natdiv[x,y] and Quaife's theorem (Q1)

Johan G. F. Belinfante and Ming Li
2007 March 4

```
In[1]:= SetDirectory["l:"]; << goedel91.03a; << tools.m

        :Package Title: goedel91.03a        2007 March 3 at 12:15 midnite

        It is now:  2007 Mar 4 at 11:36

        Loading Simplification Rules

        TOOLS.M                    Revised 2007 March 3

        weightlimit = 40
```

summary

Some properties of **natdiv** are derived.

Quaife's theorem (Q1)

The result in this section was inspired by Quaife's theorem **(Q1)**.

```
In[2]:= "Art Quaife, Automated Development of Fundamental Mathematical
           Theories, Appendix 3. Thoerems Proved in Peano's Arithmetic, Kluwer
           Academic Publishers, Dordrecht, 1992. Cf. p. 194, Theorem (Q1).";
```

Quaife's notation **x/y** stands for floored division, which can be translated in the **GOEDEL** program as **natdiv[natsub[x, natmod[x, y]], y]**. The first clause of Quaife's theorem **(Q1)** is a consequence of the following fact:

```
In[3]:= equal[natsub[x, natmod[x, y]], natmul[y, natdiv[natsub[x, natmod[x, y]], y]]]

Out[3]= True

In[4]:= natmul[y_, natdiv[natsub[x_, natmod[x_, y_]], y_]] := natsub[x, natmod[x, y]]
```

natdiv sethood rules

Theorem 1.

```
In[5]:= equal[image[V, set[natdiv[x, y]]], image[V, intersection[image[DIV, set[y]], set[x]]]]

Out[5]= True

In[6]:= image[V, set[natdiv[x_, y_]]] := image[V, intersection[image[DIV, set[y]], set[x]]]
```

Theorem 2.

```
In[7]:= equal[union[complement[image[V, intersection[image[DIV, set[y]], set[x]]]],
          natdiv[x, y]], natdiv[x, y]]

Out[7]= True

In[8]:= union[complement[image[V, intersection[image[DIV, set[y_]], set[x_]]]],
          natdiv[x_, y_]] := natdiv[x, y]
```

natdiv numberhood rules

Theorem

```
In[9]:= equal[image[V, intersection[omega, set[natdiv[x, y]]]],
         image[V, intersection[image[DIV, set[y]], set[x]]]]

Out[9]= True

In[10]:= image[V, intersection[omega, set[natdiv[x_, y_]]]] :=
          image[V, intersection[image[DIV, set[y]], set[x]]]
```

Theorem.

```
In[11]:= equal[
          union[complement[image[V, intersection[omega, set[y]]]], natdiv[x, y]], natdiv[x, y]]

Out[11]= True

In[12]:= union[complement[image[V, intersection[omega, set[y_]]]], natdiv[x_, y_]] :=
          natdiv[x, y]
```

Theorem.

```
In[13]:= equal[
          union[complement[image[V, intersection[omega, set[x]]]], natdiv[x, y]], natdiv[x, y]]

Out[13]= True

In[14]:= union[complement[image[V, intersection[omega, set[x_]]]], natdiv[x_, y_]] :=
          natdiv[x, y]
```

Theorem.

```
In[15]:= equal[union[complement[image[V, intersection[omega, set[x]]]],
           intersection[image[V, y], natdiv[natsub[x, natmod[x, y]], y]]],
          natdiv[natsub[x, natmod[x, y]], y]]

Out[15]= True

In[16]:= union[complement[image[V, intersection[omega, set[x_]]]],
           intersection[image[V, y_], natdiv[natsub[x_, natmod[x_, y_]], y_]]] :=
          natdiv[natsub[x, natmod[x, y]], y]
```

natdiv and image[V, --]

Theorem.

```
In[17]:= natdiv[image[V, x], y] // Normality

Out[17]= natdiv[image[V, x], y] ==
          union[complement[image[V, intersection[omega, set[y]]]], image[V, x]]

In[18]:= natdiv[image[V, x_], y_] :=
          union[complement[image[V, intersection[omega, set[y]]]], image[V, x]]
```

