2D Project — 50.001 SAT Solver

Ragul Balaji James Raphael Tiovalen Anirudh Shrinivason Jia Shuyi Gerald Hoo Shoham Chakraborty

Result Overview

Test file:	`test_2020.cnf`
Result:	not satisfiable
Running time:	3488.9 ms
Specification:	Macbook Pro 13' 2019, 1.4 GHz Quad-Core Intel Core i5

Approach

We employed the DPLL algorithm to solve the SAT problem.

For a given formula $\mathbb{F}$, if there are no clauses present, we know that $\mathbb{F}$ is trivially satisfiable.

If we have at least one clause in $\mathbb{F}$, then we find the smallest clause. If any clause in $\mathbb{F}$ is empty (defined to be FALSE due to our implementation of the `substitute` method, explained below), we backtrack to the previous level.

If the smallest clause found is of unit length, we set its only literal to TRUE and use the `substitute` method to simplify $\mathbb{F}$. We then recursively call the `solve` method on the simplified $\mathbb{F}$.

Else, we look for the next shortest clause. We will then pick an arbitrary literal from this clause and set the literal to TRUE. We again use the `substitute` method to simplify $\mathbb{F}$ and solve recursively. If we do not find any conflict when all the clauses are satisfied, we have found a solution. However if a conflict is detected, we backtrack and change the assigned literal to FALSE, simplify $\mathbb{F}$ and recursively solve. We repeat this until we either find a solution, or we exhaust our options.

```
function SUBSTITUTE(ℂ, l)
// ℂ : a list of clauses
// l : input literal to simplify ℂ
for c in ℂ do
    if c contains l then
        remove c from ℂ
    else if c contains ¬l
        remove ¬l from c
    end if
end for
```

To understand our `substitute` method, we use the simple example of $\mathbb{C} = (X \lor Y) \land (\neg Y \lor Z) \land (Y)$ and $l = Y$. Since the first clause $(X \lor Y)$ contains Y, the clause will be evaluated to TRUE no matter what value X takes, therefore we will delete the entire clause. By the same logic, we will also remove the last clause (Y). Since the negation of Y is present in the second clause $(\neg Y \lor Z)$, , the evaluation of this clause will solely depend on Z since $\neg Y =$ FALSE. Thus, $\mathbb{C}$ is simplified to the form (Z). We then return this simplified $\mathbb{C}$ to the `solve` method for further computation.