3CNFSAT satisfiability and the P vs NP problem

Hamza Benyahya

June 27, 2021

In this paper we will study the properties of writing a 3CNFSAT formula, this will enable us to instore a set of constraints which we'll follow to deduce an optimal writing form of 3CNFSAT that fulfills the property of maximum complexity of satisfiability. By solving this signature writing we can result to the cap polynomial time siffucient to prove the satisfiability of all remaining writing possibilities of the formula that use the number of literals figuring in the signature writing or less. The proof of SAT to be an NP-complete problem by the Cook-Levin theorem allows us to reduce every decision problem in the complexity class NP to the SAT problem for CNF formulas (CNFSAT), and the reduction of the unrestricted SAT problem to a conjunctive normal form with each clause containing at most three literals (3CNFSAT) allows us to deduce that determining the satisfiability of 3CNFSAT formulas is also NP-complete. By increasing the length of the signature formula through conjucting n of it's duplications while introducing new literals for each duplication, we can study the evolution of the cap polynomial time in function of n, thus resulting to a solution for P vs NP.

1 3CNFSAT satisfiability

To identify the constraints that will allow us to maximise the complexity of 3CNFSAT, we need to study the requirements to prove it's satisfiability.

1.1 Satisfiability constraint

A 3CNFSAT formula is satisfiable only when:

- There is at least one literal's state x or $\bar{x}$ inside each clause C of the formula that is *TRUE*.
- For each literal's state x or $\bar{x}$, this state keeps it's value *TRUE* or *FALSE* immutable through the formula.

1.2 Satisfiability proof

When deducing the satisfiability of a clause $C = (x \vee y \vee z)$ constructing a 3CNFSAT formula, it can figure in only one of the following states at a time:

1. The value of all literals x, y and z are still unknown, which implies that satisfying the clause depends on deducing the value of at least one of the literals being *TRUE* through studying the satisfiability of the remaining clauses of the formula.
2. At least one of the literals' value is known, which results to:
   (a) At least one of the known literals' value is *TRUE* and that's sufficient to satisfy the clause.
   (b) All the known literals' value is *FALSE* and the satisfiability of the clause depends on the value of the remaining literals.
3. The value of all literals x, y and z is *FALSE*, and the clause is not satisfiable.

For 3CNFSAT to be polynomialy satisfiable means that every form of 3CNFSAT that accepts a solution for it's satisfiability, this solution can be determined in a polynomial time, which implies that a signature writing of the formula that has maximum complexity of satisfiability, the latter can be determined in a polynomial time.

Based on the identified satisfiability constraints we can deduce a signature writing of 3CNFSAT that fulfills the property of maximum complexity of satisfiability.

2 3CNFSAT-Complex

To determine 3CNFSAT-Complex that is the signature writing of 3CNFSAT with maximum complexity of satisfiability we need to maximally increase

the complexity of deducing the satisfiability of all clauses of the formula and by extension maximally increase the complexity of deducing the value of all literals constructing these clauses.

2.1 Literal complexity

To maximize the complexity of determining the value of a literal, the latter needs to fulfill the following constraints:

- A literal's states should figure at least once through the formula. (1)
- A literal's states should figure with maximum possible other literals' states to increase the constraints of determining its or other literals value. (2)
- A literal should not figure with the same set of literals disregarding their states in two distinct clauses to not decrease the latters' conjunction satisfiability constraints rendering it depending only on at most two literals. (3)

2.2 3CNFSAT-Complex demonstration

We'll initiate the construction of 3CNFSAT-Complex by introducing a clause using 3 different literals.

$$(x_1 \vee x_2 \vee x_3)$$

In it's current form the satisfiability of the formula depends only on the constraint that at least one of the literals being $TRUE$ and all literals x_1, x_2 and x_3 do not attend maximum complexity for their lack of fulfilling (1) and (2), thus the introduction of a conjunction of a new clause to increase their complexity.

$$(x_1 \vee x_2 \vee x_3) \wedge (a \vee b \vee c)$$

The nature of each of a, b and c is yet to determine if it's an introduction of a new literal or a reuse of an already existing one while preferencing the latter to fulfill (2).

We can fulfill (1) for x_1 by implying that a references $\bar{x_1}$. (a)

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee b \vee c)$$

Implying that b references $\bar{x_2}$ conflicts with (2) by decreasing the complexity of deducing x_1 and x_2 giving up that $x_1 = \bar{x_2}$ for the first and second clauses not being satisfied by respectively x_3 and c. (b)

Knowing that (a) and (b) could be applied interchangeably to x_1, x_2 and x_3, we deduce that we can only fulfill (1) for all of them by introducing three conjunctions of new clauses to the first clause with each one containing a different literal from it in it's opposite state.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee a \vee b) \wedge (\bar{x_2} \vee c \vee d) \wedge (\bar{x_3} \vee e \vee f)$$

The nature of each of a, b, c, d, e and f is yet to determine if it's an introduction of a new literal or a reuse of an already existing one while preferencing the latter to fulfill (2).

