IA085: Satisfiability and Automated Reasoning

Seminar 1

Exercise 1 *Which of the following formulas are satisfiable? If the formula is satisfiable, find its model, if it is unsatisfiable, explain why.*

1. $(A \to B) \wedge (B \wedge C \to \neg A) \wedge (\neg C \leftrightarrow A)$
2. $(A \vee B) \wedge (\neg A \vee C) \wedge (\neg C \vee \neg A) \wedge (\neg C \vee \neg B) \wedge (\neg B \vee C)$
3. $(A \wedge B \wedge \neg C \wedge \neg D) \vee (\neg A \wedge B \wedge \neg C \wedge D) \vee (\neg A \wedge \neg B \wedge \neg C \wedge D) \vee (\neg A \wedge B \wedge C \wedge \neg D) \vee (A \wedge \neg B \wedge \neg C \wedge \neg D)$

Exercise 2 *Which of the following logical entailments hold? If the entailment holds, explain why, if it does not, find a counterexample.*

1. $(A \to B) \wedge \neg B \models \neg A$
2. $(A \to B) \wedge A \wedge \neg B \models C \wedge \neg C$
3. $(A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee C) \wedge (\neg A \vee B \vee C) \wedge (\neg A \vee \neg B \vee \neg C) \models \neg(A \wedge B)$
4. $(A \to (B \vee C)) \wedge (A \vee B) \wedge (\neg B \to C) \models B$

Exercise 3 *Convert the following formulas to the conjunctive normal form using* Tseitin transformation.

1. $\neg(A \wedge B) \vee (B \wedge C \wedge D) \vee \neg A$
2. $\neg(A \vee (B \to C)) \wedge (A \leftrightarrow (B \to C))$

Exercise 4 *Let a_0,a_1,a_2, b_0,b_1,b_2 be Boolean variables that represent bits of three-bit numbers $a = a_2a_1a_0$ and $b = b_2b_1b_0$. Consider the standard addition operation $+$ on binary numbers that throws away the potential overflow bit (i.e., $+$ is addition modulo 2^3). Write a formula that is satisfiable if and only if the sum of the two numbers is five, i.e.,*

$$a_2a_1a_0 + b_2b_1b_0 = 101.$$

Hint: introduce new auxiliary variables.

Exercise 5 *Install `PySAT` library and run the following Python script.*

```
from pysat.solvers import Solver

with Solver() as s:
    s.add_clause([-1, 2])
    s.add_clause([-2, 3])

    print(s.solve())
    print(s.get_model())
```

Exercise 6 *Using* `PySAT`*, write a program that gets a number n as a command line argument and solves the n queens problem. Text output of the positions in the solution is fine, but if you are feeling fancy, draw the resulting chess board in the terminal.*

BONUS: *If you have spare time, you can try out and compare different encodings of the* at-most-one constraint[1].

[1] https://manuscriptlink-society-file.s3-ap-northeast-1.amazonaws.com/kism/conference/sma2020/presentation/SMA-2020_paper_77.pdf

Exercise 7 *Modify the script from Exercise 5 so that the input formula is a CNF encoding of the formula you designed in Exercise 4. Modify the script so that it prints out not one model of the formula, but* two different models.

Modify the script so that it prints out all models of the formula.