inequalities for addition

Johan G. F. Belinfante
2002 September 1

```
<< goedel52.p26; << tools.m
```

```
:Package Title: goedel52.p26        2002 August 31 at 9:10 p.m.

It is now:  2002 Sep 1 at 12:5

Loading Simplification Rules

TOOLS.M                  Revised 2002 August 30

weightlimit = 40
```

■ Summary

This notebook contains a derivation of some inequalities involving addition of natural numbers.

■ a simple case

Perhaps the simplest inequality is this:

```
SubstTest[subclass, composite[u, id[v], w], composite[u, w],
    {u -> NATADD, v -> cart[V, singleton[x]], w -> inverse[FIRST]}]
```

```
subclass[composite[NATADD, RIGHT[x]], S] == True
```

```
subclass[composite[NATADD, RIGHT[x_]], S] := True
```

A more familiar formulation of this result can be provided by introducing a second variable:

```
SubstTest[implies, subclass[u, v],
 subclass[image[u, singleton[x]], image[v, singleton[x]]],
    {u -> composite[NATADD, RIGHT[y]], v -> S}]
```

```
subclass[x, natadd[x, y]] == True
```

Note that one does not need to explicitly assume that **x** and **y** are natural numbers. If either one fails to be a natural number, then this result reduces to the true statement: **subclass[x,V]**.

■ monotonicity

A relational formulation of monotonicity is easily derived:

```
Assoc[composite[NATADD, RIGHT[x]], NATADD, inverse[SECOND]]

composite[NATADD, RIGHT[x], S, id[omega]] == composite[id[omega], S, NATADD, RIGHT[x]]

composite[NATADD, RIGHT[x_], S, id[omega]] := composite[id[omega], S, NATADD, RIGHT[x]]
```

This says that the restriction of the subclass relation **S** to the set **omega** of natural numbers commutes with right−addition:

```
commute[composite[id[omega], S, id[omega]], composite[NATADD, RIGHT[x]]]

True
```

■ a more familiar formulation of monotonicity involving three variables

A more familiar formulation of monotonicty can be derived by introducing two extra variables. First we add one new variable:

```
SubstTest[subclass, composite[u, id[v], w], composite[u, w],
    {u -> composite[NATADD, RIGHT[x]], v -> singleton[y], w -> composite[S, id[omega]]}]

subclass[cart[intersection[omega, P[y]], singleton[natadd[x, y]]],
  composite[S, NATADD, RIGHT[x]]] == True

subclass[cart[intersection[omega, P[y_]], singleton[natadd[x_, y_]]],
  composite[S, NATADD, RIGHT[x_]]] := True
```

Adding a second variable yields a familiar result:

```
SubstTest[implies, subclass[u, v],
 subclass[image[u, singleton[x]], image[v, singleton[x]]],
    {u -> cart[intersection[omega, P[y]], singleton[natadd[z, y]]],
        v -> composite[S, NATADD, RIGHT[z]]}]

or[not[member[x, omega]], not[subclass[x, y]],
  subclass[natadd[x, z], natadd[y, z]]] == True
```

■ a counterexample

The familiar formulation of monotonicity involves three variables, but only one of the three variables needs to be explicitly restricted to natural numbers. If either **y** or **z** fails to be a natural number then the conclusion reduces to the true statement that **natadd[x,z]** is a subclass of the universal class **V**. One cannot omit the hypothesis that **x** be a natural number, as the following counterexample shows:

```
(implies[subclass[x, y], subclass[natadd[x, z], natadd[y, z]]] /.
    {x -> singleton[singleton[0]], y -> succ[singleton[0]], z -> 0})

subclass[V, succ[singleton[0]]]
```

This contradicts the fact that a subclass of a set must be a set:

```
Map[not, SubstTest[implies, and[subclass[u, v], member[v, V]], member[u, V],
    {u -> V, v -> succ[singleton[0]]}]]

subclass[V, succ[singleton[0]]] == False
```