Introduction

Marijn J.H. Heule
Warren A. Hunt Jr.

The University of Texas at Austin
From 100 variables, 200 clauses (early 90’s) to 1,000,000 vars. and 5,000,000 clauses in 15 years.
Motivation satisfiability solving

From 100 variables, 200 clauses (early 90’s) to 1,000,000 vars. and 5,000,000 clauses in 15 years.

Applications:
Hardware and Software Verification, Planning, Scheduling, Optimal Control, Protocol Design, Routing, Combinatorial problems, Equivalence Checking, etc.
Motivation satisfiability solving

From 100 variables, 200 clauses (early 90’s) to 1,000,000 vars. and 5,000,000 clauses in 15 years.

Applications:

Hardware and Software Verification, Planning, Scheduling, Optimal Control, Protocol Design, Routing, Combinatorial problems, Equivalence Checking, etc.

SAT used to solve many other problems!
Overview

- Introduction
- The Satisfiability problem
- Terminology
- SAT solving
- SAT benchmarks
”You are chief of protocol for the embassy ball. The crown prince instructs you either to invite *Peru* or to exclude *Qatar*. The queen asks you to invite either *Qatar* or *Romania* or both. The king, in a spiteful mood, wants to snub either *Romania* or *Peru* or both. Is there a guest list that will satisfy the whims of the entire royal family?”
"You are chief of protocol for the embassy ball. The crown prince instructs you either to invite *Peru* or to exclude *Qatar*. The queen asks you to invite either *Qatar* or *Romania* or both. The king, in a spiteful mood, wants to snub either *Romania* or *Peru* or both. Is there a guest list that will satisfy the whims of the entire royal family?"

\[ (P \lor \neg Q) \land (Q \lor R) \land (\neg R \lor \neg P) \]
\[ F := (P \lor \neg Q) \land (Q \lor R) \land (\neg R \lor \neg P) \]

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$R$</th>
<th>falsifies</th>
<th>$\varphi \circ F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$(Q \lor R)$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>—</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$(P \lor \neg Q)$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$(P \lor \neg Q)$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$(Q \lor R)$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$(\neg R \lor \neg P)$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>—</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$(\neg R \lor \neg P)$</td>
<td>0</td>
</tr>
</tbody>
</table>
What are the solutions for the following formula?

\[ (A \lor B \lor \neg C) \]
\[ (\neg A \lor \neg B \lor C) \]
\[ (B \lor C \lor \neg D) \]
\[ (\neg B \lor \neg C \lor D) \]
\[ (A \lor C \lor D) \]
\[ (\neg A \lor \neg C \lor \neg D) \]
\[ (\neg A \lor B \lor D) \]
What are the solutions for the following formula?

\[(A \lor B \lor \neg C)\]
\[\neg A \lor \neg B \lor C\]
\[(B \lor C \lor \neg D)\]
\[\neg B \lor \neg C \lor D\]
\[(A \lor C \lor D)\]
\[\neg A \lor \neg C \lor \neg D\]
\[\neg A \lor B \lor D\]
Given a **CNF formula**, does there exist an **assignment** to the **Boolean variables** that satisfies all **clauses**?
Terminology: Variables and literals

Boolean variable
- can be assigned the Boolean values 0 or 1

Literal
- refers either to $x_i$ or its complement $\neg x_i$
- literals $x_i$ are satisfied if variable $x_i$ is assigned to 1 (true)
- literals $\neg x_i$ are satisfied if variable $x_i$ is assigned to 0 (false)
Terminology: Clauses

Clause

- Disjunction of literals: E.g. $C_j = (l_1 \lor l_2 \lor l_3)$
- Can be falsified with only one assignment to its literals: All literals assigned to false
- Can be satisfied with $2^k - 1$ assignment to its $k$ literals
- One special clause - the empty clause (denoted by $\emptyset$) - which is always falsified
Formula

- Conjunction of clauses: E.g. $\mathcal{F} = C_1 \land C_2 \land C_3$
- Is *satisfiable* if there exists an assignment satisfying all clauses, otherwise *unsatisfiable*
- Formulae are defined in *Conjunction Normal Form* (CNF) and generally also stored as such - also learned information
Assignment

