What Do Propositional Satisfiability (SAT),
Quantified Boolean Formulas (QBF),
and Computational Fluid Dynamics (CFD) Have in Common?

Allen Van Gelder
Computer Science Department
University of California
Santa Cruz, CA, USA

avg@cs.ucsc.edu
Preface — Quick Tutorial on Global Warming (presented Mar. 2008)

Why are Carbon Dioxide and Methane called “Greenhouse Gases”?
• They are transparent to 1 µ radiation, which is most of what the sun sends.
• They block a good amount of 10 µ radiation, which is what the earth mostly reradiates.
• Water vapor is transparent to both 1 µ and 10 µ. Water droplets block both.

Why are melting polar ice-caps very dangerous?
• Ice reradiates a lot of energy from the sun at the same 1 µ that is not blocked by Greenhouse Gases.
• Less ice → less 1 µ reradiation → more warming → less ice.

How will global warming destroy civilization?
• More extreme weather will break down the infrastructure of civilization.
• Recent tornadoes, droughts, floods, snows are all indications.
• If Gulf Stream breaks down, much of Russia may become uninhabitable.

What does this have to do with this workshop?
• The future of Satisfiability is in software verification.
• Reliable computer systems will be needed to manage Global Warming.
2011 Update — What About Earthquakes?

This was too “blue sky” to bring up in 2008.

The solid earth is a thin crust floating on molten earth.

What happens if you put a lot more weight on a thin crust, but not equally distributed?

When the polar ice caps melt, where does that water go?
Computational Fluid Dynamics — Vortex Cores

- Physics of Fluid Flows Is Not Well Understood
  - Navier-Stokes partial differential equations developed about 100 years ago.
  - Direct Numerical Simulation (DNS) is primary analysis tool.
- Visualization Challenges: How to Show Important Information contained within a terabyte of output data?
- Theory Challenges: Models of atmosphere and oceans that enable better predictions.

\[
\nabla (v) = \left[ \frac{\partial v}{\partial x_1}, \frac{\partial v}{\partial x_2}, \ldots, \frac{\partial v}{\partial x_m} \right]
\]

\[
v^T \ w = \ w^T \ v
\]

\[
\alpha A = A \alpha
\]

\[
(A \ B)^T = B^T \ A^T
\]

\[
A \ (B + C) = (A \ B) + (A \ C)
\]

\[
\nabla (v^T w) = v^T \nabla (w) + w^T \nabla (v)
\]

\[
\nabla (\alpha w) = \alpha \nabla (w) + w \nabla (\alpha)
\]

\[
\nabla \left( \frac{w}{\alpha} \right) = \frac{\nabla (w)}{\alpha} - \frac{w \nabla (\alpha)}{\alpha^2}
\]
Left: Supposed *vortex-core axes* are shown in *red*.

Generated *stream lines* are shown in *blue* or *green*.

Right: Stream lines generated from different seed points.
Propositional Satisfiability

Theoreticians: You can’t do it.
Practitioners: Can too!
Industry: We need this, so we’ll go with the second group.
  • Half a loaf is better than none.
  • Some problems will remain unsolved.

Propositional formula structure is rather simple

\[
\begin{align*}
&\left[ \neg x, y, \neg z, \neg w \right], \\
&\left[ \neg x, \neg y \right], \quad \uparrow \text{ literal} \\
&\left[ \neg z, w \right], \\
&\left[ z \right], \\
&\ldots
\end{align*}
\]

Propositional variable (e.g., \( w, x, y, z \)) can be true (1) or false (0).

Literal is a propositional variable or negated propositional variable.
  • Implemented as integers

Clause is a disjunction (or) of zero or more literals.

Formula is a conjunction (and) of zero or more clauses.
Solving Propositional Satisfiability Problems

Given a formula $F$, does there exist a truth-value assignment to the variables such that $F$ evaluates to true (a satisfying assignment)?

Refutation is a proof that no such assignment exists.

