SAT Solving

Steffen Hölldobler, Norbert Manthey
International Center for Computational Logic
Technische Universität Dresden
Germany

- Introduction
- A Generic SAT Solver
- A Real SAT Solver
- Competition

"Logic is everywhere ..."
Atomic sentences (atomic formulas or atoms)

The sun is shining
The vertex with number 1 is green
LH1015 is flying from Dresden to Frankfurt
At the 4th position of the DNA-strand is the nucleotide C

Abbreviations for atoms

\[ p_1 \quad \text{The sun is shining} \]
\[ p_2 \quad \ldots \]

Truth values

true (\( \top \)) and false (\( \bot \))

Interpretation

Mapping from the set of formulas to \( \{ \top, \bot \} \)

\[ p_1 \leftrightarrow \top \quad \text{The sentence “the sun is shining” is true} \]
\[ p_2 \leftrightarrow \bot \quad \text{The sentence “the vertex with number 1 is green” is false} \]
Complex Sentences

▶ Complex sentences or formulas

The sun is shining \( p_1 \)
The sun is not shining \( \neg p_1 \)
The sun is shining or it is not shining \( (p_1 \lor \neg p_1) \)
The sun is shining and the vertex with number 1 is green \( (p_1 \land p_2) \)
If the sun is shining, then the vertex with number 1 is green \( (p_1 \rightarrow p_2) \)

▶ Definition The set of (propositional) formulas is the smallest set satisfying the following conditions

▷ All atoms are formulas
▷ If \( F \) is a formula, then so is \( \neg F \)
▷ If \( F \) and \( G \) are formulas, then so are \( (F \lor G) \), \( (F \land G) \) and \( (F \rightarrow G) \)

▶ Example \( \neg(\neg(p_1 \land p_2) \rightarrow (p_1 \lor \neg p_1)) \)

▶ Convention We remove “\( p \)” and denote atoms by natural numbers
Interpretations and Models

► Remember
Interpretations are mappings from the set of formulas to \{\top, \bot\}

► How can we interpret complex formulas?

<table>
<thead>
<tr>
<th>F</th>
<th>\neg F</th>
<th>F</th>
<th>G</th>
<th>(F \lor G)</th>
<th>(F \land G)</th>
<th>(F \rightarrow G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\top</td>
<td>\bot</td>
<td>\top</td>
<td>\top</td>
<td>\top</td>
<td>\top</td>
<td>\top</td>
</tr>
<tr>
<td>\bot</td>
<td>\top</td>
<td>\bot</td>
<td>\top</td>
<td>\top</td>
<td>\bot</td>
<td>\top</td>
</tr>
<tr>
<td>\top</td>
<td>\bot</td>
<td>\top</td>
<td>\bot</td>
<td>\bot</td>
<td>\bot</td>
<td>\bot</td>
</tr>
<tr>
<td>\bot</td>
<td>\bot</td>
<td>\bot</td>
<td>\bot</td>
<td>\bot</td>
<td>\bot</td>
<td>\bot</td>
</tr>
</tbody>
</table>

► Example
Let \( I \) be an interpretation such that \( I(1) = \top \) and \( I(2) = \bot \), then

\[ I(\neg(\neg(1 \land 2) \rightarrow (1 \lor \neg 1))) = \bot \]

► Definition
An interpretation mapping a formula \( F \) onto \( \top \) is called model for \( F \)
Satisfiability Problems

► Definition  Formula $F$ is **satisfiable** if there exists a model for $F$

► Definition  A **satisfiability problem** consists of a formula $F$ and is the question whether $F$ is satisfiable

► Example  Is $F = \neg(\neg(1 \land 2) \rightarrow (1 \lor \neg 1))$ satisfiable?  No

<table>
<thead>
<tr>
<th>interpretation</th>
<th>formula $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\top$ $\top$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\bot$ $\top$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\top$ $\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\bot$ $\bot$</td>
<td>$\bot$</td>
</tr>
</tbody>
</table>