- Mapping of the values 0 and 1 to the variables
- $\varphi \circ F$ results in a reduced formula $F_{\text{reduced}}$:
  - all satisfied clauses are removed
  - all falsified literals are removed
- satisfying assignment $\leftrightarrow F_{\text{reduced}}$ is empty
- falsifying assignment $\leftrightarrow F_{\text{reduced}}$ contains $\emptyset$
- partial assignment versus full assignment
Given two clauses $C_1 = (x \lor a_1 \lor \cdots \lor a_n)$ and $C_2 = (\neg x \lor b_1 \lor \cdots \lor b_m)$, the resolvent of $C_1$ and $C_2$ (denoted by $C_1 \boxtimes C_2$) is $R = (a_1 \lor \cdots \lor a_n \lor b_1 \lor \cdots \lor b_m)$.
Given two clauses $C_1 = (x \lor a_1 \lor \cdots \lor a_n)$ and $C_2 = (\neg x \lor b_1 \lor \cdots \lor b_m)$, the resolvent of $C_1$ and $C_2$ (denoted by $C_1 \blacktriangleright C_2$) is $R = (a_1 \lor \cdots \lor a_n \lor b_1 \lor \cdots \lor b_m)$.

Examples for $F := (P \lor \neg Q) \land (Q \lor R) \land (\neg R \lor \neg P)$:

- $(P \lor \neg Q) \blacktriangleright (Q \lor R) = (P \lor R)$
- $(P \lor \neg Q) \blacktriangleright (\neg R \lor \neg P) = (\neg Q \lor \neg R)$
- $(Q \lor R) \blacktriangleright (\neg R \lor \neg P) = (Q \lor \neg P)$
Resolution

Given two clauses \( C_1 = (x \lor a_1 \lor \cdots \lor a_n) \) and 
\( C_2 = (\neg x \lor b_1 \lor \cdots \lor b_m) \), the resolvent of \( C_1 \) and \( C_2 \) (denoted by \( C_1 \bowtie C_2 \)) is 
\( R = (a_1 \lor \cdots \lor a_n \lor b_1 \lor \cdots \lor b_m) \)

Examples for \( F : = (P \lor \neg Q) \land (Q \lor R) \land (\neg R \lor \neg P) \)

- \( (P \lor \neg Q) \bowtie (Q \lor R) = (P \lor R) \)
- \( (P \lor \neg Q) \bowtie (\neg R \lor \neg P) = (\neg Q \lor \neg R) \)
- \( (Q \lor R) \bowtie (\neg R \lor \neg P) = (Q \lor \neg P) \)

Resolution, i.e., adding resolvents until fixpoint, is a complete proof procedure. It produces the empty clause if and only if the formula is unsatisfiable
A clause $C$ is a tautology if it contains for some variable $x$, both the literals $x$ and $\neg x$.

**Slightly Harder Example 2**

Compute all non-tautological resolvents for:

$$(A \lor B \lor \neg C) \land (\neg A \lor \neg B \lor C) \land (B \lor C \lor \neg D) \land (\neg B \lor \neg C \lor D) \land (A \lor C \lor D) \land (\neg A \lor \neg C \lor \neg D) \land (\neg A \lor B \lor D)$$

Which resolvents remain after removing the supersets?
A *unit clause* is a clause of size 1

**UnitPropagation** \((\varphi, \mathcal{F})\):

1. **while** \(\emptyset \notin \mathcal{F}\) **and** unit clause \(y\) exists **do**
2. expand \(\varphi\) and simplify \(\mathcal{F}\)
3. **end while**
4. **return** \(\varphi, \mathcal{F}\)
\[ F_{\text{unit}} := (\neg x_1 \lor \neg x_3 \lor x_4) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\neg x_1 \lor x_4 \lor \neg x_5) \land (x_1 \lor \neg x_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \neg x_6) \]
\( \mathcal{F}_{\text{unit}} := (\neg x_1 \lor \neg x_3 \lor x_4) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\neg x_1 \lor x_4 \lor \neg x_5) \land (x_1 \lor \neg x_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \neg x_6) \)