State of the art SAT Solving: A two-pronged approach.
- **Search** for a satisfying assignment
- **Incrementally build** a resolution refutation as search attempts fail.
- **Conflict-Driven Clause Learning (CDCL)** is the name of the technique.

Resolution derives a new clause from two known clauses by canceling one pair of conflicting literals and unioning the rest.
- Resolvent of $[x, y, z]$ and $[x, \neg y, \neg w]$ is $[x, z, \neg w]$.
- Resolvent of $[x]$ and $[\neg x]$ is $[]$. 
Clause Learning Based on Conflict Graph

Basic Data Structure in Most Modern SAT Solvers

- Introduced in current form in GRASP by Marques-Silva and Sakallah [96,99]
- Adopted into Chaff by Moskewicz, Madigan, Zhao, L. Zhang, and Malik [01]
- Variations in zChaff by L. Zhang, Madigan, Moskewicz, and Malik [01]
- Theoretical analysis by Beame, Kautz, and Sabharwal [JAIR 2004]

Conflict graphs used by Chaff, zChaff, BerkMin, Jerusat, minisat, picosat, ...
Various Cuts in Conflict Graph Yield Different Conflict Clauses to Learn

Decision Clause: \[ [p, q, \neg b] \]

1UIP Clause: \[ [p, \neg a, t] \]

Conflict Antecedent: \[ [x_1, x_2, x_3, y] \]

\[ \bot \] denotes false.

Dashed lines go to vertices at lower (earlier) “decision levels”.
Quantified Boolean Formulas

Add universal and existential quantifiers ($\forall$ and $\exists$, resp.) in propositional formulas.

- In propositional satisfiability, all variables are implicitly existential.

\[
\text{prefix}\left\{ \forall x \exists y \forall w \exists z \right\}
\begin{cases}
    [\neg x, y, \neg z, \neg w], \\
    [\neg x, \neg y], \quad \uparrow \text{literal} \\
\end{cases}
\]

\[
\text{clauses}\left\{ \\
    [\neg z, w], \\
    [z], \\
\right. \]

Harder than satisfiability.

More compact encodings are possible.

Similar CDCL techniques have recently been developed.

- $Q$-resolution replaces resolution as the work-horse inference rule.

Many aspects are not well understood.
Beyond Resolution

Future major gains in SAT solving may require getting out of the “resolution box”.

Boolean Polynomial Calculus

• Boolean Polynomial Calculus can directly simulate resolution.
• Can combine clauses where resolution is useless:
  • \((x \lor y \lor z)\) and \((x \lor \bar{y} \lor \bar{z})\) resolve to tautologous clause.
  • \((x + 1)(y + 1)(z + 1)\) and \((x + 1) \cdot y \cdot z \vdash (x + 1)(y + z)\).
• Exponentially shorter proofs than resolution on some families (?)
• Most theoretical analysis based on degree of polynomial as metric.
  • Analogous to clause-width metric for resolution.
  • But … Clause-width metric for resolution is doubtful (Bonet-Galesi 2001).
• Good heuristics will be needed to achieve practical success, in any case.
Beyond Resolution — SAT Modulo Theories (SMT)

Integrate decidable theories with propositional logic.
- Natural for software verification and other design checking.
  - Linear real arithmetic LRA
  - Linear integer arithmetic LIA
  - Difference logic DL
  - Lists, trees, arrays, others

- **WARNING:** Many decision procedures use inference rules equivalent to resolution steps: at most a constant-factor benefit.

**Research question:**
Can Boolean Polynomial Calculus be brought into the SMT fold?
- Microsoft z3 is a leading SMT tool that they claim can do this (and everything else).
Conclusions

“Theory” in computer science usually means some combination of mathematics and logic.

In Satisfiability and Scientific Visualization, theory by itself is usually not enough to do anything useful.

Ideas need to validated by “industrial strength” implementations and experiments, in most cases.

Scientific Visualization has the potential to help us better understand global climate change.

Satisfiability has the potential to help us better control global climate change (by improving software and hardware verification, mainly).