Formal Methods - lab exam
Facsimile di prova

Exercise 1

Alphametics are cryptarithms that spell out words. Given a mathematical expression, every digit in the expression is replaced by a letter. Five rules govern *alphametics*:

1. Identical digits are replaced by the same letter.
2. Different digits are replaced by different letters.
3. After replacing all the letters with digits, the resulting arithmetic expression must be mathematically correct.
4. Numbers cannot start with 0. For example, the number 0900 is illegal.
5. The problems are in base 10. This means that the letters replace some or all of the 10 digits – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Use a SAT/SMT solver to analyze the following *alphametic* problem:

```
V I O L I N +
V I O L I N +
  V I O L A =
-------------
    T R I O +
S O N A T A
```

In particular:

- Is there a valid solution to this problem? If yes, provide the obtained solution in a comment.
- Is the solution obtained (if any) unique? Extend the model to quickly check this property. If the solution is not unique, reports the newly obtained solution in a comment. If it is unique, clearly state it in a comment.

Exercise 2

Model a simple *alarm* system installed in the *safe* of a bank using nuXmv. Its properties are:

- The *alarm* system can be activated and deactivated using a `pin`.
- After being activated, the *alarm* system enters a `waiting` period of 10 seconds, time that allows users to evacuate the *safe*. After this amount of time the *alarm* is `armed`.
- The *alarm* detects an `intrusion` when someone is inside the *safe* and the *alarm* is `armed`. When an intruder is detected the *alarm* enters a `waiting` period of 5 seconds to allow the intruder to deactivate the *alarm* using the `pin`.
- If the *alarm* is not deactivated after an intrusion is detected, it will `fire`. The *alarm* remains `fired` until deactivation.

The alarm system is comprised by:

- `state` variable, with domain { OFF, EVACUATE, ARMED, INTRUSION, FIRED };
- `s_clock` variable with domain equal to 0..59. Initially, `state` is OFF and `s_clock` is 0.

The alarm system has two boolean inputs:

- `sensor`: `true` iff a person is detected inside the safe
- `use_pin`: `true` iff the pin is being used.

Express the fact that a person must be inside the safe to use the pin as an *invariant* of the inputs.
The *alarm* changes state according to this **ordered** set of rules:

- if the state is OFF and the pin is used, then the next state is EVACUATE
- if the pin is used, then the next state is OFF
- if the state is EVACUATE and the internal clock is 0, then the next state is ARMED
- if the state is ARMED and a person is detected in the safe, then the next state is INTRUSION
- if the state is INTRUSION and the internal clock is 0, then the next state is FIRED
- otherwise, the state does not change

The value of `s_clock` is set to 10 when the state value changes from OFF to EVACUATE, and it is set to 5 when the state value changes from ARMED to INTRUSION. Otherwise, its value is decreased by one unit at each transition until it reaches 0.

In addition, encode the following LTL properties, and verify that they are true:

- if the input pin is never used, then the alarm state is always OFF
- it is always true that, whenever an intrusion is detected then sooner or later the alarm state will be either OFF or FIRED
- it is always true that "if the alarm is armed in a certain state s_k, but the pin is never used starting from s_k onward, then it is necessarily the case that either the sensor won't detect any intruder (starting from s_k onward) or the alarm will eventually fire"
- if the state of the alarm is infinitely often equal to EVACUATE, then someone must enter the safe infinitely often

PS: Next week I will upload the solution to both exercises, so that you have an idea of what type of models I expect you to provide. If you have some issues regarding the nuXmv exercise don't worry, we will see something similar (with the same reasoning about the definition of the model) on the next laboratory.