

Exercise 1: (Resolution Proof) [3 Points]

Construct a resolution proof of unsatisfiability for the following formula

$$(x_3 \vee x_4 \vee \bar{x}_1 \vee x_5) \wedge (\bar{x}_3 \vee x_4 \vee x_5) \wedge (x_3 \vee \bar{x}_4 \vee \bar{x}_1) \wedge (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_5) \wedge (\bar{x}_3 \vee \bar{x}_4 \vee x_5)$$

Exercise 2: (Hidden Horn \subseteq SLUR) [5 Points]

Prove that every hidden Horn formula is a SLUR formula, i.e., that the SLUR algorithm would never give up if its input is a hidden Horn formula.

Exercise 3: (Recognizing Hidden Horn) [6 Points]

Let F be an arbitrary CNF formula. Let R_F be a 2-SAT formula that contains the clause $(l_1 \vee l_2)$ if and only if there is a clause $C \in F$ such that $l_1 \in C$ and $l_2 \in C$ (See slide 6 of lecture 3 for an example). Prove that if R_F is satisfiable, then F is a hidden Horn formula.

Exercise 4: (Tseitin Encoding) [4 Points]

Encode the following formula into CNF using the Tseitin Encoding.

$$(\bar{x}_1 \wedge \overline{(x_3 \iff x_2)}) \vee ((x_3 \implies \bar{x}_4) \wedge (x_1 \implies (x_2 \wedge \bar{x}_3))) \wedge (x_4)$$

Exercise 5: (Logic Puzzle) [7 Points]

Solve the following puzzle. How would you solve it with the help of SAT solver?

Melanie, Nora and Pia opened a wellness business. The first week is over, each of the three women had every day between 4 and 11 clients. **How many clients exactly did each woman have on each day (Monday - Friday)?** if the following is true:

1. Melanie had more clients on Wednesday than on Monday.
2. On the day, when Pia had one more client than on Tuesday, Melanie had one more client than on Friday.
3. Nora had 5 clients on one day, it was not Thursday.
4. On the day, when Nora had one more client than on Wednesday, Melanie had one more than on Thursday.
5. On Monday both Melanie and Nora had more clients than Pia.
6. On the day (which was not Friday), when Nora had one more client than on Tuesday, Pia had one more client than on Wednesday.

Exercise 6: (Planning Encoding) [7 Points]

Design and describe a SAT encoding for SAS+ planning that uses only one kind of Boolean variables – variables representing actions ($a_i^t = True$ if and only if a_i is present in the t -th step of the plan). Compute how many variables and clauses are required for such an encoding (HINT: you may need to add some additional helper actions to the planning problem before you can create the required encoding).

Exercise 7: (Tetris Challenge) [8+(8) Points]

Design and implement a SAT encoder for the Tetris puzzle in the programming language of your choice. The tetris puzzle is to arrange a set of pieces into a rectangular block of a given size. There are 5 kinds of pieces: I, T, S, L, and O (see Figure 1) and they **cannot be rotated**. The block does not have to be completely filled. One example of a puzzle instance could be: arrange one I, two of each T, L, S and three O pieces into a 5x8 block. This instance has a solution and it is given in Figure 1. Another instance (one of each block and two of "I"s in a 5x5 block) has a solution with one empty tile (see Figure 1).

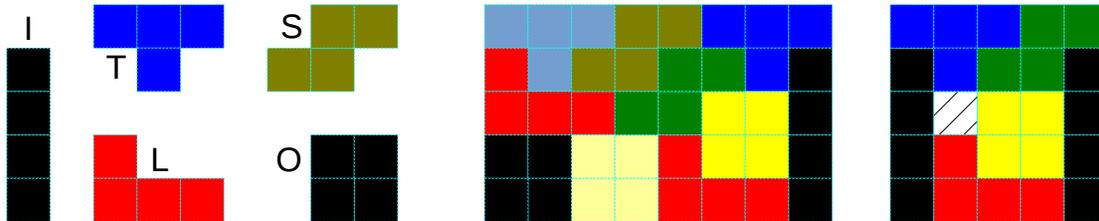


Figure 1: The 5 tetris pieces and two solved puzzles of size 5x8 and 5x5 blocks

Implement an encoder that takes 7 arguments (block height and width, number of I,T,S,L, and O pieces) and prints a CNF formula that is satisfiable if and only the given tetris puzzle is solvable. The 7 arguments for the two examples in Figure 1 would be "5 8 1 2 2 2 3" and "5 5 2 1 1 1 1". An example of an unsolvable instance could be "4 4 1 1 0 1 1".

The best encoder (producing formulas that are solvable fast with glucose) will get a bonus of 8 points.

Exercise 8: (Sliding Puzzle Challenge) [10(+10) Points]

Implement a DIMSPEC encoder for the sliding puzzle problem. Your application should take a single command line argument – a string with the initial state of the puzzle (example: "123-480-765", goal state is always "123-456-780", "0" represents the empty slot) and print the sequence of steps required to solve it (movements of the empty slot, in the example above: "DLURD"). The puzzle will have at most 15 pieces (0123456789ABCDEF) and is not necessarily a square.

Some example inputs for testing: "023-145", "1034-9267-D5B8-E6FC", "162-530-478", "012-345-678", "876-543-210", you can create your own inputs, be careful not to create unsatisfiable ones :)

Use Stephan's DIMSPEC solver (called incplan in the IPASIR app package: <https://github.com/biotomas/ipasir>) to solve DIMSPEC files. The best solver (solving the most instances and fastest) will get a bonus of 10 points.

Exercise 9: (Sokoban Challenge) [15(+15) Points]

Design and implement a DIMSPEC encoder for Sokoban (<http://en.wikipedia.org/wiki/Sokoban>). You get 15 points for any working encoding. The author of the best encoding receives additional 15 points.

- input/output format: http://www.sokobano.de/wiki/index.php?title=Level_format
- benchmark levels: <http://baldur.iti.kit.edu/sat/files/sokoban.zip>
- DIMSPEC solver: incplan app in <https://github.com/biotomas/ipasir>.