Software Verification – assignment 2: Linear Temporal Logic (part 2)

In the previous assignment you have generated the Büchi automaton for LTL properties. In this assignment you will check if a model satisfies the property by generating and exploring the cross product of the model and the Büchi automaton.

1 Exploring the Cross-Product

The cross product of a model with a Büchi automata that represents the negation of a property is accepting if the property can be violated. A Büchi automata accepts if there is a cycle that goes through an accepting state: thus, to check whether the property can be violated, we need a way to check for cycles.

One way to check for accepting cycles is the cyan/blue/red nested DFS method by Schwoon et al. [1] which you've also seen in the lectures. **Implement the nested DFS algorithm** in the function `hasCycle` (in the file `buchi.py`). In particular, you can find the relevant part in the 5th page of the paper in "Fig.3". The set "A" in the paper is the set of accepting states. In python, you can use a dictionary to store the values of the colors of the states.

The file `stoplight.py` contains a small example model (from the book); it is a traffic light with an optional power-saving mode, which can turn the light off while it is red. If your cycle check is correct, the regular traffic light should guarantee G F(*green*); but the eco-friendly version should report a cycle.

1.1 Questions

In the nested DFS algorithm shown in this section, there are two checks that report a cycle.

Question 1. The check in the blue search reports a cycle if a successor of a cyan-colored node is itself cyan. Is using this check alone sufficient to find an accepting cycle, if it exists? If not, give a counterexample.

Question 2. The check in the red search reports a cycle if a successor of a node that has been seen in the red search is cyan. Is using this check alone sufficient to find an accepting cycle, if it exists? If not, give a counterexample.

Question 3. If only one of the checks is sufficient to find all accepting cycles, why does the algorithm include the other?

2 Results

Write a short report answering the above questions and make a compressed archive of the report together with your source code.

- The archive must be in `.zip` or `.tar.gz` format.

- Be sure to include your name and student number in the report.
- Please run `make clean` before making the archive.

References

[1] Schwoon, S.; Esparza, J. *A Note on On-The-Fly Verification Algorithms*. Lecture Notes in Computer Science, vol. 3440 (2005): 174–190.