Solving Satisfiability

Hans de Nivelle, ZOSIA 2017, 03.03.2017.

Introduction

I explain modern techniques for testing satisfiability of propositional formulas.

Satisfiability testing for propositional formulas is NP-complete. It therefore unlikely that there exists a polynomial algorithm.

Nevertheless, much progress has been made in recent ($\geq$ 1995) years, and modern SAT-solvers are able to solve impressively large formulas. Big enough to be useful in industrial applications.

We Think Big, we Think Great, we Think Huge !!!

What are the most important (greatest) algorithms?

- addition,subtraction,multiplication,division in binary (decimal) number system.
- quicksort.
- Dijkstra's algorithm.
- Gauss elimination, matrix multiplication, matrix application.

Other candidates: QR-algorithm? Runge-Kutta? FFT? Euclidean algorithm?

SAT solving

Let C be a set of sets of non-zero integers. We call the sets in C clauses.

Task: Find a set of integers I, such that

1. I does not contain i and $-i$ at the same time.
2. I contains at least one element from every clause.

(Pick a number from each clause in such a way that no two opposite numbers are chosen).

Example

$$\begin{array}{cc} 1 & 2 \\ -1 & 2 \\ 1 & -2 \\ -1 & -2 \end{array}$$

Satisfiable?

Satisfiable?

$$\begin{array}{ccc} 1 & -2 & \\ 2 & 3 & \\ 2 & -4 & \\ -1 & -3 & 4 \end{array}$$

Satisfiable!

$$1 \quad -2$$
$$2 \quad 3$$
$$2 \quad -4$$
$$-1 \quad -3 \quad 4$$

Add the negated solution:

$$1 \quad -2$$
$$2 \quad 3$$
$$2 \quad -4$$
$$-1 \quad -3 \quad 4$$
$$2 \quad -3 \quad 4 \quad 1$$

Let's try again: Satisfiable?

Satisfiable!

$$1 \quad -2$$
$$2 \quad 3$$
$$2 \quad -4$$
$$-1 \quad -3 \quad 4$$
$$2 \quad -3 \quad 4 \quad 1$$

Add negation again:

$$1 \quad -2 \qquad\qquad -1 \quad -3 \quad 4$$
$$2 \quad 3 \qquad\qquad 2 \quad -3 \quad 4 \quad 1$$
$$2 \quad -4 \qquad\qquad -1 \quad -2 \quad 3$$

Satisfiable? (becoming a bit like Russian roulette)

Still Satisfiable!

$$\begin{array}{ll} 1 & -2 \\ 2 & 3 \\ 2 & -4 \end{array} \qquad \begin{array}{llll} -1 & -3 & 4 & \\ 2 & -3 & 4 & 1 \\ -1 & -2 & 3 & \end{array}$$

We keep on gambling. Let's add the negation again:

$$\begin{array}{ll} 1 & -2 \\ 2 & 3 \\ 2 & -4 \end{array} \qquad \begin{array}{llll} -1 & -3 & 4 & \\ 2 & -3 & 4 & 1 \\ -1 & -2 & 3 & \\ -1 & -2 & -3 & -4 \end{array}$$

Satisfiable?

Game Over! Game Over! Game Over! Game Over!

No, this time the problem is unsatisfiable.

What did we see?

Checking solutions is easy, even a drunk Polish audience can do it.

Finding solutions is not so easy. One has to try all possibilities.

Why? Nobody knows!

Satisfiability is NP-complete.

'In NP' means: Solutions can be easily verified. 'complete' means: All problems with similar behaviour can be translated into it.

Satisfiability

Given a set of clauses, decide if an intepretation exists. If yes, give an interpretation.

Dumb Algorithm: For every number n occurring in the problem, try n and $-n$.

Cost is 2^N, where N is the number of variables used.

Decide

Pick a natural number n, and add either n or $-n$ to the interpretation.

Our problem was:

$$\begin{array}{cc} 1 & -2 \\ 2 & 3 \\ 2 & -4 \end{array} \qquad \begin{array}{cccc} -1 & -3 & 4 & \\ 2 & -3 & 4 & 1 \\ -1 & -2 & 3 & \\ -1 & -2 & -3 & -4 \end{array}$$

Interpretation is ( ). We decide to add 1 because it is first number. We add it positively, because such is our nature. Now we have $I = (1)$.

It doesn't help much. Next we decide 2

Unit Propagation

We have

$$I = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Our problem was:

$$\begin{matrix} 1 & -2 \\ 2 & 3 \\ 2 & -4 \end{matrix} \qquad \begin{matrix} -1 & -3 & 4 & \\ 2 & -3 & 4 & 1 \\ -1 & -2 & 3 & \\ -1 & -2 & -3 & -4 \end{matrix}$$

Because of clause (-1 -2 3), we have to add 3 to I. Because of (-1 -2 -3 -4), we have to add -4 to I. This is called unit propagation.

Conflict

Now we have

$$I = \left(\begin{array}{c|cc} 1 & & \\ 2 & 3 & -4 \end{array} \right).$$

The clause (-1 -3 4) is false. This is called a conflict.

What now?

Go back to the last choice 2 and set it to -2.

But not without learning a lesson!

