"Evrething it's an expression", it's the rule for using rules in Wolfram's mathematica software. This is my attempt for use only binary or unary expressions for a try to manipulate them.

appVersion(4) = "0.99.7561.40250"

appVersion(4) == "0.99.7561.40250"

To expression with bugs

```
ApplyRules(E, R, N) :=
:= | [x := E  y := E]
   | for NIter ∈ [1..N]
   |    for n ∈ [1..rows(R)]
   |       x := str2num(concat("equrep(", num2str(x), ",", num2str(R n 1), ",", num2str(R n 2), ")"))
   |    if num2str(x) == num2str(y)
   |       break
   |    else
   |       y := x
   | x
```

```
op2ex(n, op, fn, P) := | str1 := concat("a0", op, "a1")
                       | str2 := concat(fn, "(a0,a1)")
                       | P := stack(P, [str1  str2])
                       | for k ∈ [2..n]
                       |    | str1 := concat(str1, op, "a", num2str(k))
                       |    | str2 := concat(fn, "(", str2, ",a", num2str(k), ")")
                       |    | P := eval(stack(P, [str1  str2]))
                       | P
```

```
ApplyRules(E, R) := | ApplyRules(E, R, ApplyRulesN)
```

ApplyRules(E, R)

```
ApplyRulesN := 20
```

```
op2ex(E, R) := | ans := num2str(ApplyRules(E, R))
               | ans := str2num(strrep(ans, ",1/", ",_one/"))
               | ans := num2str(ApplyRules(ans, R))
               | ans := strrep(strrep(ans, "_times(_one,1/", "_inv("), "-", "_negone*")
               | ans := ApplyRules(str2num(ans), R)
               | [_one := 1  _negone := −1  _inv(x) := _pow(x, _negone)]
               | ans
```

```
op2ex(E) := | op2ex(E, op2ex)
```

Algebra

```
op2exN := 20        op2ex := matrix(0, 2)
```

```
op2ex := op2ex(op2exN, "+", "_plus", op2ex)
```

op2ex := op2ex (op2exN , "*" , "_times" , op2ex)

op2ex := op2ex (op2exN , "^" , "_pow" , op2ex)

op2ex := stack (op2ex , ["a/b" "_times(a,_pow(b,-1)"])

op2ex := stack (op2ex , ["0-a" "_times(a,-1)"])

Algebra Example

Convert an expression $expr := \frac{a^2 \cdot x + b + x^3 \cdot y - \frac{c}{d}}{x \cdot (1 + x)}$

This don't give the desired result

ApplyRules (expr , op2ex) = _times (_times (_times (_plus (− c , _times (d , _plus (b , _times (x , _plus (_pow (a , 2) , _times (_pow (x

This do _expr := op2ex (expr) (-1 can appear but not -something)

_expr = _times (_times (_times (_plus (_times (− 1 , c) , _times (d , _plus (b , _times (x , _plus (_pow (a , 2) , _times (_pow (x , 2) , y))))))) , _pow (d , − 1))

For recover the original expression

_times (a , b) := a · b _plus (a , b) := a + b _pow (a , b) := a^b

$_expr = \frac{-c + d \cdot \left(b + x \cdot \left(a^2 + x^2 \cdot y\right)\right)}{d \cdot x \cdot (1 + x)}$ $expr - _expr = 0$

Clear (_times (a , b) , _plus (a , b) , _pow (a , b)) = 1

_exp

Boolean Algebra

op2ex := op2ex (2 , "&" , "_and" , op2ex)

op2ex := op2ex (2 , "|" , "_or" , op2ex)

op2ex := op2ex (2 , "¤" , "_xor" , op2ex)

op2ex := stack (op2ex , ["¬a" "_not(a)"])

Boolean Algebra Example

Convert an expression $expr := \text{str2num}("a\&(b|c\&a|1|a¤1)\&\neg(a\&b)\&a\&b¤\neg c|d\&0")$ $\_expr := op2ex(expr)$

$$expr = \Big(\Big(\Big(\big[a \wedge \big(\big((b \vee (c \wedge a)) \vee 1\big) \vee (a \oplus 1)\big)\big] \wedge \neg(a \wedge b)\Big) \wedge a\Big) \wedge (b \oplus \neg c)\Big) \vee (d \wedge 0)$$

$$\_expr = \_or\Big(\_or\Big(\_or\big(\_or\big(\_and(a, b), \_and(c, a)\big), 1\big), \_and\Big(\_and\Big(\_and\big(\_and\big(\_xor(a, 1), \_not(a)\big), b\big), a\Big), \_xor\big(b, \_not(c)\big)\Big)\Big), \_and(d, 0$$

Or recover the original expression with

$$\_and(a, b) := a \wedge b \qquad \_or(a, b) := a \vee b \qquad \_xor(a, b) := a \oplus b \qquad \_not(a) := \neg a$$

$$\_expr = \Big(\big(\big((a \wedge b) \vee (c \wedge a)\big) \vee 1\big) \vee \Big(\Big(\Big(\big((a \oplus 1) \wedge \neg a\big) \wedge b\Big) \wedge a\Big) \wedge (b \oplus \neg c)\Big)\Big) \vee (d \wedge 0)$$

... just for see that something is wrong somewhere

$$expr = \Big(\Big(\Big(\big[a \wedge \big(\big((b \vee (c \wedge a)) \vee 1\big) \vee (a \oplus 1)\big)\big] \wedge \neg(a \wedge b)\Big) \wedge a\Big) \wedge (b \oplus \neg c)\Big) \vee (d \wedge 0)$$

$$values := \begin{cases} a = 1 \\ b = 0 \\ c = 1 \\ d = 0 \end{cases} \qquad expr\Big|_{values} = 0 \qquad \_expr\Big|_{values} = 1$$

$\text{Clear}(\_and(a, b), \_or(a, b), \_xor(a, b), \_not(a)) = 1$ $\_exp$

$$op2ex(3 \cdot \cos(5 \cdot t + 2)) = \_times(3, \cos(\_plus(2, \_times(5, t))))$$

$$op2ex(3 \cdot \cos(5 \cdot t - 2)) = \_times(3, \cos(\_plus(\_times(2, -1), \_times(5, t))))$$

$$op2ex(-3 \cdot \cos(5 \cdot t - 2)) = \_times(-1, \_times(3, \cos(\_plus(\_times(2, -1), \_times(5, t)))))$$

Alvaro