Simplification Rules

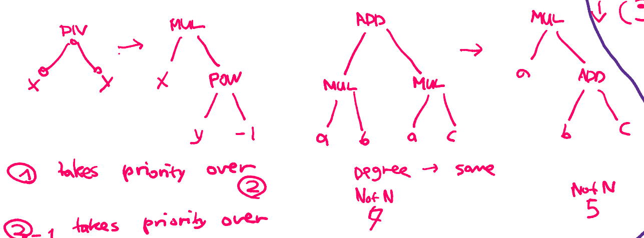
Tuesday, November 5, 2019 1:43 PM

Philosophy ① Minimize Syntactic Sugars
② Reduce Tree to have, Less Degree, less Nodes

Example

① $\frac{x}{y} \Rightarrow x \cdot y^{-1}$

③ $ab + ac \Rightarrow a(b+c)$



① takes priority over ②

②-1 takes priority over ②-2

Design We have 3 stages of simplification

(1) parser stage → (preprocessings)

(2) Operational stage (Top — down)

(3) Call stage

In order.

" We use Routine named Reduce() in call stage, and some other situation (View Equivalency).

[Table] [E

(Multiple signs) [— -

(Meaningless Parenthesis) [x(

[x+(

[(sin x

(Removing syntactic sugar

[x−y =

(Grouping)

[x · x · · · · (n times)

[x + x + · · · (n times)

(Ordering)

[yx = xy]

[y+x = x

Parser Stage

1 Do 'trivial' simplification on user-level mistakes, usually related to syntax rather than math.

2 Make expression prettier for the next stage

(To produce Normal form).

Operational Stage

We have 「operational matrix」 which can be described as a function

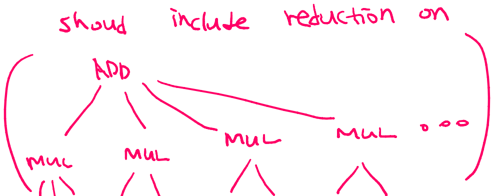
◎ (operator op_1, operator op_2)

the return value of ◎ is a simplifier function with respect to

op_1 (Parent operator) op_2 (Child operator)

For example, ◎(ADD, MUL)

should include reduction on



such as

$ab + ac$

$= a(b+c)$.

*Note that Child Trees are already simplified.

If there's Not in common, we just call Default simplifier for parent operator.

<ADD> (Identity)

$[x + 0 = 0 + x = x]$

(Inverse)

$[x + Neg(x) = 0]$

(Grouping)

$[x + x + \cdots + x = nx]$

<MUL> (Identity)

$x1 = 1x = x$

(ZD) $x0 = 0x = 0$

(Grouping)

$[x \cdot x \cdot \cdots x = x^n]$

<Pow> (Basics) $[x^1 = x,\ x^0 = 1,\ 1^x = 1,\ 0^y = 0 \ (y \neq 1)]$

(Exp law) [(a

Etc.. Specific ones

<MUL, pow>

$[(a^x)(a^y) = a^{x+y}$

$(b^x)(c^x) = (bc)^x] \ldots$

<ADD, MUL>

$ab + ac$

$= a(b+$

$[$<LN, MUL>

$\ln(ab) = \ln a + \ln b.]$

call- stage

steps are: (1) Simplify parameters. (Reduce())

(View simplifying logics)

(2) Calculate

(3) Reduce() to the result.

In call stage, we suppose that 「caller is already simplified」

Thus we simplify parameters.

And calculation (Just substitution.)

And again, Top-down simplificati

Simplifying logics Note that we have 2 ways to construct a function.

(1) $f(x) = \underline{x^2 + 3x + 4}$ (Direct Declaration) + operation

Parser parses this, to

(2) $g(x,y) = f(y) + \log(x)$ (when "call()" is CALLED)

xample]

$-x = x]$

$z) = xyz]$

$y+z) = x+y+z]$

$) = \sin x]$

$s)\left[\frac{x}{y} = xy^{-1}\right]$

$x + \text{Neg}\, y]$

$x = x^n]$

$+x = nx]$

$]$

$x+y]$

we have to recognize x as x^1 sometimes!!

$x)^y = a^{xy}]$

$+c)]$

n

(2) $g(x, y) = f(y) + \log(x)$ (when "call()" is CALLED) And again, Top-down

$\boxed{\text{Equivalency}}$ $f(x) = f(y)$ is 'false' while $f(x) \simeq f(y)$ is True.

Beta-equivalency '$\simeq$' is determined by (① change of order of parameters ② change of Name of parameters)

Ordinary 'equality' '$=$' is determined by ① Reduce() produces the same result (This includes order of operation ($xzy = xyz$.)).