Quaife's corollary 3 of (EXDEF1)

Johan G. F. Belinfante and Ming Li
2006 January 24

```
In[1]:= SetDirectory["l:"]; << goedel89.23a; << tools.m

        :Package Title: goedel89.23a        2007 January 23 at 11:40 p.m.

        It is now:  2007 Jan 25 at 10:16

        Loading Simplification Rules

        TOOLS.M                      Revised 2007 January 7

        weightlimit = 40
```

summary

Quaife's third corollary of his definition **(EXDEF1)** is derived.

```
In[2]:= "Art Quaife, Automated Development of
           Fundamental Mathematical Theories, Kluwer Academic Publishers,
           Dordrecht, The Netherlands, 1992.  See Appendix 3 on page 228.";
```

derivation

Using **nat** wrappers facilitates deriving Corollary 3 for Quaife's **(EXDEF1)**.

```
In[3]:= SubstTest[implies, and[equal[nat[y], succ[nat[z]]],
           member[PAIR[nat[x], natexp[nat[x], succ[nat[z]]]], DIV]],
         member[PAIR[nat[x], natexp[nat[x], nat[y]]], DIV], z → U[nat[y]]] // Reverse

Out[3]= or[equal[0, nat[y]], member[pair[nat[x], natexp[nat[x], nat[y]]], DIV]] == True

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

removing the nat wrappers

Lemma.

```
In[5]:= SubstTest[implies, and[equal[x, nat[u]], equal[y, nat[v]]],
          or[equal[0, y], member[pair[x, natexp[x, y]], DIV]], {u → x, v → y}] // Reverse

Out[5]= or[equal[0, y], member[pair[x, natexp[x, y]], DIV],
          not[member[x, omega]], not[member[y, omega]]] == True

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

Lemma.

```
In[7]:= SubstTest[implies, member[pair[x, z], DIV],
           member[z, omega], z → natexp[x, y]] // MapNotNot // Reverse

Out[7]= or[member[y, omega], not[member[pair[x, natexp[x, y]], DIV]]] == True

In[8]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem.

```
In[9]:= equiv[member[pair[x, natexp[x, y]], DIV], or[and[equal[x, set[0]], member[y, omega]],
            and[member[x, omega], member[y, omega], not[equal[0, y]]]]] // not // not

Out[9]= True

In[10]:= member[pair[x_, natexp[x_, y_]], DIV] := or[and[equal[x, set[0]], member[y, omega]],
           and[member[x, omega], member[y, omega], not[equal[0, y]]]]
```

---

variable-free formulation

Theorem.

```
In[11]:= SubstTest[class, pair[x, y], member[pair[x, natexp[x, y]], z], z -> DIV] // InvertFix

Out[11]= fix[composite[inverse[NATEXP], DIV, FIRST]] ==
          union[cart[omega, intersection[omega, complement[set[0]]]], cart[set[set[0]], omega]]

In[12]:= fix[composite[inverse[NATEXP], DIV, FIRST]] :=
          union[cart[omega, intersection[omega, complement[set[0]]]], cart[set[set[0]], omega]]
```