Knowledge Based AI - Programming assignment 2

The goal of this assignment is to get a thorough understanding of the hitting set algorithm and system diagnoses. For this you must implement the hitting set algorithm as discussed in the lecture and apply it on the provided conflicts sets. You must also implement multiple heuristics that will be used when creating the tree.

There is code available in Python and Java for reading a fault tree and generating its conflict sets. There is already a structure given for you to implement the algorithm. You can change anything you want about the provided code but do mention it in the final report.

To help you get an overview, here is the assignment broken down in parts:

0. Become familiar with the given code and understand the hitting set algorithm. What are hitting sets? What are conflict sets? How do I generate them using the provided code? Etc.
1. Find a good representation for the tree structure used in the algorithm.
2. Find all hitting sets.
3. Find all minimal hitting sets.
4. Experiment with multiple heuristics (so at least two) for creating the tree. Show how these affect the algorithm's correctness and runtime complexity.

Bonus points are offered for creating your own circuits and testing with more than two different heuristics. The exact grading can be found in the grading sheet.

Be sure to use informative names for variables, constants, and functions, and to document your code well. You are NOT allowed to use external libraries (e.g. NumPy, Pandas, PySAT) when implementing the hitting set algorithm. For collecting the conflict sets, you are only allowed to use z3-solver. You are always allowed to use the standard libraries (e.g. queue, collections and random).

Report

Write a short scientific report on your implementation, describing the project, your code, your experiments, their results, and your interpretation & conclusions. You can use the various parts of the assignment to help you structure the report. For a full list of elements to include in your report, check out the "writing a programming report" section at the course guide (learning task 0) and the assessment form for this assignment. We expect a formal (!) report of ca. 4 – 8 pages (not counting possible appendices)

To hand in

Both a report (pdf) and the code (zip). Do not forget to list your name(s), student number(s), course, number of the task, date, etc. in your code and report.

The deadline of this assignment will be communicated through Brightspace.

Programming questions can be sent to the teaching assistants either during the practical sessions (preferably) or by email. Their email addresses can be found in the course manual. Content-wise questions can be sent to the lecturer (johan.kwisthout@donders.ru.nl).