► Observation  If a formula contains $n$ different atoms, then there are $2^n$ different interpretations and we may need to consider them all

► Remark  Complexity theory was developed with the help of the satisfiability problem!
Applications

- Termination of programs
- Planning and configuration
- Biocomputing
- Verification of hard- and software
- Scheduling
- Cryptoanalysis
- Answer set programming
Graph Coloring Problem

► Given a finite graph and a finite set of colors
► Goal a mapping from vertices to colors such that neighboring vertices have different colors

► Formally
  ▶ Each vertex has at least one color
  ▶ Each vertex has at most one color
  ▶ Neighboring vertices have different colors
Naive Encoding of the Graph Coloring Problem (1)

► **Task** For a given graph coloring problem $G$ find a propositional formula $F$ such that each model for $F$ represents a solution for $G$

► **Example** Consider the set $\{1, 2, 3, 4\}$ of colors and the graph

![Graph Diagram]

► **Naive encoding of the assignment of colors to vertices**

11 vertex 1 is mapped to color 1
12 vertex 1 is mapped to color 2
: :
44 vertex 4 is mapped to color 4
Naive Encoding of the Graph Coloring Problem (2)

► Each vertex has at least one color

\[(11 \lor 12 \lor 13 \lor 14) \land \ldots \land (41 \lor 42 \lor 43 \lor 44)\]

► Each vertex has at most one color

\[\neg(11 \land 12) \land \neg(11 \land 13) \land \neg(11 \land 14) \ldots \land \neg(43 \land 44)\]

► Neighboring vertices have different colors

\[\neg(11 \land 21) \land \neg(12 \land 22) \land \neg(13 \land 23) \land \ldots \land \neg(34 \land 44)\]

► Claim  Let $F$ be the conjunction of the above mentioned formulas

Each model for $F$ encodes the solution of the given graph coloring problem

► Example  The interpretation mapping 11, 22, 33 und 44 onto $\top$ and all other atoms onto $\bot$ is a model for $F$
Some Observations

- The formulas will be large
- Applications may lead to $10^7$ different atoms and $10^8$ conjunctions
- The search space may be $2^{10^7}$!
- The formulas are difficult to understand
- We need help
- SAT solvers
An Advanced Encoding of the Graph Coloring Problem (1)

► Idea  We specify an ordering among the colors  $1 < 2 < 3 < 4$

► Encoding of the colors assigned to vertices

- $1_1$ vertex 1 has a color which is greater or equal 1
- $1_2$ vertex 1 has a color which is greater or equal 2
- $\vdots$
- $4_4$ vertex 4 has a color which is greater or equal 4

- $1_1 \land \neg 1_2$ vertex 1 has color 1
- $1_2 \land \neg 1_3$ vertex 1 has color 2
- $1_3 \land \neg 1_4$ vertex 1 has color 3
- $1_4$ vertex 1 has color 4
- $\vdots$
- $4_4$ vertex 4 has color 4
An Advanced Encoding of the Graph Coloring Problem (2)

- Each vertex has at least one color

\[ 11 \land 21 \land 31 \land 41 \]

- Each vertex has at most one color

\[ (\neg 11 \rightarrow \neg 12) \land (\neg 12 \rightarrow \neg 13) \land (\neg 13 \rightarrow \neg 14) \land \ldots \land (\neg 43 \rightarrow \neg 44) \]

- Neighboring vertices have different colors

\[ \neg (11 \land \neg 12 \land 21 \land \neg 22) \land \ldots \land \neg (34 \land 44) \]

- Claim Let \( F \) be the conjunction of the above mentioned formulas
  Each model for \( F \) encodes a solution of the given graph coloring problem

- Observation
  Using the advanced encoding reduced the runtime by 10% on average
Conjunctive Normal Form (1)

