Logical reasoning and programming, lab session 5

(October 18, 2021)

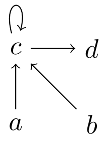
The following exercises require an SMT solver. For simplicity, you can use

- ~~an online version of Z3~~[1], or
- an online version of CVC4[2],

or both. Even better, you can install Z3 or CVC4 yourself. Another option is to use pySMT, a convenient way how to experiment with various SMT solvers in Python. ~~If you want to learn a bit more about the Z3 prover, you should start with this tutorial.~~ Moreover, if you want to play with the Z3 prover in Python, check Programming Z3. However, if you want to experiment with SMT solvers in Python, you should try pySMT.

5.1 Check API documentation of PySAT. There are various useful things, for example, `IDPool`, `enum_models`, `get_core`.

5.2 We have a language that contains only one binary predicate symbol $\in$ and we have an interpretation $\mathcal{M} = (D, i)$ such that $D = \{a, b, c, d\}$ and $i(\in)$ is given by the following diagram:



Meaning that $x \in y$ iff there is an arrow from x to y. Decide whether the following formulae are valid in $\mathcal{M}$:

(a) $\exists X \forall Y (\neg(Y \in X))$,

(b) $\exists X \forall Y (Y \in X)$,

(c) $\exists X \forall Y (Y \in X \leftrightarrow Y \in Y)$,

(d) $\exists X \forall Y (Y \in X \leftrightarrow \neg(Y \in Y))$.

5.3 Decide whether it is satisfiable in the theory of uninterpreted functions that

$$x = f(f(f(f(f(x))))) \wedge x = f(f(f(x))) \wedge x \neq f(x).$$

5.4 Is it possible to decide whether $\forall X(f(f(X)) = g(X)) \wedge f(g(a)) \neq g(f(a))$ is satisfiable by our congruence closure algorithm?

5.5 How can we extract a solution for Difference logic if there is no cycle in the graph?

5.6 Try all the examples in the SMT-LIB Examples.

5.7 Show that $x - y > 0$ iff $x > y$ holds for integers, but does not hold for bit-vectors with a fixed length.

[1] It seems that rise4fun is and will be down, see here.

[2] There is also a new version called CVC5 available.

5.8 Let x be a 32 bit-vector. You want to verify that if you do $x \gg_s 31$ (arithmetic right shift is `bvashr`) followed by one of the following

- $(x \oplus y) - y$, or
- $(x + y) \oplus y$, or
- $x - ((x + x) \& y)$,

then you get the absolute value of x.

5.9 Try CBMC, using Z3, on `f14` from this example. For details, see these lecture notes.

5.10 Check the Static Single Assignment (SSA) example in these slides.

5.11 You can find many examples in Dennis Yurichev's SAT/SMT by Example.