We can fulfill (2) for x_1 by implying that a references x_2.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee x_2 \vee b) \wedge (\bar{x_2} \vee c \vee d) \wedge (\bar{x_3} \vee e \vee f)$$

Implying that b references x_3 conflicts with (3) by decreasing the conjunction of the first and second clauses' satisfiability constraints resulting them depending only on x_2 and x_3.

Implying that b references $\bar{x_3}$ conflicts with (3) by decreasing the conjunction of the first and second clauses' satisfiability constraints resulting them depending only on x_2.

We conclude that b is an introduction of a new literal x_4.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee x_2 \vee x_4) \wedge (\bar{x_2} \vee c \vee d) \wedge (\bar{x_3} \vee e \vee f)$$

We can fulfill (2) for $\bar{x_2}$ while fulfilling (1) for x_4 by implying that c references $\bar{x_4}$.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee x_2 \vee x_4) \wedge (\bar{x_2} \vee \bar{x_4} \vee d) \wedge (\bar{x_3} \vee e \vee f)$$

Implying that d references x_1 conflicts with (3) by decreasing the conjunction of the second and third clauses' satisfiability constraints resulting them depending only on two of the three literals x_1, x_2 and x_4.

Implying that d references $\bar{x_1}$ conflicts with (3) by decreasing the conjunction of the second and third clauses' satisfiability constraints resulting them depending only on x_1.

Implying that d references x_3 conflicts with (2) by decreasing the complexity of deducing x_3 giving up that $x_3 = TRUE$ for the first and third clauses not being satisfied by respectively x_1 and $\bar{x_4}$.

Implying that d references $\bar{x_3}$ conflicts with (2) by decreasing the complexity of deducing x_2 and x_3 giving up that $x_2 = \bar{x_3}$ for the first and third clauses not being satisfied by respectively x_1 and $\bar{x_4}$.

We conclude that d is an introduction of a new literal x_5.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee x_2 \vee x_4) \wedge (\bar{x_2} \vee \bar{x_4} \vee x_5) \wedge (\bar{x_3} \vee e \vee f)$$

We can fulfill (2) for x_3 while fulfilling (1) for x_5 by implying that e references $\bar{x_4}$ and f references $\bar{x_5}$.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee x_2 \vee x_4) \wedge (\bar{x_2} \vee \bar{x_4} \vee x_5) \wedge (\bar{x_3} \vee \bar{x_4} \vee \bar{x_5})$$

The current form of the formula fulfills (1), (2) and (3) for all it's literals.

We observe by analyzing each clause for satisfiability that all literals' states figuring through the formula have a chance of being either *TRUE* or *FALSE* and each clause depends on at least one other clause to deduce the values of it's literals' states with no possibility of deducing them directly, confirming them achieving maximum complexity of determining their values and by extension confirming that the resulting signature writing is 3CNFSAT-Complex.

2.3 Demonstration result

3CNFSAT-Complex is written in the unique form of:

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee x_2 \vee x_4) \wedge (\bar{x_2} \vee \bar{x_4} \vee x_5) \wedge (\bar{x_3} \vee \bar{x_4} \vee \bar{x_5})$$

With the use of:

- Four clauses.
- Five distinct literals.
- One reuse of two distinct literals with each reuse introduced in a state.

3 3CNFSAT-Complex resolution

To solve 3CNFSAT-Complex, we proceed by solving the clause which its most literals are dependable to solve other clauses.

Since 3CNFSAT-Complex is composed of four clauses with each clause containing three literals, the best-case scenario would be having a clause with each literal figuring in one other distinct clause, which is fulfilled by the first clause $C_1 = (x_1 \vee x_2 \vee x_3)$.

There exists three possibilities to satisfy C_1:

1. Satisfy C_1 by x_1:
   - Implies that C_2 needs to be satisfied by either x_2 or x_4, meaning that the value of x_1 is depending on the values of x_2 and x_4. (1)
2. Satisfy C_1 by x_2:
   - Implies that C_3 needs to be satisfied by either $\bar{x_4}$ or x_5, meaning that the value of x_2 is depending on the values of x_4 and x_5. (2)
3. Satisfy C_1 by x_3:
   - Implies that C_4 needs to be satisfied by either $\bar{x_4}$ or $\bar{x_5}$, meaning that the value of x_3 is depending on the values of x_4 and x_5. (3)

From (1) and (2) we deduce that we can't satisfy two different clauses by both x_4 and $\bar{x_4}$.

