SAT (WIP)

Notes based on lectures for CSC 2108H
(Automated Reasoning with Machine Learning)
at the University of Toronto by Professor Xujie Si, Fall 2024

Yanning Chen

Course given: **September 09, 2024**
Last updated: **September 11, 2024**

Contents

Attention 0.1

This lecture was incomplete.

1. Syntax

Variables w, x, y, z
Literals $x, \neg y$ *(negation)*
Clauses $x \vee y$ *(disjunction)*
Formula $(x \vee y) \wedge z$ *(conjunction of disjunctions)*
Model $M = \{x \to \top, y \to \bot\}$ *(assignments)*
Result SAT/UNSAT

2. Notations

Alternative set-theoretic notations.

Literals i as x_i, $-i$ as $\neg x_i$
Clauses $\{x_1, \neg x_2, x_3\}$ or $\{1, -2, 3\}$
Formula $\{c_1, c_2, c_3\}$, e.g. $\{1, -2, 3\}, \{-1, 2, -3\}$
Model $\{x_1 \to \top, x_2 \to \bot, x_3 \to \top\}$ or $\{1, -2, 3\}$

2.1. DIMACS

```
c Comment
c DIMACS
p cnf 3 2
1 2 -3 0
-2 3 0
c Solution:
s 1 -2 3
```

3. Preprocessing

1. Remove pure literals (e.g. $\top$)
2. Remove tauto clauses (e.g. $x_1 \lor \neg x_1 \lor x_2$)
3. Subsumption: If there's a formula $(c_1 \land c_2)$, and $c_1 \Rightarrow c_2$ then we can remove c_2 (e.g. $x_1 \Rightarrow (x_1 \lor x_2)$)
4. Unit propagation *a.k.a. Boolean Constant Propagation* (e.g. a clause consists of a single lit x, then x must be true)

4. DPLL

Given a formula G,

1. Do *BCP* **Can be optmized**
2. if $G = \top$ return True, or $G = \bot$ return False
3. $p \leftarrow \text{Choose}(G)$ (choose a variable to set) **Can be optimized**
4. return $\text{DPLL}(G\{p \to \top\}) \lor \text{DPLL}(G\{p \to \bot\})$

4.1. Optimize BCP

2-watched lit pick two lits to watch for each clause, because if a clause with two or more non-falsified lits, it doesn't affect the search

TODO add more details

4.2. Optimize Choose (branching heuristics)

Intuition: Pick the most *active* lit.

But, how to define *active*?

DLIS Count apperance of all lits and pick the *most appeared* lit

VSIDS Make a score for all vars, and add score for vars *visited in a learnt clause*. *Decay* the score for all vars over time.

5. CDCL

5.1. Decision Levels

The n-th variable we guess (choose, i.e. unit propagated vars do not count) is at level n.

TODO add decision level figure

5.2. Horn Clause

Horn Clause — Definition 5.2.1

A clause with at most one positive literal

Example 5.2.1

$\neg 1 \lor \neg 2 \lor \neg 3 \lor \ldots \lor h$ is a Horn clause, because other than h all literals are negated

The most interesting part about Horn clauses is that they can be represented as an implication: $\neg 1 \vee \neg 2 \vee \neg 3 \vee ... \vee h$ is equivalent to $1 \wedge 2 \wedge 3 \wedge ... \Rightarrow h$

We'll see how this is useful in the next section.

5.3. Key Idea of CDCL

Intuition: Suppose at decision level n we find a conflict. We want to memorize this and never make the same mistake again.

TODO add image here

A naive approach is: we just negate the guess, e.g. Made guess $\{1, 8, -7\}$, we know that $\neg x_1 \vee \neg x_8 \vee x_7$.

With the naive approach, we are back-tracking chronologically, i.e. back-tracking to the last decision level. E.g. we've just assigned $x_7 \rightarrow \perp$, and there's a conflict, we back-track to $x_7 \rightarrow \top$.

Now we claim that we can do better: what if we can find the minimum clause (cut) that causes the conflict?

5.4. Non-chronological backtracing

Given the cut, rather than back-tracking to the last decision level, we can back-track using the cut.

Like, if we are now at $1, 8, -7$, normally we will try $1, 8, 7$, but with a cut of $1, 4$ we can try $1, -4$.