binary homomorphisms and the distributive law

Johan G. F. Belinfante
2002 August 29

```
<< goedel52.p19; << tools.m
```

```
:Package Title: goedel52.p19        2002 August 29 at 3:40 p.m.

It is now:  2002 Aug 29 at 17:18

Loading Simplification Rules

TOOLS.M                   Revised 2002 August 22

weightlimit = 40
```

■ Summary

The distributive law is rewritten in various ways involving fewer variables. One of these formulations says that left−multiplication by **x** is a homomorphism of the binary operation **NATADD**. A homomorphism **h** of a binary operation **b** is a function that satisfies:

```
composite[h, b] == composite[b, cross[h, h]];
```

This notion was discussed in the article on composite and cross in the *Journal of Auutomated Reasoning*, volume 22 , pages 311−339(1999).

■ Preliminary maneuver

```
ImageComp[NATADD, id[cart[V, V]], singleton[x]]
```

```
image[NATADD, singleton[x]] == singleton[natadd[first[x], second[x]]]
```

```
image[NATADD, singleton[x_]] := singleton[natadd[first[x], second[x]]]
```

■ variable−freeversion of the distributive law

A variable−freeversion of the distributive law can be given using **TWIST** and **DUP**.

```
symdif[composite[NATADD, cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]],
composite[NATMUL, cross[NATADD, Id]]] // VSNormality
```

```
union[intersection[composite[NATMUL, cross[NATADD, Id]],
   composite[complement[NATADD], cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]]],
  intersection[composite[complement[NATMUL], cross[NATADD, Id]],
   composite[NATADD, cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]]]] == 0
```

```
union[intersection[composite[NATMUL, cross[NATADD, Id]],
   composite[complement[NATADD], cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]]],
  intersection[composite[complement[NATMUL], cross[NATADD, Id]],
   composite[NATADD, cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]]]] := 0

SubstTest[equal, 0, symdif[u, v],
    {u -> composite[NATADD, cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]],
  v -> composite[NATMUL, cross[NATADD, Id]]}]

True == equal[composite[NATMUL, cross[NATADD, Id]],
  composite[NATADD, cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]]]
```

This is the variable−freeversion of the distributive law:

```
composite[NATADD, cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]] :=
 composite[NATMUL, cross[NATADD, Id]]
```

■ a twisted tale

To derive the interpretaton of the distributive law in terms of binary homomorphisms, we need a new fact about **TWIST**. To do this in complete generality takes a bit of care. We begin with two temporary rules; first:

```
SubstTest[composite, cross[RIGHT[x], RIGHT[y]], cross[u, v],
 {u -> id[image[V, singleton[x]]], v -> id[image[V, singleton[y]]]}]

cross[RIGHT[x], composite[RIGHT[y], id[image[V, singleton[x]]]]] ==
 cross[RIGHT[x], RIGHT[y]]

cross[RIGHT[x_], composite[RIGHT[y_], id[image[V, singleton[x_]]]]] :=
 cross[RIGHT[x], RIGHT[y]]
```

This is the second:

```
Assoc[cross[RIGHT[x], id[image[V, singleton[x]]]],
  id[cart[V, cart[V, singleton[y]]]], cross[Id, inverse[FIRST]]] // Reverse

cross[RIGHT[x],
  composite[id[cart[V, intersection[image[V, singleton[x]], singleton[y]]]],
   inverse[FIRST]]] == cross[RIGHT[x], RIGHT[y]]

cross[RIGHT[x_],
  composite[id[cart[V, intersection[image[V, singleton[x_]], singleton[y_]]]],
   inverse[FIRST]]] := cross[RIGHT[x], RIGHT[y]]
```

From these we derive a permanent rule:

```
composite[TWIST, RIGHT[PAIR[x, y]]] // Normality

composite[TWIST, RIGHT[PAIR[x, y]]] == cross[RIGHT[x], RIGHT[y]]

composite[TWIST, RIGHT[PAIR[x_, y_]]] := cross[RIGHT[x], RIGHT[y]]
```

■ binary homomorphism interpretation

The distributive law implies that **composite[NATMUL,LEFT[x]]** is a homomorphism of **NATADD**.

```
Assoc[NATADD,
  composite[cross[NATMUL, NATMUL], TWIST, cross[Id, DUP]], RIGHT[x]] // Reverse
```

```
composite[NATMUL, LEFT[x], NATADD] ==
 composite[NATADD, cross[composite[NATMUL, LEFT[x]], composite[NATMUL, LEFT[x]]]]
```

```
composite[NATMUL, LEFT[x_], NATADD] :=
 composite[NATADD, cross[composite[NATMUL, LEFT[x]], composite[NATMUL, LEFT[x]]]]
```

■ Comment

The temporary formula derived in the notebook **MULASSOC.NB** is subsumed by this result.

```
composite[NATMUL, LEFT[x], NATADD, RIGHT[y]]
```

```
composite[NATADD, RIGHT[natmul[x, y]], NATMUL, LEFT[x]]
```