x?

Exercise 2: Peterson's mutual exclusion algorithm
In the textbook you have seen a version of Peterson's mutual exclusion algorithm in which each of the two processes did the assignments to b_i and x in an atomic step. In this exercise we consider two variants of Peterson's algorithm. VARIANT 1 is depicted below. We note that there is a fourth location which allows us to do the above mentioned assignments non-atomically. VARIANT 2 is obtained by swapping for each process the statements at locations *nc* and *req*.

```
          while true  {
              ......
nc :          $b_1$ := true;
req :         $x$ := 2;
wt :          wait until($x = 1 \vee \neg b_2$);
cs :          ...critical section...
              $b_1$ := false;
              ......
          }
```

```
          while true  {
              ......
nc :          $b_2$ := true;
req :         $x$ := 1;
wt :          wait until($x = 2 \vee \neg b_1$);
cs :          ...critical section...
              $b_2$ := false;
              ......
          }
```

We say that a variant of Peterson's mutual exclusion algorithm *satisfies the mutual exclusion property* if there is no execution such that both processes are in the critical section at the same time (i.e. the location (cs_1, cs_2) in the interleaving of the program graphs is unreachable).

Analyze for each variant formally if the mutual exclusion property is satisfied.

We propose the following approach.

- Construct the interleaving of the program graphs for both processes.
- Translate the interleaving into a transition system (reachable states are sufficient).
- Check if there is some reachable state in which both processes are in the critical section.

In this exercise, you only have to draw the interleaving and the transition system if the variant satisfies the property. In case the property is violated it is also sufficient if you explain an execution that shows the violation.

Exercise 3: Relations and Sets
A k-ary relation is a set of k-tuples.

(a) Given a set U (for "universe"), the subset relation between subsets of U is a binary relation over the powerset of U. Define this relation as a set.

(b) A unary relation is a set of 1-tuples, i.e., a set of elements. We also call a unary relation a "property". So $even(x)$, $x \in even$, and "x is an even number" are different ways to express the same thing.
Give an example of a unary relation over the powerset of a given set U. State the relation as a property (in natural language).

(c) Given the set U, express the property "r is a transitive relation over U" as a relation (over a new universe which you have to define first).

Exercise 4: Feedback
What don't you like about the lecture? Please list one or two points.
What do you like about the lecture? Please don't list more than 12 points ;-)