Quaife's corollary (LD3cor)

Johan G. F. Belinfante and Ming Li
2007 April 7

```
In[1]:= SetDirectory["l:"]; << goedel92.05b; << tools.m

        :Package Title: goedel92.05a        2007 April 5 at 10:30 p.m.

        It is now:  2007 Apr 7 at 6:3

        Loading Simplification Rules

        TOOLS.M                       Revised 2007 March 25

        weightlimit = 40
```

summary

The equality rewrite rules for **natsub** and **monus** in the **GOEDEL** program sometimes interfere with equality substitution. Such conflicts can be resolved by adding additional rewrite rules. A case in point is Quaife's corollary **(LD3cor)**, which is related to Quaife's clause **(LD4g)** by a simple equality substitution. Although the **GOEDEL** program already has a rewrite rule analogous to **(LD4g)**, this does not suffice for it to recognize the truth of corollary **(LD3cor)**. The reason is that the **GOEDEL** program first rewrites the equation in **(LD3cor)**, thereby blocking application of the equality substitution rule. To remedy this, an additional rewrite rule is derived in this notebook.

```
In[2]:= "Art Quaife, Automated Development of Fundamental Mathematical
           Theories, Appendix 3. Theorems Proved in Peano's Arithmetic,
           Kluwer Academic Publishers, Dordrecht, 1992. Cf. pp. 200–201";
```

derivation

The **GOEDEL** program contains the following rewrite rule analogous to Quaife's clause **(LD4g)**.

```
In[3]:= member[monus[natmul[u, x], natmul[v, y]], ld[x, y]]

Out[3]= and[member[x, omega], member[y, omega]]
```

The **GOEDEL** also has the following rewrite rule for equations involving **monus**.

```
In[4]:= equal[z, monus[u, v]]

Out[4]= or[and[equal[0, z], not[member[u, omega]]], and[equal[0, z], not[member[v, u]]],
         and[equal[u, natadd[v, z]], member[v, omega], member[z, omega], not[equal[0, z]]]]
```

The following lemma helps clean up the result obtained when the above facts are combined.

```
In[6]:= equiv[or[equal[0, z], member[z, ld[nat[x], nat[y]]]], member[z, ld[nat[x], nat[y]]]]

Out[6]= True

In[7]:= or[equal[0, z_], member[z_, ld[nat[x_], nat[y_]]]] := member[z, ld[nat[x], nat[y]]]
```

Theorem. Analog of Quaife's **(LD3cor)**. To avoid five numberhood literals, all variables are wrapped with **nat**.

```
In[8]:= SubstTest[implies,
          and[equal[t, w], member[t, ld[nat[x], nat[y]]], member[w, ld[nat[x], nat[y]]],
          {t → monus[natmul[nat[u], nat[x]], natmul[nat[v], nat[y]]], w → nat[z]}] //
         Reverse // MapNotNot

Out[8]= or[member[nat[z], ld[nat[x], nat[y]]],
         not[equal[natadd[nat[z], natmul[nat[v], nat[y]]], natmul[nat[u], nat[x]]]]] == True

In[9]:= or[member[nat[z_], ld[nat[x_], nat[y_]]], not[
          equal[natadd[nat[z_], natmul[nat[v_], nat[y_]]], natmul[nat[u_], nat[x_]]]]] := True
```