Automated Reasoning IMC009
Practical Assignment

The assignment

The assignment consists of two parts, which together determine half of your final grade (the other half is determined by the exam). The first part counts for 20% of the assignment grade. We highly recommend you taking this part seriously, as the practice and feedback will help you in the second part. The second part will be more difficult and contributes for the remaining 80%. Each part has to be executed by one or two persons. It is not mandatory to use the same group twice. The solutions should be described in a report, as a PDF of at most 20 pages and, together with your Python code, must be submitted in Brightspace. Every report has to contain the name, student number and email address of each of the authors.

Grading

In the assignments you will construct logical formulas, describing the solution of some SMT problem. *For every such a formula we require a clear and concise description, explaining how the formula corresponds to (a certain part of) the solution of the problem.* An example problem with solution, showing the appreciated style, is given on the next page of this document.

We will take into account the following aspects when grading your reports

- For every requirement we expect a concrete SMT formula (a wordy description of how the formula should look like is not sufficient).
- Explanation is way more important than implementation. A correct Python implementation without any explanation in your report will yield zero points! On the other hand, you can still earn a considerable amount of points by providing a coherent reasoning that correctly describes the formulas that have to be implemented.
- Clear, concise and generic descriptions are appreciated, both of the formulas themselves and their explanation. The meaning of each introduced variable must be *explicitly defined* (check out the example report to see how the meaning of p_{ij} is defined in terms of i and j). Formulas of more than half a page should not be contained in the report; instead the structure of the formula should be explained.
- Python code does not need to be included in the report. Results from Z3 should be included in the report and you must explain how they answer the questions of the corresponding exercise. Results should be easy to check and therefore have to be presented in an organized manner, for example by using tables, pictures, etc. If we are not able to decipher your results then we cannot give you credits for it.

Implementation

Solutions should be implemented using the Z3 API in Python. A tutorial on how to use the Z3 API in Python can be found on https://ericpony.github.io/z3py-tutorial/guide-examples.htm. All exercises can be implemented by only using the functionalities as shown in this tutorial.

Example problem with solution

Authors:
Cynthia Kop s123456 `c.kop@cs.ru.nl`
Sebastian Junges s654321 `sjunges@cs.ru.nl`

Problem: eight queens problem

Find a way to put eight queens on a chess board such that no two queens share a column, row or diagonal.

Solution

We generalize this problem to putting n queens on an $n \times n$ chess board, for any $n \geq 1$, with the restriction that no two share a column, row or diagonal.

For each row $i = 1, \ldots n$ and column $j = 1, \ldots, n$ we introduce a boolean variable p_{ij}, whose meaning is defined by

$$p_{ij} \text{ is true } \iff \text{ there is a queen on position } (i, j)$$

At least one queen in every row/column As we have to put exactly n queens, and no two are allowed to be on the same row, every row i should contain at least one queen. That is, for each i there is at least one j such that p_{ij} is true. This is expressed as

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} p_{ij}.$$

In a similar way every column j should contain at least one queen

$$\bigwedge_{j=1}^{n} \bigvee_{i=1}^{n} p_{ij}.$$

At most one queen in every row/column Every row i contains at most one queen, that is, for every two distinct positions j, k it is not allowed that both p_{ij} and p_{ik} are true. This is expressed as

$$\bigwedge_{i=1}^{n} \bigwedge_{1 \leq j < k \leq n} (\neg p_{ij} \vee \neg p_{ik})$$

Similarly, every column j should contain at most one queen

$$\bigwedge_{j=1}^{n} \bigwedge_{1 \leq i < k \leq n} (\neg p_{ij} \vee \neg p_{kj})$$

Diagonals Two positions (i,j) and (k,m) are on the same diagonal in the one direction if and only if $i+j=k+m$, and they are on the same diagonal in the other direction if and only if $i-j=k-m$. For every pair of such positions it is not allowed that they are both occupied by a queen.

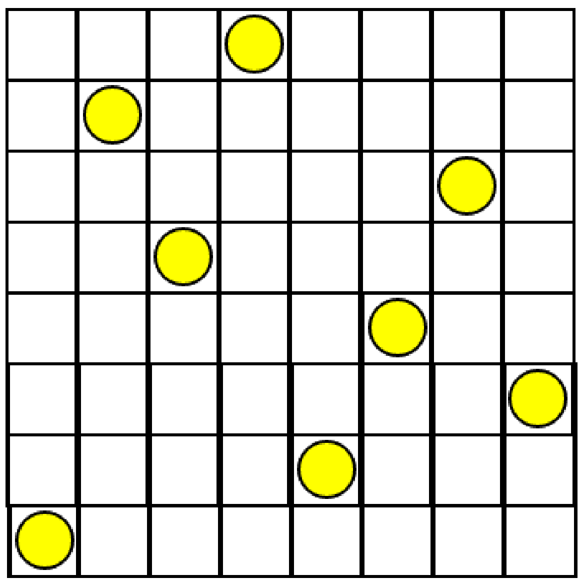
This can be phrased as follows: for every two non-equal rows i and k we require that, if $i+j=k+m$ or $i-j=k-m$, then at least one of p_{ij}, p_{km} is false. Expressed as a formula:

$$\bigwedge_{1\leq i<k\leq n}\ \bigwedge_{i+j=k+m \text{ or } i-j=k-m} (\neg p_{ij} \vee \neg p_{km})$$

The total formula is the conjunction of all previous formulas.

Results

The p_{ij} that are true are given by $p_{14}, p_{22}, p_{37}, p_{43}, p_{56}, p_{68}, p_{75}, p_{81}$. All others are false. Shown in a picture:



Indeed, there are eight queens and no two are on the same row, column or diagonal.