

## Satisfiability Checking - WS 2023/2024

### Series 2

teaching@ths.rwth-aachen.de  
https://ths.rwth-aachen.de/teaching/

### Exercise 1

We define the famous *Pigeon Hole Problem (PHP)* over  $\mathbb{N} = \mathbb{N} \setminus \{0\}$ :

**Given:**  $n \in \mathbb{N}$ .

**Question:** Do  $n + 1$  pigeons fit into  $n$  holes, if no two pigeons fit into one hole?

- What is the solution to the pigeon hole problems for all  $n \in \mathbb{N}$ ?
- Formulate the pigeon hole problem for  $n = 2$  holes (and thus 3 pigeons) in propositional logic.
- If your formula is not already in CNF, convert it into CNF. Use resolution to deduce the empty clause.
- Specify a preferably small unsatisfiable core of the problem, that is a subset of the clause set that is already unsatisfiable.
- The pigeon hole problems are a “worst-case” for many SAT-solvers. Can you guess why? Substantiate your claims!

*Solution:*

- The assignment due to the pigeon hole problem corresponds to the **existence** of a surjective **mapping**  $\{1, \dots, n\} \rightarrow \{1, \dots, n + 1\}$  or, **equivalently**, the existence of an injective **mapping**  $\{1, \dots, n + 1\} \rightarrow \{1, \dots, n\}$ . Because such a **mapping does not exist** the solution to the pigeon hole problem is “**unsatisfiable**”.
- Let  $x_{ij}$  stand for **pigeon  $i$  being in hole  $j$**  ( $1 \leq i \leq 3$ ,  $1 \leq j \leq 2$ ). The pigeon hole problem can be encoded into the following propositional formula describing that each pigeon is in a hole, no pigeon is assigned twice and no hole holds two pigeons at the same time:

$$\begin{aligned} \varphi_0 := & c_1 : (x_{11} \vee x_{12}) \wedge c_2 : (x_{21} \vee x_{22}) \wedge c_3 : (x_{31} \vee x_{32}) \wedge \text{对于3个鸽子, 必须在一个洞里: 1洞或2洞} \\ & c_4 : (\neg x_{11} \vee \neg x_{12}) \wedge c_5 : (\neg x_{21} \vee \neg x_{22}) \wedge c_6 : (\neg x_{31} \vee \neg x_{32}) \wedge \text{不可同时在1洞或2洞} \\ & c_7 : (\neg x_{11} \vee \neg x_{21}) \wedge c_8 : (\neg x_{11} \vee \neg x_{31}) \wedge c_9 : (\neg x_{21} \vee \neg x_{31}) \wedge \text{不可同时有2个鸽子在一个洞1} \\ & c_{10} : (\neg x_{12} \vee \neg x_{22}) \wedge c_{11} : (\neg x_{12} \vee \neg x_{32}) \wedge c_{12} : (\neg x_{22} \vee \neg x_{32}) \wedge \text{不可同时有2个鸽子在一个洞2} \end{aligned}$$

- First eliminate  $x_{11}$  by resolution.  $x_{11}$  occurs in  $c_1$ ,  $\neg x_{11}$  occurs in  $c_4, c_7, c_8$ .

$$\begin{aligned} & \frac{c_1 : (x_{11} \vee x_{12}) \quad c_4 : (\neg x_{11} \vee \neg x_{12})}{c_{13} : (x_{12} \vee \neg x_{12})} \quad (\text{Tautology!}) \\ & \frac{c_1 : (x_{11} \vee x_{12}) \quad c_7 : (\neg x_{11} \vee \neg x_{21})}{c_{14} : (x_{12} \vee \neg x_{21})} \\ & \frac{c_1 : (x_{11} \vee x_{12}) \quad c_8 : (\neg x_{11} \vee \neg x_{31})}{c_{15} : (x_{12} \vee \neg x_{31})} \end{aligned}$$

Resulting formula:

$$\begin{aligned} \varphi_1 := & \text{false} \wedge c_2 : (x_{21} \vee x_{22}) \wedge c_3 : (x_{31} \vee x_{32}) \wedge \\ & \text{false} \wedge c_5 : (\neg x_{21} \vee \neg x_{22}) \wedge c_6 : (\neg x_{31} \vee \neg x_{32}) \wedge \\ & \text{false} \wedge c_8 : (\neg x_{21} \vee \neg x_{31}) \wedge \\ & c_{10} : (\neg x_{12} \vee \neg x_{22}) \wedge c_{11} : (\neg x_{12} \vee \neg x_{32}) \wedge c_{12} : (\neg x_{22} \vee \neg x_{32}) \wedge \\ & c_{14} : (x_{12} \vee \neg x_{21}) \wedge c_{15} : (x_{12} \vee \neg x_{31}) \end{aligned}$$

Next eliminate  $x_{12}$  occurring positively in  $c_{14}, c_{15}$  and negatively in  $c_{10}, c_{11}$ .

