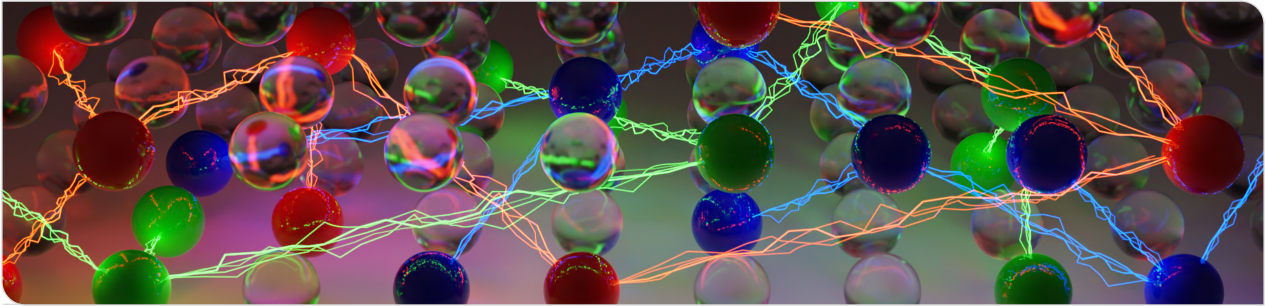
Karlsruher Institut für Technologie

Practical SAT Solving

Lecture 2

Markus Iser, Dominik Schreiber, Tomáš Balyo | April 22, 2024



Overview

Recap. Lecture 1

- Satisfiability: Propositional Logic, CNF Formulas, NP-completeness, Applications
- Examples: Pythagorean Triples, Arithmetic Progressions, k-Colorability
- Incremental SAT: IPASIR, Sample Code

Today's Topics

- Tractable Subclasses
- Constraint Encodings
- Encoding Techniques

Tractable Subclasses

Tractable Subclasses (cf. Schaefer, 1978)

- **2-SAT**
  Exactly two literals per clause
- **HORN-SAT**
  At most one positive literal per clause
- **Inverted HORN-SAT**
  At most one negative literal per clause
- **Positive** / **Negative**
  Literals occur only pure (either positive or negative)
- **XOR-SAT**
  No clauses, only XOR constraints

2-SAT

Each clause has exactly two literals.

Example (2-SAT Formulas)

$F_5 = \{\{x_1, x_2\}, \{\overline{x_1}, x_2\}, \{x_1, \overline{x_2}\}, \{\overline{x_1}, \overline{x_2}\}\}$

$F_7 = \{\{\overline{x_1}, x_2\}, \{\overline{x_2}, x_3\}, \{\overline{x_3}, x_1\}, \{x_2, x_4\}, \{x_3, x_4\}, \{x_1, x_3\}\}$

Linear Time Algorithm for 2-SAT (cf. Aspvall et al., 1979)

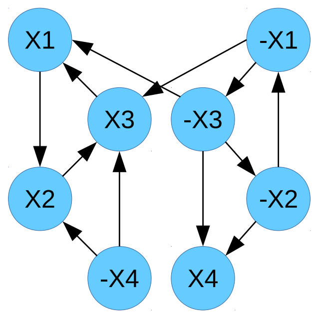
- Construct Implication Graph
- Find Strongly Connected Components (SCC) with Tarjan's Algorithm
  Complexity: $\mathcal{O}(n + m)$, where m is the number of clauses
- Check for Complementary Literals in the same SCC

Implication Graph

An **implication graph** of a 2-SAT formula F is a directed graph with a vertex for each literal of F and 2 edges for each clause $(l_1 \lor l_2)$: $\bar{l}_1 \rightarrow l_2$ and $\bar{l}_2 \rightarrow l_1$.

Example (Implication Graph)

$F_7 = \{\{\overline{x_1}, x_2\}, \{\overline{x_2}, x_3\}, \{\overline{x_3}, x_1\}, \{x_2, x_4\}, \{x_3, x_4\}, \{x_1, x_3\}\}$



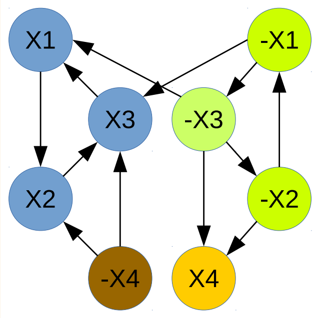
Implication Graph

An **implication graph** of a 2-SAT formula F is a directed graph with a vertex for each literal of F and 2 edges for each clause $(l_1 \vee l_2)$: $\bar{l}_1 \rightarrow l_2$ and $\bar{l}_2 \rightarrow l_1$.

Example (Implication Graph)

$F_7 = \{\{\overline{x_1}, x_2\}, \{\overline{x_2}, x_3\}, \{\overline{x_3}, x_1\}, \{x_2, x_4\}, \{x_3, x_4\}, \{x_1, x_3\}\}$

- Find Strongly Connected Components (SCC)
- SCC: There is a path from every vertex to every other vertex
- Tarjan's algorithm finds SCCs in $\mathcal{O}(|V| + |E|)$



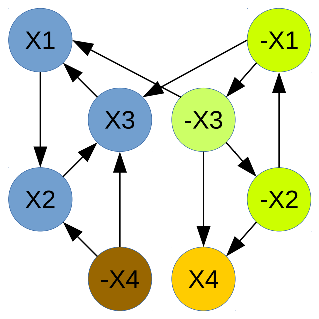
Implication Graph

An **implication graph** of a 2-SAT formula F is a directed graph with a vertex for each literal of F and 2 edges for each clause $(l_1 \vee l_2)$: $\bar{l}_1 \rightarrow l_2$ and $\bar{l}_2 \rightarrow l_1$.

