Advanced Topics in SAT-Solving
Part II: Theoretical Aspects

Carsten Sinz

Wilhelm-Schickard-Institut for Computer Science
University of Tübingen

28.09.2004
**SAT is NP-complete**

This means (consider $\text{SAT} = \{ F \text{ in CNF} \mid F \text{ satisfiable} \}$):

1. **SAT $\in$ NP**, i.e. there is a non-deterministic Turing machine which recognizes the set of satisfiable formulae in polynomial time
   
   **Alg.:** non-deterministically build a solution vector (assignment); check whether the generated vector is a solution.

2. Every problem $P \in \text{NP}$ is polynom. reducible to SAT
   
   **Idea of Cook’s proof (1971):** show that any such $P$ can be computed by a Boolean circuit (i.e. a computer)
NP-completeness of SAT

- There is no known deterministic polynomial time algorithm to solve SAT (unless P=NP)

- An algorithm solving all SAT instances in polynomial time would imply P=NP

- Although there are SAT instances that require exponential time, subclasses of SAT may be tractable (i.e. solvable in polynomial time)

- 3-SAT is still in NP, even if all variables appear at most 3 times
Tractable Subclasses

Syntactically recognizable: (tractable: decidable in polytime)

- **2-SAT** (only Krom-clauses): each clause contains at most two literals

- **Horn-SAT** (only Horn clauses): in each clause at most one literal is positive; Horn clauses usually written as implications with negative literals on the left:

  \((\neg x_1 \lor \cdots \lor \neg x_n \lor y)\) is written as \(x_1 \land \cdots \land x_n \Rightarrow y\)

- **Trivial subclasses**: no positive (negative) clauses [positive (negative) clause: all literals positive (negative)]; ...
Complexity: Upper Bounds

Deterministic 3-SAT algorithms:

\[ 2^n \quad \text{truth table method} \]
\[ 1.618^n \quad \text{Monien, Speckenmeyer (1985)} \]
\[ 1.505^n \quad \text{Kullmann (1999)} \]
\[ 1.481^n \quad \text{Dantsin, Goerdt, Hirsch, Schöning (2000)} \]

Probabilistic 3-SAT algorithms:

\[ 1.587^n \quad \text{Paturi, Pudlák, Zane (1997)} \]
\[ 1.330^n \quad \text{Hofmeister, Schöning, Schuler, Watanabe (2001)} \]
\[ 1.329^n \quad \text{Baumer, Schuler (2002)} \]
\[ 1.324^n \quad \text{Iwama, Tamaki (2003)} \]

(from Schöning: Algorithms for the SAT Problem)
Upper Bounds: Runtime Comparison

- $2^n$
- $1.481^n$
- $1.324^n$
- $n^3$
Lower Bounds

Why are lower bounds of interest?
‘Optimality’ / limits of algorithms; P-NP problem

(Lower bounds are generally harder to obtain than upper bounds)

Results depend on algorithm: e.g. Resolution needs at least $f(n)$ steps to solve problem $X(n)$

Haken’s result (1985): Any resolution proof of PHP$_{n-1}^n$ is of length $2^{\Omega(n)}$ (for sufficiently large $n$)

(PHP$_{n-1}^n$: pigeon hole formula := $n$ pigeons cannot sit in $n - 1$ holes)
Random SAT Problems

Variable clause size model: To generate a clause \( C \), scan through all variables and add variable \( x \) with fixed probability \( p \) to the clause; negate variable with probability \( 1/2 \).
\( \implies \) Shown to be solvable in polynomial average time.

Fixed clause size model: To generate a \( k \)-clause, select \( k \) variables uniformly at random from \( V \), negate each variable with probability \( 1/2 \).
\( \implies \) Produces hard problems (dependent on clause/variable ratio.)
Comparison: Fixed vs. Variable Clause Size

(Figures on this and next slides from: Mitchell, Selman, Levesque (1992))
Random 3-SAT: Threshold Phenomena I

![Diagram showing the probability of being satisfiable as a function of the ratio of clauses to variables. The x-axis represents the ratio of clauses to variables, ranging from 2 to 8. The y-axis represents the probability, ranging from 0.0 to 1.0. The graph shows a sharp transition at a certain ratio, indicating the threshold phenomenon.]

Probability of being satisfiable

50% satisfiable point

Ratio of clauses-to-variables
Random $k$-SAT

1. Hardest problems are at phase transition point (transition from satisfiable to unsatisfiable)

2. Easy-hard-easy pattern at increasing clause/var.-ratio
   Intuition:
   At low ratios: few clauses (constraints), many satisfying assignments
   At high ratios: many constraints, inconsistencies easily detected

3. Phase transition points (Mertens, Mézard, Zecchina 2003):

$$
\begin{array}{|c|c|}
\hline
k & \text{phase transition point } (m/n) \\
\hline
3 & 4.26675 \pm 0.00015 \\
4 & 9.331 \\
5 & 21.117 \\
\hline
\end{array}
$$
\((2 + p)\)-\textbf{SAT}: Complexity

\( (2 + p)\)-\textbf{SAT}: \text{mixture of random 2- and 3-clauses}; \( p \): fraction of 3-clauses (\( p=0 \): 2-SAT, \( p=1 \): 3-SAT)

\textbf{Motivation}: \text{Where is the borderline between P and NP?}

\textbf{Phase transition phenomenon}: similar to random \( k \)-\textbf{SAT}: low clause/variable ratio \( \rightsquigarrow \) almost always satisfiable, high clause/variable ratio \( \rightsquigarrow \) almost always unsatisfiable

\textbf{Experimental result}: for \( p < \approx 0.41 \), \((2 + p)\)-\textbf{SAT} essentially behaves like 2-SAT (i.e. tractable)
(2 + p)-SAT: Phase Transition

(Figures on this and next slide from Selman’s ISAT’99 talk)
$(2 + p)$-SAT: Computational Cost