

CNF · SAT

Anatoly Weinstein

January 9, 2024

TODO A CNF-SAT is an equation ...

1 Equations

Set Notation · CNF	$\{\{a, \bar{b}\}, \{b, \bar{c}, \bar{d}\}, \{\bar{a}, \bar{c}\}\}$
Arithmetic · CNF	$(a + \bar{b}) \cdot (b + \bar{c} + \bar{d}) \cdot (\bar{a} + \bar{c})$
Negated · DNF	$\neg(ab + \bar{b}cd + ac)$

Formulas

Zugzwang. If a disjunction consists of one literal, the value of this literal is fixed.

$$\{X\} \in CNF \Rightarrow X = \top$$

Absorption (CNF). If a disjunction is a superset of another disjunction, it can be ignored.

$$\forall X \subseteq Y : X \cdot Y \equiv_{CNF} X$$

Proof: $(X \Rightarrow Y) \vee \neg X$ makes Y irrelevant.

		X				
Y		<table><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>				

The Absorption Rule allows to freely create new disjunctions, as long it's consisting of an already existing disjunction.

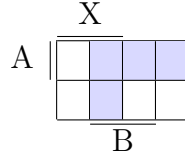
Absorption (DNF).

$$\forall X \subseteq Y : X + Y \equiv_{\text{DNF}} X$$

Proof: $(X \Rightarrow \top) \wedge (\neg X \Rightarrow \neg Y)$

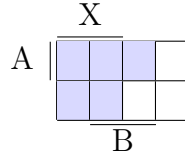
Common Contradiction.

$$(A + X) \cdot (B + \overline{X}) \equiv_{\text{CNF}} (BX + A\overline{X})$$



Common Part.

$$(A + X) \cdot (B + X) \equiv_{\text{CNF}} (X + AB)$$



Semi-Common Part.

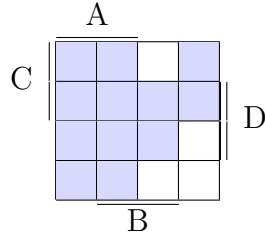
$$(AX + B)(A + C) \equiv_{\text{CNF}} A(X + B) + BC$$

Proof:

$$\begin{aligned} & (AX + B)(A + C) \\ \equiv_{\text{CNF}} & AX + AXC + AB + BC \\ \equiv_{\text{CNF}}^{\text{Absorption}} & AX + AB + BC \end{aligned}$$

Common and Contradicting Part

$$(A + B + C) \cdot (A + \neg B + D) \equiv_{\text{CNF}} (A + BD + C\overline{B})$$



Proof:

$$\begin{aligned}
 & (A + B + C) \cdot (A + \neg B + D) \\
 \equiv_{\text{CNF}} & (A + (B + C)(\neg B + D)) \\
 \equiv_{\text{CNF}} & (A + BD + C\bar{B})
 \end{aligned}$$

2 Algorithms

bf0 · Brute Force

1. Try out every combination of variable interpretations.

dbf · Disjunctive Brute Force

1. Use trivial optimisations.
2. Count the number of occurrences of every literals (positive and negated) and find the literal with most occurrences.
3. Branch: Assume the variable and accordingly clean up the equation using trivial transformations.
4. Absorption: Absorb as many disjunctions to the two- literals clauses.
5. If the equation is in 2-CNF-SAT, proceed with the next step, otherwise jump to the first step in a recursive step.
6. Solve 2-SAT the equation in polynomial time.