Example (Implication Graph)

$F_7 = \{\{\overline{x_1}, x_2\}, \{\overline{x_2}, x_3\}, \{\overline{x_3}, x_1\}, \{x_2, x_4\}, \{x_3, x_4\}, \{x_1, x_3\}\}$

- If an SCC contains both x and $\overline{x}$, the formula is UNSAT
  - x implies its own negation!
  - Literals in an SCC must be either all true or all false



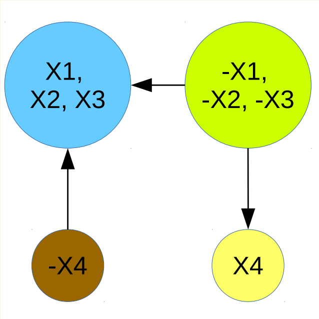
Implication Graph

An **implication graph** of a 2-SAT formula F is a directed graph with a vertex for each literal of F and 2 edges for each clause $(l_1 \vee l_2)$: $\overline{l}_1 \rightarrow l_2$ and $\overline{l}_2 \rightarrow l_1$.

Example (Implication Graph)

$F_7 = \{\{\overline{x_1}, x_2\}, \{\overline{x_2}, x_3\}, \{\overline{x_3}, x_1\}, \{x_2, x_4\}, \{x_3, x_4\}, \{x_1, x_3\}\}$

- If an SCC contains both x and $\overline{x}$, the formula is UNSAT
  - x implies its own negation!
  - Literals in an SCC must be either all true or all false
- What about SAT? How to get a solution?
  - Contract each SCC into one vertex
  - In reverse topological order, set unassigned literals to `true`.



HornSAT

Each clause contains at most one positive literal.

Example (Horn Formula)

Each clause can be written as an implication with positive literals only and a single consequent:

$$F_6 = \{\{\overline{x_1}, x_2\}, \{\overline{x_1}, \overline{x_2}, x_3\}, \{x_1\}\}$$
$$\equiv (x_1 \rightarrow x_2) \wedge ((x_1 \wedge x_2) \rightarrow x_3) \wedge (\top \rightarrow x_1)$$

HornSAT

Each clause contains at most one positive literal.

Example (Horn Formula)

Each clause can be written as an implication with positive literals only and a single consequent:

$$F_6 = \{\{\overline{x_1}, x_2\}, \{\overline{x_1}, \overline{x_2}, x_3\}, \{x_1\}\}$$
$$\equiv (x_1 \rightarrow x_2) \wedge ((x_1 \wedge x_2) \rightarrow x_3) \wedge (\top \rightarrow x_1)$$

Solving Horn Formulas

- Propagate until fixpoint
- If $\top \rightarrow \bot$ then the formula is UNSAT. Otherwise it is SAT.
- Construct a satisfying assignment by setting the remaining variables to `false`

Hidden / Renamable / Disguised Horn

A CNF formula is Hidden Horn if it can be made Horn by flipping the polarity of some of its variables.

Example (Hidden Horn Formula)

$$F_8 = \{\{x_1, x_2, x_4\}, \{x_2, \overline{x_4}\}, \{x_1\}\}$$
$$\rightsquigarrow \{\{\overline{x_1}, \overline{x_2}, x_4\}, \{\overline{x_2}, \overline{x_4}\}, \{\overline{x_1}\}\}$$

How to recognize a Hidden Horn formula? And how to hard is it?

Hidden / Renamable / Disguised Horn

A CNF formula is Hidden Horn if it can be made Horn by flipping the polarity of some of its variables.

Example (Hidden Horn Formula)

$$F_8 = \{\{x_1, x_2, x_4\}, \{x_2, \overline{x_4}\}, \{x_1\}\}$$
$$\rightsquigarrow \{\{\overline{x_1}, \overline{x_2}, x_4\}, \{\overline{x_2}, \overline{x_4}\}, \{\overline{x_1}\}\}$$

How to recognize a Hidden Horn formula? And how to hard is it?

Recognizing Hidden Horn Formula F

Construct 2-SAT formula R_F that contains the clause $\{l_1, l_2\}$ iff there is a clause $C \in F$ such that $\{l_1, l_2\} \subseteq C$.

- Example: $R_{F_8} = \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_4\}, \{x_2, \overline{x_4}\}\}$
- If the 2-SAT formula is satisfiable, then F is Hidden Horn
- If $x_i = \texttt{true}$ in ϕ, then x_i needs to be renamed to $\overline{x}_i$

Mixed Horn

A CNF formula is Mixed Horn if it contains only binary and Horn clauses.

Example (Mixed Horn Formula)

$F_9 = \{\{\overline{x_1}, \overline{x_7}, x_3\}, \{\overline{x_2}, \overline{x_4}\}, \{x_1, x_5\}, \{x_3\}\}$

How to solve a Mixed Horn formula? And how to hard is it?

Mixed Horn

A CNF formula is Mixed Horn if it contains only binary and Horn clauses.

Example (Mixed Horn Formula)

$F_9 = \{\{\overline{x_1}, \overline{x_7}, x_3\}, \{\overline{x_2}, \overline{x_4}\}, \{x_1, x_5\}, \{x_3\}\}$

How to solve a Mixed Horn formula? And how to hard is it?

Mixed Horn is NP-complete

Proof: Reduce SAT to Mixed Horn SAT

- For each non-Horn non-quadratic clause $C = (l_1 \vee l_2 \vee l_3 \vee \dots)$
  - for each but one positive $l_i \in C$ introduce a new variable l_i'
  - replace l_i in C by $\overline{l_i'}$
  - add $(l_i' \vee l_i) \wedge (\overline{l_i'} \vee \overline{l_i})$ to establish $l_i = \overline{l_i'}$