\( \varphi = \{ x_1 = 1 \} \)
Unit propagation: Example

\( \mathcal{F}_{\text{unit}} := (\neg x_1 \lor \neg x_3 \lor x_4) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\neg x_1 \lor x_4 \lor \neg x_5) \land (x_1 \lor \neg x_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \neg x_6) \)

\( \varphi = \{x_1 = 1, x_2 = 1\} \)
Unit propagation: Example

\[ \mathcal{F}_{\text{unit}} := (\neg x_1 \lor \neg x_3 \lor x_4) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\neg x_1 \lor x_4 \lor \neg x_5) \land (x_1 \lor \neg x_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \neg x_6) \]

\[ \varphi = \{x_1=1, x_2=1, x_3=1\} \]
Unit propagation: Example

\[ F_{\text{unit}} := (\neg x_1 \lor \neg x_3 \lor x_4) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\neg x_1 \lor x_4 \lor \neg x_5) \land (x_1 \lor \neg x_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \neg x_6) \]

\[ \varphi = \{ x_1=1, x_2=1, x_3=1, x_4=1 \} \]
Davis Putnam Logemann Loveland [DP60,DLL62]

- Simplify (Unit Propagation)
- Split the formula
  - Variable Selection Heuristics
  - Direction heuristics
\[ F_{\text{DPLL}} := (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_3) \land (
\neg x_1 \lor \neg x_2 \lor x_3) \land (x_1 \lor x_3) \land (\neg x_1 \lor \neg x_3) \]
$\mathcal{F}_{\text{DPLL}} := (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_3) \land \\
(\neg x_1 \lor \neg x_2 \lor x_3) \land (x_1 \lor x_3) \land (\neg x_1 \lor \neg x_3)$
$\mathcal{F}_{DPLL} := (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (x_1 \lor x_3) \land (\neg x_1 \lor \neg x_3)$
DPLL: Slightly Harder Example

Slightly Harder Example 3

Construct a DPLL tree for:

\[(A \lor B \lor \neg C) \land (\neg A \lor \neg B \lor C) \land
(B \lor C \lor \neg D) \land (\neg B \lor \neg C \lor D) \land
(A \lor C \lor D) \land (\neg A \lor \neg C \lor \neg D) \land
(\neg A \lor B \lor D)\]
Decision variables
- Selected by the heuristics
- Play a crucial role in performance

Implied variables
- Assigned by reasoning (e.g. unit propagation)
- Maximizing the number of implied variables is an important aspect of look-ahead SAT solvers
A clause $C$ represents a set of falsified assignments, i.e. those assignments that falsify all literals in $C$.

A falsifying assignment $\varphi$ for a given formula represents a set of clauses that follow from the formula.

- For instance with all decision variables.
- Important feature of conflict-driven SAT solvers.
Conflict-driven
- "brute-force", complete
- examples: zchaff, minisat, rsat

Look-ahead
- lots of reasoning, complete
- examples: march, OKsolver, kcnfs

Local search
- local optimizations, incomplete
- examples: WalkSAT, UnitWalk
Applications: Industrial

- Model Checking
  - Turing award ’07 Clarke, Emerson, and Sifakis
- Software Verification
- Hardware Verification
- Equivalence Checking Problems
Applications: Crafted

- Combinatorial problems
- Sudoku
- Factorization problems
Random $k$-SAT: Introduction

- All clauses have length $k$
- Variables have the same probability to occur
- Each literal is negated with probability of 50%
- Density is ratio Clauses to Variables
Random 3-$\text{SAT}$: % satisfiable, the phase transition

variables
- 50
- 40
- 30
- 20
- 10

clause-variable density

Heule & Hunt (UT Austin)
SAT game

by Olivier Roussel

http://www.cs.utexas.edu/~marijn/game/
Introduction

Marijn J.H. Heule
Warren A. Hunt Jr.

The University of Texas at Austin