

Quaife's theorem (Q7)

Johan G. F. Belinfante
2007 March 21

```
In[1]:= SetDirectory["1:"]; << goedel91.18a; << tools.m

:Package Title: goedel91.18a      2007 March 18 at 8:00 p.m.

It is now: 2007 Mar 21 at 14:33

Loading Simplification Rules

TOOLS.M                          Revised 2007 March 3

weightlimit = 40
```

summary

Quaife's theorem (Q7) 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. p. 195";
```

derivation of Q7

Theorem 1.

```
In[3]:= Map[not, SubstTest[and, implies[p1, p3], not[and[p2, p3]], and[p1, p2],
        {p1 -> member[natadd[natmod[nat[x], nat[y]], natmul[nat[y], nat[z]]], nat[x]],
          p2 -> member[nat[x], natmul[nat[y], nat[z]]],
          p3 -> member[natmul[nat[y], nat[z]], nat[x]]}]] // Reverse

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

In[4]:= or[not[member[nat[x_], natmul[nat[y_], nat[z_]]]], not[
        member[natadd[natmod[nat[x_], nat[y_]], natmul[nat[y_], nat[z_]]], nat[x_]]]] := True
```

Theorem 2.

```

In[5]:= Map[implies[and[not[equal[0, nat[z]]],
  member[natadd[natmod[nat[x], nat[y]], natmul[nat[y], nat[z]]], nat[x]], #] &,
  SubstTest[member, natmod[t, nat[y]], t,
  t -> natsub[nat[x], natmul[nat[y], nat[z]]]] // MapNotNot

Out[5]= or[equal[0, nat[z]], not[member[nat[x], natadd[nat[y], natmul[nat[y], nat[z]]]]],
  not[member[natadd[natmod[nat[x], nat[y]], natmul[nat[y], nat[z]]], nat[x]]] == True

In[6]:= or[equal[0, nat[z_]],
  not[member[nat[x_], natadd[nat[y_], natmul[nat[y_], nat[z_]]]]], not[
  member[natadd[natmod[nat[x_], nat[y_]], natmul[nat[y_], nat[z_]]], nat[x_]]] := True

```

Corollary of trichotomy:

```

In[7]:= SubstTest[or, member[nat[x], nat[t]], equal[nat[x], nat[t]],
  t -> natadd[natmod[nat[x], nat[z]], natmul[nat[y], nat[z]]] // Reverse

Out[7]= or[equal[nat[x], natadd[natmod[nat[x], nat[z]], natmul[nat[y], nat[z]]]],
  member[nat[x], natmul[nat[y], nat[z]]] ==
  not[member[natadd[natmod[nat[x], nat[z]], natmul[nat[y], nat[z]]], nat[x]]]

In[8]:= or[equal[nat[x_], natadd[natmod[nat[x_], nat[z_]], natmul[nat[y_], nat[z_]]]],
  member[nat[x_], natmul[nat[y_], nat[z_]]] :=
  not[member[natadd[natmod[nat[x], nat[z]], natmul[nat[y], nat[z]]], nat[x]]]

```

floored division

Quaife's notation x/y is here interpreted as floored division:

```

In[9]:= natquot[x_, y_] := natdiv[natsub[x, natmod[x, y]], y]

```

(Q7) alternative definition of quotient

The rewrite rules derived in this notebook allow the **GOEDEL** program to recognize Quaife's theorem **(Q7)** with a double negation, and all variables wrapped with **nat**:

```
In[10]:= example[q7, "implies[member[nat[x],natmul[nat[y],natadd[set[0],nat[z]]]],  
or[member[nat[x],natmul[nat[y],nat[z]]],  
equal[natquot[nat[x],nat[y]],nat[z]]]]//not//not", ""]
```

```
In[q7]:= implies[member[nat[x],natmul[nat[y],natadd[set[0],nat[z]]]], or[member[  
nat[x],natmul[nat[y],nat[z]]],equal[natquot[nat[x],nat[y]],nat[z]]]]//not//not
```

```
Out[q7]= True
```

```
Execution time = 0 Seconds
```