Quaife's Theorem (Q19)

Johan G. F. Belinfante
2007 March 25

```
In[1]:= SetDirectory["l:"]; << goedel91.25a; << tools.m

        :Package Title: goedel91.25a        2007 March 25 at 10:10 a.m.

        It is now:  2007 Mar 25 at 22:0

        Loading Simplification Rules

        TOOLS.M                      Revised 2007 March 25

        weightlimit = 40
```

summary

In this notebook Quaife's Theorem **(Q19)** is derived.

```
In[2]:= "Art Quaife, Automated Development of Fundamental
           Mathematical Theories, Appendix 3. Theorems Proved in Peano's
           Arithmetic, Kluwer Academic Publishers, Dordrecht, 1992. Cf. p. 196";
```

derivation

Lemma 1.

```
In[3]:= (SubstTest[implies,
             and[equal[nat[x], natadd[nat[y], nat[u]]], equal[nat[u], natmul[nat[z], nat[t]]]],
             equal[nat[x], natadd[nat[y], natmul[nat[z],
                nat[natsub[natdiv[natsub[nat[x], natmod[nat[x], nat[z]]], nat[z]],
                  natdiv[natsub[nat[y], natmod[nat[y], nat[z]]], nat[z]]]]]]],
             t → natquot[nat[u], nat[z]]] // MapNotNot // Reverse) /. u → monus[nat[x], nat[y]]

Out[3]= or[equal[nat[x], natadd[nat[y],
            natmul[nat[z], nat[natsub[natdiv[natsub[nat[x], natmod[nat[x], nat[z]]], nat[z]],
               natdiv[natsub[nat[y], natmod[nat[y], nat[z]]], nat[z]]]]]]], member[nat[x],
           nat[y]], not[equal[0, natmod[nat[natsub[nat[x], nat[y]]], nat[z]]]]] == True

In[4]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Lemma 2.

```
In[5]:= Map[not, SubstTest[and, implies[and[p1, p2], p3], implies[p1, p4],
           implies[p4, p5], implies[and[p3, p5], p6], not[implies[and[p1, p2], p6]],
           {p1 -> equal[natmod[nat[x], nat[z]], natmod[nat[y], nat[z]]], p2 -> member[
              natadd[nat[y], natmod[nat[x], nat[z]]], natadd[nat[x], natmod[nat[y], nat[z]]]],
            p3 → not[member[nat[x], nat[y]]], p4 → member[
              pair[nat[z], monus[nat[x], nat[y]]], DIV],
            p5 -> equal[0, natmod[nat[natsub[nat[x], nat[y]]], nat[z]]],
            p6 → equal[nat[x], natadd[nat[y], natmul[nat[z],
                nat[natsub[natdiv[natsub[nat[x], natmod[nat[x], nat[z]]], nat[z]],
                  natdiv[natsub[nat[y], natmod[nat[y], nat[z]]], nat[z]]]]]]]}]] // Reverse

Out[5]= or[equal[nat[x], natadd[nat[y],
            natmul[nat[z], nat[natsub[natdiv[natsub[nat[x], natmod[nat[x], nat[z]]], nat[z]],
               natdiv[natsub[nat[y], natmod[nat[y], nat[z]]], nat[z]]]]]],
          not[equal[natmod[nat[x], nat[z]], natmod[nat[y], nat[z]]]],
          not[member[natadd[nat[y], natmod[nat[x], nat[z]]],
            natadd[nat[x], natmod[nat[y], nat[z]]]]] == True

In[6]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Theorem.

```
In[7]:= equiv[equal[nat[x], natadd[nat[y],
             natmul[nat[natsub[natdiv[natsub[nat[x], natmod[nat[x], nat[z]]], nat[z]],
                natdiv[natsub[nat[y], natmod[nat[y], nat[z]]], nat[z]]]], nat[z]]],
           or[equal[nat[x], nat[y]], and[equal[natmod[nat[x], nat[z]], natmod[nat[y], nat[z]]],
             member[natadd[nat[y], natmod[nat[x], nat[z]]],
              natadd[nat[x], natmod[nat[y], nat[z]]]]]] // not // not

Out[7]= True

In[8]:= equal[nat[x_], natadd[nat[y_], natmul[nat[z_],
            nat[natsub[natdiv[natsub[nat[x_], natmod[nat[x_], nat[z_]]], nat[z_]],
              natdiv[natsub[nat[y_], natmod[nat[y_], nat[z_]]], nat[z_]]]]]] :=
         or[and[equal[natmod[nat[x], nat[z]], natmod[nat[y], nat[z]]],
           member[natadd[nat[y], natmod[nat[x], nat[z]]],
            natadd[nat[x], natmod[nat[y], nat[z]]]]], equal[nat[x], nat[y]]]
```

Quaife's theorem (Q19)

Quaife's Theorem **(Q19)** is now recognized by the **GOEDEL** program.

In[9]:=
```
example[q19, "implies[equal[natmod[nat[x],nat[z]],natmod[nat[y],nat[z]]],
  equal[natmul[monus[natquot[nat[x],nat[z]],natquot[nat[y],nat[
    z]]],nat[z]],monus[nat[x],nat[y]]]]//not//not", "lemma for (DV22)"]
```

```
In[q19]:= implies[equal[natmod[nat[x],nat[z]],natmod[nat[y],nat[z]]],  equal[natmul[monus[
   natquot[nat[x],nat[z]],natquot[nat[y],nat[z]]],nat[z]],monus[nat[x],nat[y]]]]//not//not

lemma for (DV22)


Out[q19]= True


Execution time = 1 Seconds
```