ARITHMETICA UNIVERSALIS

PROOF OF FUNDAMENTAL THEOREM OF ARITHMETICA UNIVERSALIS

Oscar Riveros

2016

To Misfits...

Proof of Arithmetic Boolean Satisfiability Equation...

1 ARITHMETIC BOOLEAN SATISFIABILITY EQUATION

Theorem.

$$\sum_{j=0}^{m-1} 2^{\sum_{i=0}^{n-1} x_{(n-1-i)j} 2^i} = (x_{0,0}^{\pm} \vee \cdots \vee x_{n-1,0}^{\pm}) \wedge \cdots \wedge (x_{0,m}^{\pm} \vee \cdots \vee x_{n-1,m-1}^{\pm})$$

Proof. A CNF clause, is $False$ only when all elements are $False$, then if $False \rightarrow 1$ and $True \rightarrow 0$ you can write any CNF clause how $(2^{|X|} - 1) - 1$ where $X = \{x_0 \ldots x_{n-1}\}$ the set of variables and the lowest significan bit, correspond to the assignment $\{x_0 = False \rightarrow 1 \ldots x_{n-1} = False \rightarrow 1\}$, repeating this process, covering all combinations:

$$\begin{array}{ccc}
0 & \underbrace{00\ldots00}_{|X|} & \{x_0 = True \rightarrow 0 \ldots x_{n-1} = True \rightarrow 0\} \\
1 & \underbrace{00\ldots01}_{|X|} & \{x_0 = True \rightarrow 0 \ldots x_{n-1} = False \rightarrow 1\} \\
\vdots & \vdots & \vdots \\
2^{|X|} - 2 & \underbrace{11\ldots10}_{|X|} & \{x_0 = False \rightarrow 1 \ldots x_{n-1} = True \rightarrow 0\} \\
2^{|X|} - 1 & \underbrace{11\ldots11}_{|X|} & \{x_0 = False \rightarrow 1 \ldots x_{n-1} = False \rightarrow 1\}
\end{array}$$

The $True$ values of a CNF clause, is given by the positive assignments of each variable, then a particular clause have $True$ values according to $\sum_{i=0}^{n-1} x_{(n-1-i)j} 2^i$ this is the binary number representation of their assignment, and then the entire CNF formula is given by the sum of this assignments over all clauses of the formula.

This proof the theorem. □

2 IMPLEMENTATION (Python)

```
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))
```

3 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...

4 LINKS

Contact me at `oscar.riveros@peqnp.com`:
`http://twitter.com/maxtuno`
`http://soundcloud.com/maxtuno`
`http://www.reverbnation.com/maxtuno`
`http://independent.academia.edu/oarr`
`http://github.com/maxtuno/SAT_EQUATION`

5 Giordano Bruno Project

Revolutionary Science for Misfits! ⇔ `http://peqnp.com`

- Read the *mandatory rules*.
- Research.
- Send Advances and Discoveries to `contact@peqnp.com`.
- Publish in the Journal of Project.