

Notes on SAT Solving

William Schultz

April 20, 2022

1 DPLL

The original DPLL algorithm views satisfiability as a search problem over the tree of possible boolean assignments. It combines this with *conditioning* and *unit propagation* techniques. The original DPLL algorithm also included *pure literal elimination*, which simply sets a variable to true if it only ever appears in a non-negated form. In its most basic form, given a CNF formula Δ , we start by assigning some variable a value of *true* or *false*, and then simplify the overall formula. We say that Δ is *conditioned* on a literal L , notated as $\Delta|L$, if we replace the value of literal L with *true*, and the value of literal $\neg L$ with *false*. After assigning a literal with a concrete value, we simplify the overall formula. If we simplify to something that is satisfied, then we're done, and can return a satisfying assignment. If we ever reach a state where all clauses cannot be satisfied, then we return unsatisfiable. In addition to this simple recursive search and formula simplification approach, DPLL included two basic rules for optimizing the search:

1. *Unit propagation*: If a unit clause (consisting of a single variable) appears in a formula, then assign the literal's variable to make it true, and simplify again.
2. *Pure literal elimination*: If a propositional variable appears with only one polarity (i.e either only non-negated or negated), then we assign it to make all clauses containing it true, and simplify.

Combining the basic recursive search and backtracking technique the above two rules for formula simplification constitutes the core of the basic DPLL algorithm.

2 CDCL

3 Random Satisfiability

Almost every 3-SAT formula is satisfiable when its ratio of (number of variables / number of clauses), is less than a value of about 4.25.