Logic of Computer Science
Lecture 7: Resolution SAT Solving

Emmanuel Kwesi Tandoh

University of Mines and Technology (UMaT)

June 2025

UNIVERSITY OF MINES
AND TECHNOLOGY
(UMaT)

Knowledge | Truth | Excellence

Motivation

Why SAT Solving Matters

Many real-world problems—from planning and hardware verification to cryptography—reduce to SAT. Efficient SAT solvers power automated reasoning across industry and research.

temsep=6pt**Hardware Design:** Check circuit equivalence.temsep=6pt**Software Testing:** Generate inputs that cover edge cases.temsep=6pt**AI Planning:** Encode planning problems as SAT for efficient solution.

Warm-Up Question

Think-Pair-Share

Why do modern SAT solvers accept only CNF formulas? List benefits of CNF.

Learning Objectives

By the end of this lecture, you should be able to:

temsep=6ptConvert propositional formulas to CNF using simple steps.temsep=6ptUse resolution to detect unsatisfiability.temsep=6ptExplain the core ideas of the DPLL algorithm.temsep=6ptAppreciate how SAT solvers automate search.

Converting to CNF

CNF: A conjunction of clauses, each a disjunction of literals. **Steps (Simplified):**

- Remove $\leftrightarrow$ and $\rightarrow$ using equivalences.
- Push negations to atoms (De Morgan).
- Distribute $\vee$ over $\wedge$.

Quick Example:

$$(p \rightarrow (q \vee r)) \wedge (\neg q \rightarrow \neg p)$$

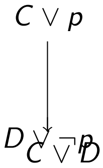
$= (\neg p \vee q \vee r) \wedge (q \vee \neg p)$. This final form is ready for resolution.

Resolution Rule

Inference Rule

From $(C \vee p)$ and $(D \vee \neg p)$ infer $(C \vee D)$.

Intuition: Eliminates p by combining remaining literals.



Key Fact

Deriving the empty clause via resolution shows unsatisfiability.

DPLL Algorithm (Core Mechanics)

DPLL augments resolution with search and propagation:

temsep=4pt**Unit Propagation:** If clause has single literal, assign it true.temsep=4pt**Pure Literal Elim:** If var appears only positive/negative, assign accordingly.temsep=4pt**Decision:** Guess a var value and recurse.temsep=4pt**Backtracking:** On conflict (empty clause), undo last guess.

Practical Note: Modern solvers add learning and heuristics on top.

Example: Manual Resolution

Test CNF:

$$(p \lor q) \land (\lnot p \lor r) \land (\lnot q \lor \lnot r).$$

temsep=4ptResolve on p: $(p \lor q), (\lnot p \lor r) \to (q \lor r)$.temsep=4ptThen with $(\lnot q \lor \lnot r)$ on q: $(q \lor r), (\lnot q \lor \lnot r) \to (r \lor \lnot r)$. (tautology)temsep=4ptAlternative resolution yields only tautologies.temsep=4ptNo empty clause produced $\Rightarrow$ formula is SAT.

Example: DPLL Walkthrough

Formula as before.

- Guess $p = \text{True}$.
- Unit-propagate $r = \text{True}$ from $(\neg p \vee r)$.
- Unit-propagate $q = \text{False}$ from $(\neg q \vee \neg r)$.
- All clauses true $\Rightarrow$ SAT assignment found.

In-Class Exercise

- **CNF Conversion:** $(p \leftrightarrow (q \wedge \neg r)) \rightarrow s$.
- **Resolution:** Add $\neg s$ to your CNF; check for conflict.
- **DPLL Simulation:** On $(p \vee q \vee r) \wedge (\neg p \vee q) \wedge (\neg q \vee r)$, show two decision paths.

Summary Takeaways

temsep=6ptCNF is the lingua franca of SAT solving.temsep=6ptResolution provides a simple proof mechanism.temsep=6ptDPLL uses search + propagation to solve SAT efficiently.temsep=6ptReal solvers integrate learning and heuristics for scale.

References I

- M. Huth M. Ryan, *Logic in Computer Science*, Cambridge UP, 2004.
- S. Russell P. Norvig, *AI: A Modern Approach*, 4th ed., Pearson, 2020.
- A. Biere et al., *Handbook of Satisfiability*, IOS Press, 2009.
- R. Nieuwenhuis et al., ”Solving SAT and SMT by Lazy Grounding,” IJCAR 2006.

Thank You! Any questions?