[4] **1** Determine satisfiability of the following formulas using enumerative instantiation if the formula is satisfiable, and both enumerative and E-matching based instantiation if the formula is unsatisfiable

(a)
$$\neg P(a) \ \wedge \ R(b) \ \wedge \ S(c) \ \wedge \ \forall x.R(x) \vee S(x) \ \wedge \ \forall x.\neg R(x) \vee P(x) \ \wedge \ \forall x.P(x) \vee \neg S(x)$$

(b)
$$(a = b \vee f(a) = f(f(b))) \ \wedge \ (g(a) = g(b) \vee g(a) = b) \ \wedge \ f(a) = g(a) \ \wedge$$
$$((\forall x.f(f(x)) \neq x) \vee (\forall x\, y.g(y) = g(x)))$$

[2] **2** Find an example of an unsatisfiable formula where E-matching fails to generate the right instances required for a proof. That is, find an example where E-matching with a non-empty set of instantiation patterns does not produce the correct instances. Show how unsatisfiability can be proved using enumeration-based instantiation instead.

3 Donald Knuth mentions in Section 7 of *The Art of Computer Programming* devoted to bitwise tricks that $((x + 2) \oplus 3) - 2 = ((x - 2) \oplus 3) + 2$ holds for all bitvectors x of length 4, where $\oplus$ denotes exclusive or.

[2] (a) Check whether this is true, using an SMT encoding with quantifiers.

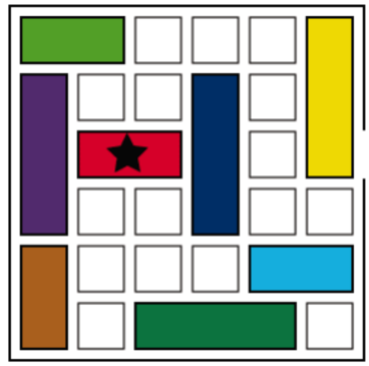
[2] (b) Find all constants a and b such that $((x + a) \oplus b) - a = ((x - a) \oplus b) + a$ holds for all bitvectors x of length 4.

[4] ⋆ **4** Dietz' formula computes the average of two numbers, that is, $(x + y)/2$, *without overflow*:

```
(((y ^ x) >>s 1) + (y & x)
```

Use an SMT encoding to prove that the result is the average for all x and y, and indeed no overflow occurs. Here `^` denotes exclusive or, and `>>s` denotes signed right shift. Choose some reasonable bit width for x and y.

[5] ⋆ **5** In the *Rush Hour* puzzle the goal is to drive the red car out of a jammed parking lot, like in the following picture:



The playing board (i.e., the parking lot) is always a 6×6 grid. A *move* in the game consists of moving one car back or forth onto empty fields (but the cars cannot turn). You can also play online.

Solve the above rush hour puzzle using an SMT encoding and bounded model checking, i.e., by limiting the number of steps.

The following approach might be helpful (though there are many possibilities for an encoding):

- The state, i.e., the current configuration can be described by the x-positions of all horizontal and the y-positions of all vertical cars. So in the example, a state is of the form

$$s = \langle x_{\text{lgreen}},\ x_{\text{red}},\ x_{\text{lblue}},\ x_{\text{dgreen}}, y_{\text{brown}},\ y_{\text{purple}},\ y_{\text{dblue}},\ y_{\text{yellow}} \rangle$$

- It is easy to define a predicate $I(s)$ describing the initial state. In the example, it has to include $y_{\text{brown}} = 1$, $x_{\text{red}} = 2$, etc. (assuming we count from bottom left).
- Define when a cell (X, Y) is *free* in a state s. A cell is free if it is not occupied by any car. For example, (X, Y) is not occupied by the red car if $\neg(x_{\text{red}} \leqslant X < x_{\text{red}} + 2 \wedge Y = 4)$ holds, because $len_{\text{red}} = 2$ and 4 is the (constant) y-position of the red car.
- Using the encoding of a free cell, one can define a predicate $T(s, s')$ to describe a valid transition from state s to state s': In a valid transition every car was moved in a valid way. E.g. for the red car, this can be described as fgollows: if x_{red} in s is different from x'_{red} in s' then the difference cell(s) must have been free in s. (This approach also allows parallel moves, but this is ok.)
- Finally, a success predicate $S(s)$ demanding $x_{\text{red}} = 5$ states that the red car can exit.
- Suppose there are $k + 1$ states $s_0, \ldots, s_k$. There is a solution in at most k steps iff

$$I(s_0) \wedge \bigwedge_{i=0}^{k-1} T(s_i, s_{i+1}) \wedge \bigvee_{i=0}^{k} S(s_i)$$

  is satisfiable.
- One has to make sure that all states are *valid* in the sense that all its variables are between 1 and $6 - len_{\text{color}}$, where len_{color} is the length of the respective car.

Exercises marked with a $\star$ are optional. Solving them gives bonus points if you submit them before the course via OLAT or email.