$$\begin{aligned} & \frac{c_{14} : (x_{12} \vee \neg x_{21}) \quad c_{10} : (\neg x_{12} \vee \neg x_{22})}{c_{16} : (\neg x_{21} \vee \neg x_{22}) = c_5} \\ & \frac{c_{14} : (x_{12} \vee \neg x_{21}) \quad c_{11} : (\neg x_{12} \vee \neg x_{32})}{c_{17} : (\neg x_{21} \vee \neg x_{32})} \\ & \frac{c_{15} : (x_{12} \vee \neg x_{31}) \quad c_{10} : (\neg x_{12} \vee \neg x_{22})}{c_{18} : (\neg x_{31} \vee \neg x_{22})} \\ & \frac{c_{15} : (x_{12} \vee \neg x_{31}) \quad c_{11} : (\neg x_{12} \vee \neg x_{32})}{c_{19} : (\neg x_{31} \vee \neg x_{32}) = c_6} \end{aligned}$$

Resulting formula:

$$\begin{aligned} \varphi_2 := & \text{false} \wedge c_2 : (x_{21} \vee x_{22}) \wedge c_3 : (x_{31} \vee x_{32}) \wedge \\ & \text{false} \wedge c_5 : (\neg x_{21} \vee \neg x_{22}) \wedge c_6 : (\neg x_{31} \vee \neg x_{32}) \wedge \\ & \text{false} \wedge c_8 : (\neg x_{21} \vee \neg x_{31}) \wedge \\ & \text{false} \wedge c_{12} : (\neg x_{22} \vee \neg x_{32}) \wedge \\ & \text{false} \wedge c_{17} : (\neg x_{21} \vee \neg x_{32}) \wedge c_{18} : (\neg x_{31} \vee \neg x_{22}) \end{aligned}$$

Next eliminate  $x_{21}$  occurring positively in  $c_2$  and negatively in  $c_5, c_8, c_{17}$ .

$$\begin{aligned} & \frac{c_2 : (x_{21} \vee x_{22}) \quad c_5 : (\neg x_{21} \vee \neg x_{22})}{c_{20} : (x_{22} \vee \neg x_{22}) \quad (\text{Tautology!})} \\ & \frac{c_2 : (x_{21} \vee x_{22}) \quad c_8 : (\neg x_{21} \vee \neg x_{31})}{c_{21} : (x_{22} \vee \neg x_{31})} \\ & \frac{c_2 : (x_{21} \vee x_{22}) \quad c_{17} : (\neg x_{21} \vee \neg x_{32})}{c_{22} : (x_{22} \vee \neg x_{32})} \end{aligned}$$

Resulting formula:

$$\begin{aligned} \varphi_3 := & \text{false} \wedge c_3 : (x_{31} \vee x_{32}) \wedge \\ & \text{false} \wedge c_6 : (\neg x_{31} \vee \neg x_{32}) \wedge \\ & \text{false} \wedge c_9 : (\neg x_{21} \vee \neg x_{31}) \wedge \\ & \text{false} \wedge c_{12} : (\neg x_{22} \vee \neg x_{32}) \wedge \\ & \text{false} \wedge c_{18} : (\neg x_{31} \vee \neg x_{22}) \wedge \\ & c_{21} : (x_{22} \vee \neg x_{31}) \wedge c_{22} : (x_{22} \vee \neg x_{32}) \end{aligned}$$

Next eliminate  $x_{22}$  occurring positively in  $c_{21}, c_{22}$  and negatively in  $c_{12}, c_{18}$ .

$$\begin{array}{c}
 \frac{c_{21} : (x_{22} \vee \neg x_{31}) \quad c_{12} : (\neg x_{22} \vee \neg x_{32})}{c_{23} : (\neg x_{31} \vee \neg x_{32})} \\
 \frac{c_{21} : (x_{22} \vee \neg x_{31}) \quad c_{18} : (\neg x_{31} \vee \neg x_{22})}{c_{24} : (\neg x_{31})} \\
 \frac{c_{22} : (x_{22} \vee \neg x_{32}) \quad c_{12} : (\neg x_{22} \vee \neg x_{32})}{c_{25} : (\neg x_{32})} \\
 \frac{c_{22} : (x_{22} \vee \neg x_{32}) \quad c_{18} : (\neg x_{31} \vee \neg x_{22})}{c_{26} : (\neg x_{32} \vee \neg x_{31}) = c_{23}}
 \end{array}$$

The clause  $c_{23}$  is implied by  $c_{24}$ , so we omit it. Resulting formula:

$$\begin{aligned}
 \varphi_3 := & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{false} \vee \text{true}) \wedge c_3 : (x_{31} \vee x_{32}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge c_6 : (\neg x_{31} \vee \neg x_{32}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & c_{24} : (\neg x_{31}) \wedge c_{25} : (\neg x_{32})
 \end{aligned}$$

