# Modelling with SAT, a Tutorial

Mohamed Siala
LAAS-CNRS & INSA-Toulouse
siala.github.io

January 18, 2023

The purpose of this tutorial is to get familiar with SAT modelling. You will be guided step by step to model a real-life timetabling problem.

## 1 The DIMACS CNF Format

- We use the kissat solver available here https://github.com/siala/kissat. Follow the instructions to install it.
- A toy example is provided in instances/tp1/toy.cnf. Have a look at the file to discover the DIMACS CNF format.
- Write a simple .cnf file for the following instance:

$$x \vee \neg y \vee z$$
$$y \vee \neg w$$
$$w \vee z$$
$$x \vee \neg z \vee w$$
$$w \vee z \vee x$$

  Call the solver on the previous instance. Is it satisfiable? unsatisfiable? Try to run other examples to get used to the format.

## 2 Modelling and Solving

- Get the following modelling library: https://github.com/siala/sat-modelling-for-students
- Inspect it and run few examples
- Your task is to encode the "at least k" constraint and to test it

## 3 Modelling a General-Purpose Timetabling Problem

The problem is the following: Let $S = \{s_1, s_2, \dots s_n\}$ be a set of $n$ nurses. Every nurse $s_i \in S$ has to work some shifts in a horizon of $D = d_1, d_2, \dots d_m$ of consecutive days. We use a Boolean variable $w_{i,j}$ that is **true** iff the nurse $s_i$ is working during day $d_j$ where $i \in \{1, 2, \dots n\}$ and $j \in \{1, 2, \dots m\}$. We study different variants of this problem.

1. Model 1:
   - Every nurse has to work at least one day
   - Each day has to be covered by at least one nurse.
2. Model 2:

- Every nurse has to work exactly one day
- Each day has to be covered by exactly one nurse

3. Model 3:

   - Every nurse has to work between $a$ and $b$ days
   - The number of nurses working in each day is in the interval $[c, d]$

   Where $a, b, c, d$ are positive integers.

4. Model 4:

   - Every nurse $n_i$ has to work between $a_i$ and $b_i$ days
   - The number of nurses working in each day is in the interval $[c_j, d_j]$

   Where $a_i, b_i, c_j, d_j$ are positive integers.

For each model:

- On a paper, write the corresponding SAT model
- Write a program that takes as input the required parameters and generates a correspondent .cnf file
- Run the solver on the generated cnf files
- Print some solutions (if they exist) visually

Once you reach this point, you can start evaluating the scalability of the different models.