recursive subtraction

Johan G. F. Belinfante
2002 September 17

```
<< goedel52.p54; << tools.m
```

```
:Package Title: goedel52.p54          2002 September 17 at 12:35 noon

It is now:  2002 Sep 17 at 15:2

Loading Simplification Rules

TOOLS.M                  Revised 2002 September 16

weightlimit = 40
```

■ summary

Quaife’s theorem DF6 about recursive subtraction is derived in this notebook in two different ways, as well as a variable−free version thereof.

```
reference :  Art Quaife,
Automated Development of Fundamental Mathematical Theories, page 186.
```

■ Quaife’s Theorem DF6

The quickest derivation of Quaife’s theorem uses **IminComp**:

```
Map[A, IminComp[composite[NATADD, RIGHT[y]],
   composite[NATADD, RIGHT[z]], singleton[x]]] // Reverse
```

```
natsub[natsub[x, y], z] == natsub[x, natadd[y, z]]
```

We hold off adding this rule for now. Another derivation of this result will be presented shortly.

■ variable−freeversion

One could use Quaife’s theorem DF6 to derive versions with fewer variables , but it is more instructive to do this independently, starting with the associative law:

```
Assoc[NATADD, composite[cross[Id, NATADD], ASSOC], cross[RIGHT[x], Id]]
```

```
composite[NATADD, cross[Id, composite[NATADD, RIGHT[x]]]] ==
 composite[NATADD, cross[composite[NATADD, RIGHT[x]], Id]]
```

```
composite[NATADD, cross[Id, composite[NATADD, RIGHT[x_]]]] :=
 composite[NATADD, cross[composite[NATADD, RIGHT[x]], Id]]
```

```
SubstTest[rotate, composite[u, cross[v, w]],
    {u -> NATADD, v -> Id, w -> composite[NATADD, RIGHT[x]]}]
```

```
composite[rotate[NATADD], cross[Id, composite[NATADD, RIGHT[x]]]] ==
 composite[inverse[RIGHT[x]], inverse[NATADD], rotate[NATADD]]
```

It is unclear how to orient this equation. Mapping with **LEFT[y]** yields:

```
Map[composite[#, LEFT[y]] &, %]
```

```
composite[image[inverse[NATADD], singleton[y]], NATADD, RIGHT[x]] ==
 composite[inverse[RIGHT[x]], inverse[NATADD], image[inverse[NATADD], singleton[y]]]
```

This relation is its own inverse. This can be made into a rewrite rule:

```
composite[inverse[RIGHT[x_]], inverse[NATADD], image[inverse[NATADD], singleton[y_]]] :=
 composite[image[inverse[NATADD], singleton[y]], NATADD, RIGHT[x]]
```

From this we obtain Quaife's theorem:

```
Map[A, ImageComp[composite[inverse[RIGHT[x]], inverse[NATADD]],
        image[inverse[NATADD], singleton[y]], singleton[z]]] // Reverse
```

```
natsub[natsub[y, z], x] == natsub[y, natadd[x, z]]
```

The rule is added now:

```
natsub[natsub[y_, z_], x_] := natsub[y, natadd[x, z]]
```

■ variable−freeversion

One can derive a variable−freeversion by using **symdif** and **VSNormality**:

```
symdif[composite[rotate[NATADD], cross[rotate[NATADD], Id]],
    composite[rotate[NATADD], cross[Id, NATADD], ASSOC]] // VSNormality
```

```
union[intersection[composite[complement[rotate[NATADD]], cross[rotate[NATADD], Id]],
   composite[rotate[NATADD], cross[Id, NATADD], ASSOC]],
  intersection[composite[rotate[NATADD], cross[rotate[NATADD], Id]],
   composite[complement[rotate[NATADD]], cross[Id, NATADD], ASSOC]]] == 0
```

We add this as a temporary rule:

```
union[intersection[composite[complement[rotate[NATADD]], cross[rotate[NATADD], Id]],
   composite[rotate[NATADD], cross[Id, NATADD], ASSOC]],
  intersection[composite[rotate[NATADD], cross[rotate[NATADD], Id]],
   composite[complement[rotate[NATADD]], cross[Id, NATADD], ASSOC]]] := 0
```

The rule we really want is this:

```
SubstTest[equal, 0, symdif[u, v],
    {u -> composite[rotate[NATADD], cross[rotate[NATADD], Id]],
     v -> composite[rotate[NATADD], cross[Id, NATADD], ASSOC]}]
```

```
True == equal[composite[rotate[NATADD], cross[rotate[NATADD], Id]],
  composite[rotate[NATADD], cross[Id, NATADD], ASSOC]]
```

Note the close resemblance with the associative law for **NATADD**.

```
composite[rotate[NATADD], cross[Id, NATADD], ASSOC] :=
 composite[rotate[NATADD], cross[rotate[NATADD], Id]]
```

The rule we held off adding in the preceding section can now be rederived:

```
Assoc[rotate[NATADD], composite[cross[Id, NATADD], ASSOC], RIGHT[x]]

composite[rotate[NATADD], cross[Id, composite[NATADD, RIGHT[x]]]] ==
 composite[inverse[RIGHT[x]], inverse[NATADD], rotate[NATADD]]
```

This helps establish the connection between the variable−freeformula and Quaife’s formula DF6.