Projet Logique
2014/15
Première partie bis

Stefan Schwoon

Introduction

In the first part, we used SAT solving to solve Reflexion puzzles, however with one relaxation of the problem: It was not necessary to ensure that the path is acyclic, so the solver for this exercise may come up with false solutions consisting of one path from `S` to `E` and additional cycles that together collect all the treasures, like in the figure below.

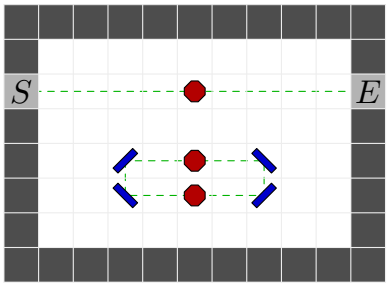

Figure 1: A false solution allowed in the first part of the exercise.

The second part of this project is to eliminate this weakness by making use of the SMT (satisfiability-modulo-theory) capabilities of the Z3 solver.

1 Objectives

The objectives of the second part of this exercise is as follows:

(i) Make your solver generate a formula in the SMT2 language of Z3 (see below).

(ii) Extend the translation to ensure that all diamonds are collected by a single acyclic path.

(iii) Use the SMT solver to check whether a solution exists, and output it in the same way as in the first part.

Note (1): The "pure" SAT solving approach we used so far is more efficient than SMT solving, and the acyclicity constraint is not required for all puzzles. Therefore, it is not desirable to throw the translation from the first part away, and you have the following options:

- Your solver may provide translations both to DIMACS *and* SMT2, for the two parts of this exercise.
- You can make two different solvers (not really recommended because of additioanl maintenance effort).
- You are authorized to provide only an SMT2 translation, for a minor deduction in the mark (1-2 points).

Note (2): Correctly implementing the acyclicity constraint permits you to obtain not only the correct setting for the mirrors, but also the path itself. Moreover, you find a package for graphical output of the puzzles on the project webpage, which will permit you to display the path. Implementing this functionality is optional but gives a bonus.

2 Organisation and Evaluation

The comments from the previous exercise sheet concerning the organisation and evalution of the project apply. The *soutenance* for both parts of this project takes place during the session on March 6, and the deadline for handing in your solutions is still March 4 until midnight.

3 Getting started with Z3

It is recommended to read the start of the Z3 tutorial, linked on the project website. It is not necessary to read the entire page, just the sections *Basic commands* and *Propositional logic*. We are going to use only a small (decidable) part of the theories supported by the solver, i.e. constraints on integers and constants with inequality.

Following is a summary of said tutorial for our purposes:

- Run the solver with either `z3 -in` (when your commands are coming from the standard input) or `z3 myfile` when they are contained in a file.
- Variables must be declared. We will have a mixture of Booleans and integers, declared as follows:

```
(declare-const a Bool)
(declare-const x Int)
```

- Your formula will be stated as a series of assertions, which must all be true. For instance, a formula $a \vee b \vee \neg c$ (e.g., a clause in a CNF formula, where a, b, c are all Booleans) is expressed as follows:

  ```
  (assert (or a b (not c)))
  ```

  Requirement that integer x must be smaller than integer y and that integer z must be 0 are expressed as follows:

  ```
  (assert (< x y))
  (assert (= z 0))
  ```

  Assertions may be nested and mix constraints on Booleans and integers, e.g. in the following way, where `=>` represents implication:

  ```
  (assert (=> (and a b) (< x y)))
  ```

- Finally, two commands must be added to check satisfiability of your assertions and to obtain a satisfying model:

  ```
  (check-sat)
  (get-model)
  ```

  Note that if the satisfiability check fails, then `(get-model)` causes an error message, which you can safely ignore. Attention: The models printed by the solver sometimes do not always include all variables, e.g., if their values are deemed inconsequential to the satisfaction of the formula (in which case you can set them to some default value).