multiplicative law of exponents

Johan G. F. Belinfante
2002 June 28

```
<< goedel52.o77; << tools.m

:Package Title: goedel52.o76           2002 June 26 at 6:15 p.m.

It is now:  2002 Jun 28 at 10:43

Loading Simplification Rules

TOOLS.M                   Revised 2002 June 12

weightlimit = 40
```

■ Introduction

In this notebook, a basic form of the multiplicative law of exponents $(x^m)^n = x^{mn}$ is derived. The catch is that at the moment the function for multiplication has not yet been introduced, so a formula for products will be revealed in the process.

■ Preliminaries

We begin by recalling the formulas

```
composite[id[cart[V, singleton[x]]], inverse[FIRST]]

RIGHT[x]
```

and

```
composite[NATADD, RIGHT[x]]

composite[id[omega], iterate[SUCC, singleton[x]]]
```

These are easily generalized to classes that are not singletons:

```
ImageComp[id[cart[V, omega]], power[SUCC], x]

composite[NATADD, id[cart[V, x]], inverse[FIRST]] ==
 composite[id[omega], image[power[SUCC], x]]
```

This equation will be added as a temporary rewrite rule:

```
composite[NATADD, id[cart[V, x_]], inverse[FIRST]] :=
 composite[id[omega], image[power[SUCC], x]]
```

Next we recall the **RIF** version of the additive law of exponents:

```
composite[SWAP, RIF, cross[power[x], power[x]]]

composite[power[x], NATADD]
```

We need an inverse rotated variant of this rule:

```
Map[inverse, composite[inverse[power[x]], RIF, cross[power[x], SWAP]] // TripleRotate]

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

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

From this we derive another temporary rule:

```
ImageComp[cross[inverse[power[x]], SWAP],
  composite[inverse[RIF], power[x]], y] // Reverse

composite[cross[Id, image[power[x], y]], power[x]] ==
 composite[power[x], image[power[SUCC], y]]

composite[cross[Id, image[power[x_], y_]], power[x_]] :=
 composite[power[x], image[power[SUCC], y]]
```

■ derivation

The whole derivation of the law of exponents takes just one step, using the uniqueness theorem for iteration:

```
SubstTest[implies, and[equal[composite[u, w], composite[w, SUCC]],
        equal[image[w, singleton[0]], v]],
    equal[composite[w, id[omega]], iterate[u, v]],
    {u -> cross[Id, image[power[x], y]], v -> Id,
        w -> composite[power[x], iterate[image[power[SUCC], y], singleton[0]]]}]

equal[composite[power[x], iterate[image[power[SUCC], y], singleton[0]]],
  power[image[power[x], y]]] == True
```

It is not immediately clear how this equation should be oriented if we want to add it as a rewrite rule. To help decide this issue, we first obtain a more familiar version of the law of exponents by mapping with **image**:

```
Map[image[#, z] &,
  Equal[composite[power[x], iterate[image[power[SUCC], y], singleton[0]]],
   power[image[power[x], y]]]] // Reverse

image[power[image[power[x], y]], z] ==
 image[power[x], image[iterate[image[power[SUCC], y], singleton[0]], z]]
```

The variables **y** and **z** here are like exponents, but we hasten to point out that they are not individual exponents, but rather collections of exponents. To get the common version of the law of exponents, one would need to replace these with singletons, say **y = singleton[m]** and **z = singleton[n]**. The usual expression $\mathbf{x}^{m}$ in the present formalism is written as the expression **image[power[x],singleton[m]]**. Since one would probably want $(\mathbf{x}^2)^3$ for example to be rewritten as $\mathbf{x}^6$, and not vice versa, it is suggestive to orient the equation for the law of exponents as follows:

```
image[power[image[power[x_], y_]], z_] :=
 image[power[x], image[iterate[image[power[SUCC], y], singleton[0]], z]]

image[power[image[power[x_], y_]], z_] =.
```

The variable **z** has been eliminated in the first version of the law of exponents. If this equation is oriented the same way, it would yield this rewrite rule:

```
power[image[power[x_], y_]] :=
 composite[power[x], iterate[image[power[SUCC], y], singleton[0]]]
```

■ A corollary.

Since **iterate** can be expressed in terms of **power**, we obtain this corollary:

```
SubstTest[composite, SECOND, id[cart[y, V]], power[w],
    {w -> image[power[SUCC], x], y -> singleton[0]}]

composite[id[omega], iterate[image[power[SUCC], x], singleton[0]]] ==
 iterate[image[power[SUCC], x], singleton[0]]

composite[id[omega], iterate[image[power[SUCC], x_], singleton[0]]] :=
 iterate[image[power[SUCC], x], singleton[0]]
```

■ Final remarks

Note that even in the most familiar case with single exponents, we are not entirely finished with the law of exponents, because we have yet to show that in this case the final expression obtained involves just a single exponent. That is, it remains to be shown that the class **image[iterate[image[power[SUCC],singleton[m]],singleton[0]],singleton[n]]** is a singleton, namely the singleton of the product of the numbers **m** and **n**. This will emerge later as a consequence of the fact that multiplication of natural numbers is a function.