Computability and Complexity: $\neq$SAT

Hakim Boulahya
Université Libre de Bruxelles

December 16, 2017

Proof Idea

To proof that $\neq$ SAT is NP-complete we have to prove that a NP problem is reducible in polynomial time to $\neq SAT$.

a ou b ou c

f, f, f is not valid because it doesnt satisfy

List:

* Look at the bookmarks * Proove NP-Complete: Proove that it is 3SAT by adding more formulas confirming the !=assign

Proof

Hint proof

If we have a diffassign that satifsfy the formula, it means that in each clause there is either two true with a false two false with a true. By taking the negation it will jjust convert the 2t1f to a 1t2f and the 2f1t to 1f2t which also satisfy the formula.

To proof that it is NP-complete we have to proove that all NP problem are reducible in P time to diffSAT. Since 3SAT is NP-complete, by reducing it to diffSAT it will prove that diffSAT is NP-Complete.

Note about the hint: the idea is to convert each clause into diffSAT. Since we known that if a diffassign sat a clause, it will also sat the neg of the clause.

3SAT clause: c = (x y z)

The idea is if (x y z) is true, i.e. an assignment sat the formula, we must find a 3cnf-formula that is true also but never contains three true in any clauses.

Actually when converting we must fine a formula that if c is sat, then there exist a diffassing that sat the new formula.

1 1 1 1 1 0 1 0 1 1 0 0 0 1 1 0 1 0 0 0 1 0 0 0

if c = T

(c x y) (nc z F)