Exercises on Model Checking

2007-05-25 Martin Fränzle, Andreas Eggers

1 Basics of symbolic model checking

John Nerd has implemented a special 4 bit shift register in which the bit that is shifted out of the register to the right is negated and shifted in again on the left. The transition function of the four bits are thus:

$$\begin{aligned} b_3' &= \neg b_0 \\ b_2' &= b_3 \\ b_1' &= b_2 \\ b_0' &= b_1 \end{aligned}$$

He is convinced that his circuit can reach only states in which at most three bits are set to active when it is initiated with exacly one bit being set.

Task 1

Check whether this assumption is correct by

- building a model $\mathcal{M}$ of the shift register in NuSMV,
- formulate the property as a CTL formula ϕ,
- and use NuSMV to check $\mathcal{M} \models^? \phi$.

2 Model checking for planning

Finding "error traces" with model checking can also be used not to actually find errors but plans e.g. in scheduling or pathfinding problems. In order to achieve this, a model $\mathcal{M}$ is created of which the set of traces is exactly the set of possible action sequences of the problem that is to be solved. Afterwards a temporal formula ϕ is generated that expresses the searched problem solution. The model checker is used to check $\mathcal{M} \models^? \neg\phi$. If its answer is $\mathcal{M} \models \neg\phi$ then we know that no solution exists that satisfies the formulated requirements and thus the planning problem itself is infeasible. Otherwise a counter-example is found which is an action sequence that follows the rules of $\mathcal{M}$ and satisfies ϕ. It is therefore a solution to the original planning problem.

You will now use this approach to solve the problem of a farmer that has to transport a wolf, a goat and a cabbage over a small brook. His boat offers space only to him and one additional object. He also has to be aboard as none of the objects he wants to transport has the ability to steer the boat. What makes this situation problematic is that neither goat and wolf nor cabbage and goat may ever be left alone on one of the shores.

Task 2:

Model the problem as a model $\mathcal{M}$ in NuSMV. Use a variable that can take three different values for each of the entities (farmer, wolf, goat, cabbage). The value of that variable can be one of the two shores or the boat and denotes where the entity is at that specific point of time. The boat itself needs not to be modeled by such a variable because it is always where the farmer is. The transition system of your model shall reflect all possible changes, i.e. all movements of the boat with the farmer and at most one "passenger" without leaving one of the dangerous combinations behind.

Task 3:

Write the solution property as a CTL formula ϕ and use NuSMV to find a solution for it. Interpret the result.

3 Fairness: Alternating bit protocol

The alternating bit protocol (ABP) is used to send messages over unreliable channels, which can lose messages but not change them. If a message gets lost, the channel duplicates the previous message. In order to ensure a reliable communication over such channels, the ABP appends an alternating bit to the message and uses a return path on which the last received check bit is communicated back to the sender. Provided that all channels lose only a finite number of packets, this mechanism together with the repetition of messages that have not been answered suffices to reproduce the original order of the messages.

In more detail:

1. The system consists of four *asynchronous* components: a sender, a receiver, a message channel from sender to receiver and a Boolean return channel in the opposite direction.
2. The sender transmits in each message a value $x \in \{0, \ldots, 15\}$ and a check bit $c \in \{0, 1\}$ to the channel. The valuations of x and c are defined as follows:
   (a) If the previous message has been received correctly, a new message $x' \in \{0, \ldots, 15\}$ is generated nondeterministically[1]. In this case $c' = 1 - c$ is alternated and x' and c' are passed to the channel.
   (b) If it is unknown whether the previous message was received correctly, $x' = x$ and $c' = c$ are repeated and thus the old values given to the channel.

   In order to check whether the previous message has arived at the receiver, the sender checks the return channel in each step. The sender regards a message as safely transmitted if the bit on the return channel equals the previously sent c.
3. The receiver accepts a message from the channel as soon as it performs a step and detects that the check bit has changed. As long as c is constant, the message is ignored even if its contents has changed.

   The receiver always transmits the check bit of the last message that it considered safely received. As long as no new message is received, this bit is thus repeated.
4. In each of its steps the message channel can decide to either copy its inputs to the output (i.e. to transmit the message correctly) or not to change its output and thus to drop the message. It cannot change neither message nor check bit and it can also not decide to transmit only one of them and drop the other.

   Each combination of message and check bit can be dropped only a finite number of times.
5. The return channel can either copy its input that is connected to the receiver component or can keep the old value. Each check bit can be dropped only a limited number of times, too.

Task 4:

Model this system with four asynchronous processes and suitable fairness conditions. Prove that

1. each message that has been sent finally arrives
2. each message that arrives has been sent before (hint: it is advisable to prove a stronger property instead)
3. the sender can send an infinite amount of messages independent of the channel's actions.

[1] This ensures that we overapproximate all possible behaviours of the sender process.