Introduction

This notebook contains a few basic rules about the addition of natural numbers, based entirely on the following unwrapped membership rule:

\[
\text{member}[x, \text{NATADD}]
\]

\[
\text{and}[\text{member}[\text{first}[\text{first}[x]], V], \text{member}[\text{pair}[\text{first}[\text{first}[x]], \text{second}[x]], \text{iterate}[\text{SUCC}, \text{singleton}[\text{second}[\text{first}[x]]]])], \text{member}[\text{second}[x], \text{omega}]\]

Among the things we prove are the rule of exponents for the composite of two powers, and the commutativity of addition.

relation of NATADD to power[SUCC]

The connection between \text{NATADD} and \text{power}[\text{SUCC}] was already evident in the \text{RIF} rule for composites of powers derived 2002 June 6 in the notebook \text{POWERADD.NB}. On the basis of that formula we are led to study the following relation:

\[
\text{flip[rotate[inverse[power[SUCC]]]}}] // \text{VSTriNormality // Reverse}
\]

With this rule in place, we find:
NATADD // Normality // Reverse

\[ \text{composite[id[omega], rotate[\text{inverse[\text{power[SUCC]]}]]]} = \text{NATADD} \]

\[ \text{composite[id[omega], rotate[\text{inverse[\text{power[SUCC]]}]]]} := \text{NATADD} \]

■ composites with identities

\[ \text{Assoc[Id, id[omega], rotate[\text{inverse[\text{power[SUCC]]}]]]} \]

\[ \text{composite[Id, NATADD]} = \text{NATADD} \]

\[ \text{composite[Id, NATADD]} := \text{NATADD} \]

\[ \text{Assoc[id[omega], rotate[\text{inverse[\text{power[SUCC]]}]}, \text{id[cart[V, V]]}]/\text{Reverse} \]

\[ \text{composite[NATADD, id[cart[V, V]]]} = \text{NATADD} \]

\[ \text{composite[NATADD, id[cart[V, V]]]} := \text{NATADD} \]

■ image rules

\[ \text{SubstTest[image, composite[id[omega], rotate[\text{inverse[w]}]], cart[x, y], w -> power[SUCC]]} \]

\[ \text{image[NATADD, cart[x, y]]} = \text{intersection[omega, image[iterate[SUCC, y], x]]} \]

\[ \text{image[NATADD, cart[x, y]]} := \text{intersection[omega, image[iterate[SUCC, y], x]]} \]

In particular we have rules corresponding to \( 0 + x = x \) and \( x + 0 = x \):

\[ \text{image[NATADD, cart[\text{singleton[0]}, x]]} \]

\[ \text{intersection[omega, x]} \]

\[ \text{image[NATADD, cart[x, \text{singleton[0]]}]} \]

\[ \text{intersection[omega, x]} \]

■ range

\[ \text{SubstTest[range, composite[id[omega], rotate[\text{inverse[w]}]], w -> power[SUCC]]} \]

\[ \text{range[NATADD]} = \text{omega} \]

\[ \text{range[NATADD]} := \text{omega} \]
iterated inverses

We recall this formula connecting power and iterate.

\[
\text{composite[SECOND, id[cart[y, V]], power[x]]}
\]

\[
\text{iterate[x, y]}
\]

There is an analogous formula for inverse iterates:

\[
\text{SubstTest[composite, FIRST, id[cart[V, y]], composite[SWAP, power[z]], z \to \text{inverse[x]}]}
\]

\[
\text{composite[FIRST, id[cart[V, y]], power[x]]} = \text{iterate[\text{inverse[x]}, y]}
\]

\[
\text{composite[FIRST, id[cart[V, y_]], power[x_]]} := \text{iterate[\text{inverse[x]}, y]}
\]

In particular, we have the following fact, which is needed to compute the domain of NATADD.

\[
\text{composite[FIRST, id[cart[V, omega]], power[SUCC]]}
\]

\[
\text{cart[omega, omega]}
\]

domain

\[
\text{SubstTest[domain, composite[id[omega], rotate[\text{inverse[w]}]], w \to \text{power[SUCC]}]}
\]

\[
\text{domain[NATADD]} = \text{cart[omega, omega]}
\]

\[
\text{domain[NATADD]} := \text{cart[omega, omega]}
\]

As a corollary, we have:

\[
\text{subclass[NATADD, cart[cart[V, V], V]]}
\]

True

derivation of 1 + 1 = 2

We recall that 1 and 2 are

\[
1 = \text{succ[0]}
\]

\[
1 = \text{singleton[0]}
\]

\[
2 = \text{succ[succ[0]]}
\]

\[
2 = \text{succ[singleton[0]]}
\]

The formulas we obtained all contain intersections with omega that can be removed:
equal[intersection[omega, singleton[succ[singleton[0]]]],
      singleton[succ[singleton[0]]]]

True

The corresponding rewrite rule is:

intersection[omega, singleton[succ[singleton[0]]]] := singleton[succ[singleton[0]]]

With this simplification, the basic fact 1+1=2 looks like this:

image[NATADD, cart[singleton[singleton[0]], singleton[singleton[0]]]]

singleton[succ[singleton[0]]]

- **law of exponents**

The factor of id[omega] requires special care:

Assoc[power[x], id[omega], rotate[inverse[power[SUCCE]]]] /* Reverse

composite[power[x], rotate[inverse[power[SUCCE]]]] == composite[power[x], NATADD]

composite[power[x_], rotate[inverse[power[SUCCE]]]] := composite[power[x], NATADD]

Now we rederive the variable-free form for the law of exponents. We have moved the SWAP to the right side of the equation

Map[composite[SWAP, #] &, 
   composite[SWAP, RIF, cross[power[x], power[x]]]] /* VSTriNormality]

composite[RIF, cross[power[x], power[x]]] == composite[SWAP, power[x], NATADD]

composite[RIF, cross[power[x_], power[x_]]] := composite[SWAP, power[x], NATADD]

- **Commutative law of addition**

The commutative law for addition for natural numbers now appears as a corollary of the law of exponents. Since powers commute, we have:

SubstTest[flip, composite[SWAP, RIF, cross[w, w]], w -> power[x]]

composite[power[x], NATADD, SWAP] == composite[power[x], NATADD]

In particular, we can set x equal to SUCC.

% /. x -> SUCC

composite[power[SUCC], NATADD, SWAP] == composite[power[SUCC], NATADD]

We now evaluate at 0 by forming the composite with inverse[LEFT[0]]:
This says that \( m + n = n + m \) without using variables. The elimination of variables is important, because this rule can be made into a rewrite rule, but the version with variables cannot because it would cause looping.

\[
\text{composite}[\text{NATADD}, \text{SWAP}] := \text{NATADD}
\]

**connection between NATADD and power[SUCC] revisited**

We can solve the formula obtained for NATADD to obtain for a formula for a certain restriction of power[SUCC]:

\[
\text{Map}[	ext{Inverse}, \text{composite[\text{id[cart[V, omega]], power[SUCC]]]} \// \text{inverse} \// \text{TripleRotate}]
\]

\[
\text{composite[\text{id[cart[V, omega]], power[SUCC]]]} = \text{composite[\text{SWAP, inverse[rotate[NATADD]]}]} 
\]

From this one obtains:

\[
\text{power[composite[\text{id[omega], SUCC]]}}
\]

\[
\text{union[cart[singleton[0], Id], composite[SWAP, inverse[rotate[NATADD]]}]]
\]

In particular, the law of exponents can be specialized to composite[id[omega],SWAP] to yield a formula for NATADD, but it needs cleaning up.