Theorem.

```
In[19]:= natdiv[x, image[V, y]] // Normality

Out[19]= natdiv[x, image[V, y]] == union[image[V, x], image[V, y]]

In[20]:= natdiv[x_, image[V, y_]] := union[image[V, x], image[V, y]]
```

Theorem.

```
In[21]:= natdiv[union[x, image[V, y]], z] // Normality

Out[21]= natdiv[union[x, image[V, y]], z] == union[image[V, y], natdiv[x, z]]

In[22]:= natdiv[union[x_, image[V, y_]], z_] := union[image[V, y], natdiv[x, z]]
```

Corollary.

```
In[23]:= SubstTest[natdiv, union[x, image[V, t]], z, t -> complement[image[V, y]]] // Reverse

Out[23]= natdiv[union[x, complement[image[V, y]]], z] ==
          union[complement[image[V, y]], natdiv[x, z]]

In[24]:= natdiv[union[x_, complement[image[V, y_]]], z_] :=
          union[complement[image[V, y]], natdiv[x, z]]
```

Theorem.

```
In[25]:= natdiv[x, union[y, image[V, z]]] // Normality

Out[25]= natdiv[x, union[y, image[V, z]]] == union[image[V, z], natdiv[x, y]]

In[26]:= natdiv[x_, union[y_, image[V, z_]]] := union[image[V, z], natdiv[x, y]]
```

Corollary.

```
In[27]:= SubstTest[natdiv, x, union[y, image[V, t]], t → complement[image[V, z]]] // Reverse

Out[27]= natdiv[x, union[y, complement[image[V, z]]]] ==
          union[complement[image[V, z]], natdiv[x, y]]

In[28]:= natdiv[x_, union[y_, complement[image[V, z_]]]] :=
          union[complement[image[V, z]], natdiv[x, y]]
```

Lemma.

```
In[29]:= equal[intersection[complement[image[V, intersection[omega, set[z]]]], natdiv[x, z]],
          complement[image[V, intersection[omega, set[z]]]]]

Out[29]= True

In[30]:= intersection[complement[image[V, intersection[omega, set[z_]]]], natdiv[x_, z_]] :=
          complement[image[V, intersection[omega, set[z]]]]
```

Lemma.

```
In[31]:= equal[union[complement[image[V, intersection[omega, set[z]]]],
           intersection[image[V, x], image[V, y], natdiv[x, z]]],
          union[complement[image[V, intersection[omega, set[z]]]],
           intersection[image[V, y], natdiv[x, z]]]]

Out[31]= True

In[32]:= union[complement[image[V, intersection[omega, set[z_]]]],
           intersection[image[V, x_], image[V, y_], natdiv[x_, z_]]] :=
          union[complement[image[V, intersection[omega, set[z]]]],
           intersection[image[V, y], natdiv[x, z]]]
```

Theorem.

```
In[33]:= natdiv[intersection[x, image[V, y]], z] // Normality

Out[33]= natdiv[intersection[x, image[V, y]], z] ==
          union[complement[image[V, intersection[omega, set[z]]]],
           intersection[image[V, y], natdiv[x, z]]]

In[34]:= natdiv[intersection[x_, image[V, y_]], z_] :=
          union[complement[image[V, intersection[omega, set[z]]]],
           intersection[image[V, y], natdiv[x, z]]]
```

Corollary.

```
In[35]:= SubstTest[natdiv, intersection[x, image[V, t]],
           z, t → complement[image[V, y]]] // Reverse

Out[35]= natdiv[intersection[x, complement[image[V, y]]], z] ==
          union[complement[image[V, intersection[omega, set[z]]]],
           intersection[complement[image[V, y]], natdiv[x, z]]]

In[36]:= natdiv[intersection[x_, complement[image[V, y_]]], z_] :=
          union[complement[image[V, intersection[omega, set[z]]]],
           intersection[complement[image[V, y]], natdiv[x, z]]]
```