FUNDAMENTAL LAW OF BINARY THEORIES

UNIVERSAL NUMBER THEORY EQUATION OF SAT

Oscar Riveros

2016

$$\mathtt{SAT}(\mathtt{v}_0, \ldots, \mathtt{v}_{\mathtt{n-1}} | \mathtt{c}_0, \ldots, \mathtt{c}_{\mathtt{m-1}}) = \sum_{\mathtt{j=0}}^{\mathtt{m-1}} 2^{\sum_{\mathtt{i=0}}^{\mathtt{n-1}} \mathtt{v}_{(\mathtt{n-1-i})\mathtt{j}} 2^{\mathtt{i}}}$$

oscar.riveros@peqnp.com
twitter.com/maxtuno
http://soundcloud.com/maxtuno
http://independent.academia.edu/oarr
http://github.com/maxtuno/SAT_EQUATION

SAT EQUATION [1]

$$\sum_{j=0}^{m-1} 2^{\sum_{i=0}^{n-1} v_{(n-1-i)j} 2^i} = \underbrace{(v_0^{\pm} \vee \cdots \vee v_{n-1}^{\pm})}_{c_0} \wedge \underbrace{(v_0^{\pm} \vee \cdots \vee v_{n-1}^{\pm})}_{c_1} \wedge \cdots \wedge \underbrace{(v_0^{\pm} \vee \cdots \vee v_{n-1}^{\pm})}_{c_{m-1}}$$

$$\mathtt{SAT}(c_0, c_1 \ldots c_{m-1}) = \sum_{j=0}^{m-1} 2^{c_j}$$

$$c_j = \sum_{i=0}^{n-1} v_{(n-1-i)j} 2^i$$

$$v_{(n-1-i)j} = \begin{cases} 0 & +v_{(n-1-i)j} \\ 1 & -v_{(n-1-i)j} \end{cases}$$

EXAMPLE

$$(a \vee b \vee c \vee d) \wedge (a \vee \neg b \vee c \vee \neg d) \wedge (\neg a \vee \neg b \vee c \vee d)$$

d	c	b	a	value
False	False	False	False	False
False	False	False	True	True
False	False	True	False	True
False	False	True	True	True
False	True	False	False	True
False	True	False	True	False
False	True	True	False	True
False	True	True	True	True
True	False	False	False	True
True	False	False	True	True
True	False	True	False	True
True	False	True	True	True
True	True	False	False	False
True	True	False	True	True
True	True	True	False	True
True	True	True	True	True

$$m = 3, n = 4$$

$$v_{11}v_{21}v_{31}v_{41} = \{1, 1, 1, 1\}$$
$$v_{12}v_{22}v_{32}v_{42} = \{1, 0, 1, 0\}$$
$$v_{13}v_{23}v_{33}v_{43} = \{0, 0, 1, 1\}$$

$$2^{c_1} = 2^{15} = 32768$$
$$2^{c_2} = 2^{10} = 1024$$
$$2^{c_3} = 2^{3} = 8$$

$$SAT(c_1, c_2, c_3) = 32768 + 32 + 8 = 33800$$

$$33800_2 = 1000010000001000 = \begin{cases} 0 & True \\ 1 & False \end{cases}$$

$$False, True, True, True, True, False, True, True, True, True, True, True, False, True, True, True$$

IMPLEMENTATION

http://github.com/maxtuno/SAT_EQUATION

```
def bits(n, p):
    s = []
    while n:
        s = [n % 2 == 0] + s
        n //= 2
    s = (p - len(s)) * [True] + s
    return s


def sat_equation(cnf, n, m):
    sat = 0
    for j in range(m):
        v = 0
        for i in range(n):
            v += int(cnf[j][n - 1 - i] > 0) * 2 ** i
        sat += 2 ** v
    return sat


if __name__ == '__main__':
    cnf = [(1, 2, 3, 4), (1, -2, 3, -4), (-1, -2, 3, 4)]

    n = 4
    m = len(cnf)

    print(bits(sat_equation(cnf, n, m), 2 ** n))
```

DISCLAIMER

All this work is the effort of 5 years of research, everything is completely original and of my authority, no reference exist, and no exist collaborators, only elementary concepts of the theory of complexity were used. Special thanks to my wife, my favorite persons, all my friends and followers, thank for all, today is my official retirement of the sciences...