

# — Exercise 4 —

**Deadline: TODO**

## Task 1 - Basic Solver

Start with the implementation of your data structures and algorithm. Make sure to have a first working solution (no wrong outputs) latest in three weeks.

This version does neither need to generate *infeasible subsets* (use `generateTrivialInfeasibleSubset` for the time being) nor work incrementally. Your module should however generate a model.

## Task 2 - Testing

Test your implementation using the test cases you created in the first exercise. Additionally you can use some of the examples from the SMT-LIB benchmarks for the respective logic (QF\_UF<sup>1</sup>).

We will also test your implementation regularly on the SMT-LIB benchmarks and send you the corresponding results.

## Delta

If your solver does not work correctly *delta* might help you to find your bug by reducing the size of the input on which an error occurs. There are two ways to use *delta* depending on whether a `segfault` or a `wrong` result occurred. In case the input causes a `segfault` you can add a new exit code using `std::quick_exit(x)` with some number `x` at the place in your code that caused the incomplete execution. Then you execute

```
./delta <file>
```

If your solver computes a wrong output remove the line (`set-info :status ...`) from your input file and execute

```
./delta -s ../../src/delta/scripts/result-incorrect.py <file>
```

Note that `result-incorrect.py` requires some reference solver that we assume to be correct, usually `z3`. Please install `z3` and adapt the python script if necessary. The exit code of this script should be 1, indicating that the solvers disagree. If the exit code is 0 the solvers agree (there is no error) and 2 means that one of the solvers returned an unexpected output, usually a `z3` parser error.

`delta` recognizes the error condition (the exit code of the solver or the script) and tries to simplify the input file with small changes, always verifying that the exit code is retained. Once no further change can be made, `delta` saves the resulting file as `delta.out.smt2`. Then you can run your solver on that file instead, to hopefully find your bug easier.

---

<sup>1</sup>[https://clc-gitlab.cs.uiowa.edu:2443/SMT-LIB-benchmarks/QF\\_UF](https://clc-gitlab.cs.uiowa.edu:2443/SMT-LIB-benchmarks/QF_UF)