From (2) and (3) we deduce that we can't satisfy two different clauses by both x_5 and $\bar{x_5}$.

From (1), (2) and (3) we know that to satisfy C_1:

- $x_1 = \mathit{TRUE}$ and $x_4 = \bar{x_5}$. (4)

From (2) and (3) we know that we can't satisfy C_1 by x_2 or x_3:

- $x_2 = \mathit{FALSE}$ and $x_3 = \mathit{FALSE}$.

To determine the values of x_4 and x_5 we need to use the maximum of literals with known values (x_1, x_2 and x_3) with the minimum of literals with unknown values figuring at the same time in a clause.

Since a clause contains three literals, the best-case scenario would be two literals with known values and one literal with unknown value, which is fulfilled by the second clause $C_2 = (\bar{x_1} \vee x_2 \vee x_4)$.

Satisfying C_2 means that $x_4 = \mathit{TRUE}$, and using (4) we deduce that $x_5 = \mathit{FALSE}$.

3.1 Resolution result

3CNFSAT-Complex's satisfiability accepts only one unique solution of:

- $x_1 = \mathit{TRUE}$
- $x_2 = \mathit{FALSE}$
- $x_3 = \mathit{FALSE}$
- $x_4 = \mathit{TRUE}$
- $x_5 = \mathit{FALSE}$

4 P vs NP

We use:

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee x_2 \vee x_4) \wedge (\bar{x_2} \vee \bar{x_4} \vee x_5) \wedge (\bar{x_3} \vee \bar{x_4} \vee \bar{x_5})$$

Since we demonstrated that ϕ is written in its most complex form means that increasing the length of the formula while utilizing only the five literals x_1, x_2, x_3, x_4 and x_5 doesn't affect in any way the complexity of deducing its satisfiability, since it would depend only on verifying the existance of the four clauses C_1, C_2, C_3 and C_4 then deducing the satisfiability of the remaining clauses by checking them against the values of the unique result:

- $x_1 = \mathit{TRUE}$
- $x_2 = \mathit{FALSE}$
- $x_3 = \mathit{FALSE}$
- $x_4 = \mathit{TRUE}$

- $x_5 = FALSE$

Increasing the length of the formula by introducing new literals recursively implies that for each set of newly added clauses constructed with up to 5 distinct literals its satisfiability complexity is capped by 3CNFSAT-Complex's satisfiability complexity meaning that the satisfiability complexity of a 3CNFSAT formula using n distinct literals is capped by the satisfiability complexity of the conjunction of m 3CNFSAT-Complex formulas with $(n \leqslant 5m \leqslant n + 4)$ and since each instance of 3CNFSAT-Complex doesn't share its literals with other instances we deduce that the complexity of their conjunctions doesn't increase from the complexity of satisfying one instance and that the polynomial time depends only on the length of the formula to verify the satisfiability of its instances, thus an algorithm that takes t polynomial time to deduce the satisfiability of 3CNFSAT-Complex will require mt polynomial time to satisfy the conjunction of m 3CNFSAT-Complex formulas.

We result that for a 3CNFSAT formula written as a conjunction of n 3CNFSAT-Complex formulas, the increase of complexity in function of length is represented by the function $f(n) = a$ while the necessary polynomial time to verify the satisfiability of the formula in function of length is represented by the function $g(n) = nt$ with a representing the complexity of deducing the satisfiability of 3CNFSAT-Complex and t representing the polynomial time to satisfy 3CNFSAT-Complex. By observing the growth rate of the two functions $f(n)$ and $g(n)$ we observe that $f(n)$ has a constant growth rate and $g(n)$ has a linear growth rate meaning that the growth rate of $f(n)$ will always be slower than the growth rate of $g(n)$, Thus $P \neq NP$.

5 Conclusion

By studying the properties of writing 3CNFSAT formulas, we were able to deduce the signature writing 3CNFSAT-Complex and by solving the latter we resulted the existance of the cap polynomial time necessary to prove its satisfiability and by extension the polynomial time siffucient to prove the satisfiability of all remaining writing possibilities of 3CNFSAT that use the number of literals figuring in 3CNFSAT-Complex or less. We demonstrated that for every satisfiable 3CNFSAT formula, its complexity of deducing its satisfiability is capped by the complexity of deducing the satisfiability of a number of conjunctions of 3CNFSAT-Complex and by studying the evolution

of the cap polynomial time necessary to satisfy 3CNFSAT-Complex while increasing its length through conjucting a number of its duplications with the introduction of new litterals for each duplication, we were able to decide the result of P vs NP.