► Usually SAT-solvers accept formulas only in a particular form

► Definition A literal is an atom or its negation

► Examples 11, ¬12, ¬13, 14

► Definition A formula is in conjunctive normal form (CNF), if it is of the form

\[(L_{11} \vee \ldots \vee L_{1n_1}) \land \ldots \land (L_{m1} \vee \ldots \vee L_{mn_m})\]

where the \(L_{ij}\) are literals

► Definition Two formulas \(F\) and \(G\) are equivalent if \(I(F) = I(G)\) holds for all interpretations \(I\)

► Example \(\neg(11 \land 12)\) and \((\neg11 \lor \neg12)\) are equivalent
Conjunctive Normal Form (2)

- **Theorem**  For each formula exists an equivalent formula in CNF

- **Example**

  \[(\neg 11 \lor \neg 12) \land (\neg 11 \lor \neg 13) \land (\neg 11 \lor \neg 14) \ldots \land (\neg 43 \lor \neg 44)\]

  is an equivalent CNF of

  \[\neg(11 \land 12) \land \neg(11 \land 13) \land \neg(11 \land 14) \ldots \land \neg(43 \land 44)\]

- **Observation**

  There are algorithms which transform a given formula into CNF
SAT Solvers

- Several SAT solvers are freely available
  - MiniSAT
  - RSAT
  - riss
- They solve problems in CNF with up to $10^7$ atoms and $10^8$ conjunctions
- They are applied in industry
- There are yearly international competitions
- SAT solvers are continuously improved
- These improvements concern all aspects of computer science
A Solution of the Graph Coloring Problem with Naive Encoding

► Input to a SAT solver

```
p cnf nv nc
11 12 13 14 0
.
41 42 43 44 0
-11 -12 0
.
-43 -44 0
-11 -21 0
.
-34 -44 0
```

where \(nv\) and \(nc\) are the number of atoms (or variables) and conjunctions (or clauses), respectively

► Possible output of the SAT solver

```
11 22 33 44
```
A Hamiltonian cycle is a cyclic path through a graph which contains each vertex exactly once (except for the vertex that is both the start and the end, which is visited twice).

The Hamiltonian cycle problem consists of a graph and is the question whether there exists a Hamiltonian cycle in the given graph.

- It is a fundamental problem in graph theory.
- It is NP-complete.
Exercise

► Write a program which
  ▶ successively reads Hamiltonian cycle problems from a file,
  ▶ transforms the question whether the problem has a solution into a SAT problem,
  ▶ and, if the SAT problem is solvable, writes a cycle to stdout and terminates with exit code 10
  ▶ and, if the SAT problem is unsolvable, terminates with exit code 20

► You may use existing algorithms for transforming a formula into CNF

► You may use any available SAT solver

► The correctness of your solutions is automatically checked
Competition

- A sequence of Hamiltonian cycle problems has to be considered in a certain period of time.
- The measured time is the time from reading the problem until writing the solution (a particular cycle or ’unsolvable’).
- For each team $t$ of at most 3 participants we compute:
  - $C(t)$: the number of problems which were correctly solved
  - $T(t)$: the time needed to solve these problems
  - $F(t)$: the number of submitted solutions which were wrong
- The ranking of the teams is determined by the following relation:
  $$s \succ t \iff \begin{array}{l}
  F(s) < F(t) \\
  \lor (F(s) = F(t) \land C(s) > C(t)) \\
  \lor (F(s) = F(t) \land C(s) = C(t) \land T(s) < T(t))
  \end{array}$$
- The competition will take place on Sep 25.
Further Information

➤ For riss see
   http://tools.computational-logic.org/content/riss.php

➤ For Hamiltonian cycle problems see
   https://ddll.inf.tu-dresden.de/w/images/8/83/
   Forschungslinie2015-Graphs.tar.gz

➤ For a description of the representation of graphs see
   http://mat.gsia.cmu.edu/COLOR/general/ccformat.ps