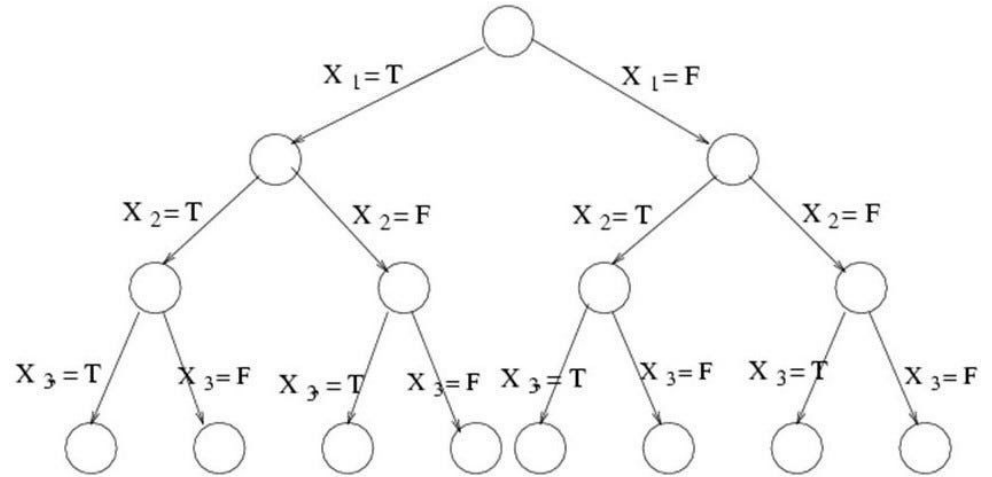


SAT Solver - Variant 2

Lukas Kaas Andersen

What is it?

- Aims to solve the boolean satisfiability problem
- Determines whether a Boolean Formula (CNF) can be satisfied with boolean assignments
- NP-Complete - Nondeterministic polynomial time complete



<https://medium.com/introduction-to-knowledge-representation/propositional-logic-and-sat-problems-c0a3edac6a26>

Conjunctive Normal Form (CNF)

- Literals: Variable or its Negation
 - $x, -x$
- Clauses: Disjunction of Literals
 - $x \vee -y$
- Conjunction (AND): A conjunction of clauses forming a CNF
 - $(x \vee y) \wedge (-x \vee y \vee z)$

DPLL Algorithm

- David-Putman-Logemann-Loveland, names of researchers
- Algorithm for solving SAT
- Depth First Search, operates in CNF
- Decide (Variable and it's Value), Deduce (Unit Clause), Resolve Conflict(Backtracking)

Variant 2 - Tuned to Randomly Generate imbalance Instances

- Variables in input CNF will appear more often in either polarity, different for each variable
- For examples:
 - x appears a lot more often than $\neg x$, and $\neg y$ appears a lot more often than y in CNF Formula

Next Steps

- Fully solidify my understanding of DPLL and SAT concepts, such as backtracking, decision heuristics and conflict resolution
- Plan requirements and design for my solver
- Decide on techniques I will use
- Decide what technologies I would like to use
- Prepare to begin coding

Gantt Chart