Resolution

Definition: Let c_1 and c_2 be two clauses. Let n be a number. Then the clause $(c_1 \backslash \{n\}) \cup (c_2 \backslash \{-n\})$ is called resolvent of c_1 and c_2.

We write $\mathrm{RES}(n, c_1, c_2)$ for the resolvent.

$$\left.\begin{array}{cccc} 1 & 2 & 4 & -3 \\ 4 & -2 & 4 & \end{array}\right\} \Rightarrow \quad 1 \quad -3 \quad 4$$

If c_1, c_2 are true in an interpretation, then $\mathrm{RES}(n, c_1, c_2)$ is also true. By adding resolvents, you will not lose solutions.

Conflict Clause

Whenever we reverse a unit propagation step, we can obtain a false clause by resolution:

$$I = \left(\begin{array}{c|cc} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & \end{array}\right) \Leftarrow \left(\begin{array}{c|cc} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{array}\right)$$

There must be a clause $(-x_1 \;\; -x_3 \;\; -y_3 \;\; z_3)$, that was used in the propagation. We assume that the extended interpretation has a conflict clause $(-z_3 \;\; -x_3 \;\; -y_2)$. We can resolve on z_3, and obtain $(-x_1 \;\; -x_3 \;\; -y_2 \;\; -y_3)$.

Undoing Unit Propagation

Our problem was:

$$\begin{array}{cc} 1 & -2 \\ 2 & 3 \\ 2 & -4 \end{array} \qquad \begin{array}{cccc} -1 & -3 & 4 & \\ 2 & -3 & 4 & 1 \\ -1 & -2 & 3 & \\ -1 & -2 & -3 & -4 \end{array}$$

Interpretation was

$$I = \left(\begin{array}{c|cc} 1 & & \\ 2 & 3 & -4 \end{array} \right).$$

The clause (-1 -3 4) is false. We used (-1 -2 3) and (-1 -2 -3 -4) in the propagations.

Resolve (-1 -3 4) with (-1 -2 -3 -4) into (-1 -2 -3). Resolve result with (-1 -2 3). This gives (-1 -2).

Learning

$$I = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

We have obtained (-1 -2). Decision 2 was clearly wrong. Unit propagation with (-1 -2) shows that it should have been -2. We are very sorry. We will never make such mistake again.

Add (-1 -2) to the problem set, and correct I.

$$\begin{array}{cc} 1 & -2 \\ 2 & 3 \\ 2 & -4 \\ -1 & -2 \end{array} \qquad\qquad \begin{array}{cccc} -1 & -3 & 4 & \\ 2 & -3 & 4 & 1 \\ -1 & -2 & 3 & \\ -1 & -2 & -3 & -4 \end{array}$$

$$I = \left(\begin{array}{c|c} 1 & -2 \end{array} \right).$$

$$\begin{array}{cc} 1 & -2 \\ 2 & 3 \\ 2 & -4 \\ -1 & -2 \end{array} \qquad\qquad \begin{array}{cccc} -1 & -3 & 4 & \\ 2 & -3 & 4 & 1 \\ -1 & -2 & 3 & \\ -1 & -2 & -3 & -4 \end{array}$$

$$I = \left(\begin{array}{c|c} 1 & -2 \end{array} \right).$$

Unit propagation with (2 3) gives $\left(\begin{array}{c|cc} 1 & -2 & 3 \end{array} \right)$.

Unit propagation with (2 -4) gives $\left(\begin{array}{c|ccc} 1 & -2 & 3 & -4 \end{array} \right)$. Now clause (-1 -3 4) is false.

We can resolve back (-1 -3 4) $\Rightarrow$ (-1 2 -3) $\Rightarrow$ (-1 2) $\Rightarrow$ (-1). We learn this clause. Decision 1 was wrong, etc.

Unique Implication Points (UIP)s

We know how to reverse propagations and produce conflict clauses. When to stop? One can always stop at a decision. Better: Stop at a UIP.

$$I = \left(\begin{array}{c|cc} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & \end{array} \right) \Rightarrow \left(\begin{array}{c|cc} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{array} \right)$$

As before, assume that extended interpretation has conflict clause $(-z_3 \;\; -x_3 \;\; -y_2)$. Forget about how z_3 was obtained. Ignore the last decision. Just learn this clause and backtrack to

$$I = \left(\begin{array}{c|ccc} x_1 & x_2 & x_3 & \\ y_1 & y_2 & y_3 & -z_3 \end{array} \right) \Leftarrow$$

Implementation

Unit propagations are put in a graph. (implication graph). Instead of doing the individual steps, analyze the graph, and see where the best UIP is. Then create the learned clause at once.

Two literal watching. Clauses can be long, and there are many. After every extension of I one has to look for possible propagations and conflicts. Pick two literals from each clause, and look at the clause when one of the literals gets assigned.

Good forgetting is useful as learning.

Question

Should DPLL+CDCL be inserted into the gallery of great algorithms?

Some Final Remarks

Polish science is still too theoretical.

- Programming is trivial, dirty handwork that should be left to lower personel.
- Only proofs matter.
- A harder result is a better result.

How it should be: In good research, the theory guides the implementation, and the implementation guides the research.

Theoretical complexity is bad measure for practicality.

Computer Science is similar to physics (mathematics + experiment).