Next eliminate  $x_{31}$  occurring positively in  $c_3$  and negatively in  $c_6, c_{24}$ .

$$\begin{array}{c}
 \frac{c_3 : (x_{31} \vee x_{32}) \quad c_6 : (\neg x_{31} \vee \neg x_{32})}{c_{27} : (x_{32} \vee \neg x_{32}) \quad (\text{Tautology!})} \\
 \frac{c_3 : (x_{31} \vee x_{32}) \quad c_{24} : (\neg x_{31})}{c_{28} : (x_{32})}
 \end{array}$$

Resulting formula:

$$\begin{aligned}
 \varphi_3 := & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{false} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge \text{false} \vee (\text{true} \vee \text{true}) \wedge \\
 & \text{false} \vee (\text{true} \vee \text{true}) \wedge c_{25} : (\neg x_{32}) \wedge \\
 & c_{28} : (x_{32})
 \end{aligned}$$

At last eliminate  $x_{32}$  occurring positively in  $c_{28}$  and negatively in  $c_{25}$ .

$$\frac{c_{28} : (x_{32}) \quad c_{25} : (\neg x_{32})}{\square}$$

Since we derived the **empty clause**, the formula is **unsatisfiable**.

- d) We can trace back the resolution from the empty clause up to the original clauses. This yields the unsatisfiable core  $\{c_1, c_2, c_3, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}\}$ . Only the clauses encoding that one pigeon is not in two holes are not needed. Note that we may obtain other unsatisfiable cores by different resolution strategies, but all of them will be large.
- e) i. The **decision tree** of the pigeon hole problem must be **fully expanded** until the unsatisfiability is proven. In most **common problems** there are **subtrees** which have **not to be computed** again.
- ii. The **unsatisfiable core** is **very large** so that the resolution always uses nearly the whole input formula to derive a conflict.

## Exercise 2

- a) Transfer the formula you created in Exercise 1 b) into the standard SAT input format (DIMACS<sup>1</sup>). Also store your result as a text file and check it for satisfiability by using MiniSat<sup>2</sup>
- b) Download additional pigeon whole problems for  $n = 6, 7, \dots$  from the Moodle room. They are already in the DIMACS format. Use MiniSat to check for satisfiability. Note the running times of each computation in a table. What do you think is the largest  $n$  whose corresponding formula can be solved within one hour? Give a reason!
- c) Download the modified versions of the pigeon hole problems having  $n$  holes and  $n + 2$  pigeons. How are the running times compared to the original problems?

*Solution:*

- a) The formula for  $n = 2$  in DIMACS can be written as follows:

```
p cnf 6 12
1 2 0
3 4 0
5 6 0
-1 -2 0
-3 -4 0
-5 -6 0
-1 -3 0
-1 -5 0
-3 -5 0
-2 -4 0
-2 -6 0
-4 -6 0
```

The output of MiniSat 2.0 on the above input is:

```
===== [ Problem Statistics ] =====
|
| Number of variables: 6
| Number of clauses: 9
| Parsing time: 0.00 s
|
===== [ Search Statistics ] =====
| Conflicts | ORIGINAL | LEARNT | Progress |
| Vars | Clauses | Literals | Limit | Clauses | Lit/Cl |
|-----|-----|-----|-----|-----|-----|
```

<sup>1</sup>See for example <http://www.satcompetition.org/2009/format-benchmarks2009.html>

<sup>2</sup>Website: <http://minisat.se/MiniSat.html>. Install it on *Debian/Ubuntu* via `sudo apt install minisat`. Verify the result and give the running time of the computation; you may use the *Windows Subsystem for Linux* to install Ubuntu on Windows.

```

=====
|          0 |          6          9          18 |          3          0  -nan |  0.000 % |
=====
restarts           : 1
conflicts          : 2              (inf /sec)
decisions          : 1              (0.00 % random) (inf /sec)
propagations       : 10             (inf /sec)
conflict literals  : 1              (0.00 % deleted)
Memory used        : 2.07 MB
CPU time           : 0 s

```

UNSATISFIABLE

b) Running times using a slow desktop computer.

<i>Holes</i>	<i>Pigeons</i>	<i>Clauses</i>	<i>CPU time (sec)</i>	<i>Pigeons</i>	<i>Clauses</i>	<i>CPU time (sec)</i>
6	7	133	0.002999	8		
7	8	204	0.033994	9		
8	9	297	0.167974	10	370	0.174973
9	10	415	1.22681	11	506	2.10068
10	11	561	12.7131	12	672	27.3078
11	12	738	356.391	13	871	1142.8
12	13	949	9819.13	14	1106	39587.6

c) For more than seven holes the modified problem seems to be harder to solve, particularly MiniSat is about two times slower in these cases. One reason could be the significantly higher number of clauses in the modified case for large numbers of holes.