"Model checking"

Prof. Dr. Marcel Kyas

Assignment 10, February 3, 2010

Freie Universität Berlin

Exercise 28 (6 Points) Consider an elevator system that services $N > 0$ floors numbered 0 though $N - 1$. There is an elevator floor at each floor with a call button and an indicator light that signals whether or not the elevator has been called. In the elevator cabin there are N send buttons (one per floor) and N indicator lights that inform to which floor(s) the elevator is going to be sent. For simplicity, restrict to the case $N = 4$. Present a set of atomic propositions — try to minimise the number of propositions — that are needed to describe the following properties of the elevator system as CTL formulae and give the corresponding CTL formulae:

- The doors are "safe", i.e. a floor door is never open unless the cabin is present at the given floor.
- The indicator lights correctly reflect the current requests. That is, each time a button is pressed, there is a corresponding request to be memorised until fulfilment (if ever).
- The elevator only services the requested floors and does not move when there is no request.
- All requests are eventually satisfied.

Exercise 29 (12 Points) Consider the mutual exclusion algorithm by the dutch mathematician Dekker. There are two processes P_0 and P_1, two Boolean-valued variables b_0 and b_1 whose initial values are **false**, and a variable k which may take the values 0 or 1 and whose initial value is arbitrary. The ith process ($i = 0$ or $i = 1$) may be describes as follows:

```
for (;;) {
  b[i] = true;
  while (b[1-i]) {
    if (k == 1-i) {
      b[i] = false;
      while (k = 1-i) ;
      b[i] = true;
    }
  }
  /* Critical section */
  k = 1-i;
  b[i] = false;
}
```

Questions:

1. Model Dekker's algorithm in NuSMV.
2. Verify whether this algorithm satisfies the following properties:
   (a) Mutual exclusion: two processes cannot be in their critical section at the same time.
   (b) Absence of individual starvation: if a processes wants to enter its critical section, it is eventually able to do so. *(Hint: use the* `JUSTICE running` *statement in your NuSMV specification to prohibit unfair executions that might violate this requirement).*

Exercise 30 (12 Points) Model and analyse the following gossipping girls problem in UPPAAL.

A number of girls initially know one distinct secret each. Each girl has access to a phone which can be used to call another girl to share their secrets. Each time two girls talk to each other they always exchange all secrets with each other (thus after the phone call they both know all secrets they knew together before the phone call). The girls can only communicate in pairs (no conference calls) but it is possible that different pairs of girls talk concurrently.

Your tasks are the following:

- Model the problem as a network of UPPAAL timed automata and use UPPAAL to find the minimum number of phone calls needed for the four girls to learn all secrets.
- Redefine the model so that each phone call lasts exactly 60 seconds (for simplicity this time duration is irrelevant for the number of exchanged secrets). Find the minimum time to solve the gossipping girls problem.
- Experiment with the UPPAAL search options breadth-first and depth-first search, upper and lower approximations, and with the diagnostic trace settings fastest and shortest. Try to solve the problem for five girls.

Hints:

- Design a single template for all girls.
- For each girl, you may choose to remember the currently known secrets either in a local array of booleans or using an integer variable (use a binary encoding such that if a girl knows the secrets of e.g. girls one and three and not of two and four, the value in the integer variable will be $0101_2 = 5$; you might the operation | for bitwise OR useful).

Handing in this Assignment Please submit Your solution no later than February 10, 2010, 18:00 (before the tutorial).

The models shall be placed in a directory that carries the last name of one of the group members. Add a `README` file, or better, a `Makefile`, that explains or automates the modelling and checking procedures. Explain, how to interprete the results of model checking in an accompanying PDF or ASCII file.

Put all this into a tape archive that shares the name with the directory and send it by e-mail to marcel.kyas@fu-berlin.de. Use "Model checking 09 Series 10 *your last names*" as the subject line.