About

Modeling problems with propositional and first-order logic, and searching for satisfiable solutions.

Understanding logic is useful for understanding how to formulate constraints for optimization problems.

Satisfiability (SAT)

Model problem with propositional logic.

Satisfiability aims to find assignment for variables that make the formula true.

Formulas consists of

- boolean constants:
  - false $\perp$
  - true $\top$
- boolean variables:
  - $x \in \{\perp, \top\}$
- connectives:
  - and $\wedge$
  - or $\vee$
  - not $\neg$

There are also other useful connectives such as * implication $\rightarrow$ * bi-implication $\leftrightarrow$ * xor $\otimes$

These can be written in terms of “and”, “or”, and “not” connectives.

Example: Formula

$$\phi \equiv (x_1 \wedge x_2) \vee \neg x_3$$

with variables

$$x_1, x_2, x_3 \in \{\bot, \top\}$$

has a satisfying solutions such as

$$(x_1 := \top, x_2 := \top, x_3 := \top)$$

and

$$(x_1 := \bot, x_2 := \top, x_3 := \bot).$$

In practice, formulas are reduced to a conjuctive normal form (CNF) and solved with Conflict-Driven Clause Learning (CDCL) algorithm.

Constraint Programming (CP)

Constraint programming (CP) is generalization of satisfiability.

Variables:

- $y \in N$ have values from domain $N \subseteq \mathbb{Z}$ which is a **finite** subset of integers.

Example: Constraints

- $y_1 + y_2 \leq y_3$

- $y_1 \cdot y_2 = y_3$
- `alldifferent`(y_1, y_2, y_3)
  - We call this a global constraint.
  - There are many different types of global constraints and efficient algorithms for them.
  - Logically equivalent to $(y_1 \neq y_2) \wedge (y_1 \neq y_3) \wedge (y_2 \neq y_3)$

Connection to satisfiability:

- Think about a constraints above as boolean variables.
  - $x_1 \equiv (x_1 + y_2 \leq y_3)$
  - $x_2 \equiv (y_1 \cdot y_2 = y_3)$
  - $x_3 \equiv \texttt{alldifferent}(y_1, y_2, y_3)$.
- Use the boolean variable to form a formula, such as $\phi \equiv (x_1 \wedge x_2) \vee x_3$.
- Solve the formula for satisfiability.

Example: Modeling all permutations of sequence $(1, 2, 3)$.

Variables:

$$y_1, y_2, y_3 \in \{1, 2, 3\}$$

Constraints:

$$\texttt{alldifferent}(y_1, y_2, y_3)$$

Now, all satisfying assignments for variables (y_1, y_2, y_3) form a permutation of sequence $(1, 2, 3)$.

Combinatorial Optimization (COP)

We can form a combinatorial optimization problem (COP) by adding an objective function to constrain program (CP).

Example: By expanding the previous example with distance function $d : N \times N \to \mathbb{N}$, we can write the Traveling Salesman Problem (TSP) as follows:

Variables:

$$y_1, y_2, ..., y_n \in N = \{1, 2, ..., n\}$$

Constraints:

$$\texttt{alldifferent}(y_1, y_2, ..., y_n)$$

Objective:

$$\min \left(\sum_{i=1}^{n-1} d(y_i, y_{i+1}) + d(y_n, y_1) \right)$$

We can use the **MiniZinc** modeling language for modeling.

The **Google OR-Tools** solver is effective for solving combinatorial optimization problems.