Quaife's theorem (A4)

Johan G. F. Belinfante and Claudia Huang
2005 June 2

```
In[1]:= SetDirectory["i:"]; << goedel70.03a; << tools.m

        :Package Title: goedel70.03a       2005 June 3 at 11:45 a.m.

        It is now:  2005 Jun 3 at 14:31

        Loading Simplification Rules

        TOOLS.M                       Revised 2005 May 17

        weightlimit = 40
```

---

summary

The **GOEDEL** program currently recognizes all the theorems in Quaife's **A** group about addition of natural numbers, except Theorem **(A4)**. In this notebook, a new rewrite rule is derived to close this gap. (The theorem in question appears on page 185 of Quaife's book.)

```
In[2]:= "Art Quaife, Automated Development of Fundamental Mathematical Theories,
           Kluwer Academic Publishers, Dordrecht, the Netherlands, 1992.";
```

Quaife's theorem **(A4)** says one can cancel **z** in an equation about natural numbers of the form **x + z = y + z**. In the **GOEDEL** program, the arithmetic of natural numbers is being developed within the context of set theory, so variables are not automatically assumed to refer to natural numbers. One could of course simply add literals of the form **member[x, omega]** as hypotheses to limit the cancellation law to natural numbers, but this would have the disadvantage that the resulting rule would become a statement of fact rather than a rewrite rule that actually causes a cancellation to take place.

deriving a rule akin to Quaife's theorem (A4)

The expression **natadd[x, y]** in the **GOEDEL** program which replaces Quaife's **x + y** is equal to **V** when **x** or **y** is not a natural number. The following new rewrite rule does not require the variables to refer to natural numbers. As a result, this rewrite rule is necessarily more complicated than Quaife's result.

```
In[3]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> natadd[x, z], v -> natadd[y, z]}] // Reverse

Out[3]= equal[natadd[x, z], natadd[y, z]] ==
         or[and[not[member[x, omega]], not[member[y, omega]]],
          and[equal[x, y], member[x, omega], member[y, omega]], not[member[z, omega]]]

In[4]:= equal[natadd[x_, z_], natadd[y_, z_]] :=
         or[and[not[member[x, omega]], not[member[y, omega]]],
          and[equal[x, y], member[x, omega], member[y, omega]], not[member[z, omega]]]
```

nat wrappers

The **nat** wrapper provides a convenient way to restrict to the case of natural numbers. Its definition is:

```
In[5]:= intersection[x, image[V, intersection[omega, set[x]]]]

Out[5]= nat[x]
```

This wrapper satisfies:

```
In[6]:= member[nat[x], omega]

Out[6]= True

In[7]:= equal[x, nat[x]]

Out[7]= member[x, omega]
```

The following rewrite rule is new:

```
In[8]:= equal[V, nat[x]] // AssertTest

Out[8]= equal[V, nat[x]] == False

In[9]:= equal[V, nat[x_]] := False
```

When all variables are wrapped with **nat**, the new rewrite rule derived in the preceding section behaves just like Quaife's.

```
In[10]:= equal[natadd[nat[x], nat[z]], natadd[nat[y], nat[z]]]

Out[10]= equal[nat[x], nat[y]]
```