reducing $x - y \cdot z$ modulo $z$

Johan G. F. Belinfante
2007 March 16

**summary**

A formula for reducing $x - y \cdot z$ modulo $z$ is derived, together with a corollary that enables the GOEDEL program to recognize the truth of the Corollary to Quaife's Theorem (Q6) and Quaife's Theorem (Q8).

**reducing $x - y \cdot z$ modulo $z$**

Lemma. (Temporary rewrite rule.)

In[3]:= equal[union[complement[image[V, intersection[omega, set[y]]]],
    natmod[natsub[x, natmul[y, z]], z]], natmod[natsub[x, natmul[y, z]], z]]


In[4]:= union[complement[image[V, intersection[omega, set[y]]]],
    natmod[natsub[x_, natmul[y_, z_]], z_]] := natmod[natsub[x, natmul[y, z]], z]

Theorem.

In[5]:= SubstTest[natmod, natadd[t, natmul[y, z]], z, t \rightarrow natsub[x, natmul[y, z]]]

Out[5]= natmod[natsub[x, natmul[y, z]], z] =
    union[complement[image[V, intersection[omega, set[y]]]],
    image[V, intersection[complement[x], natmul[y, z]]], natmod[x, z]]
main theorem

Using nat wrappers yields a result which can be used to derived the Corollary to Quaife's theorem (Q6).

Removing the nat wrappers yields a wrapper-free counterpart, but with redundant literals.

The following lemma suffices to eliminate the redundant literals:

Removing the redundant literals yields:

One final lemma is needed.
In[15]:= Map[or[member[x, omega], #] &, SubstTest[implies, and[member[x, V], equal[V, v]], member[x, v], v -> natadd[natmod[x, z], natmul[y, z]]]] // Reverse // MapNotNot

Out[15]= or[member[x, omega],
    member[x, natadd[natmod[x, z], natmul[y, z]]], not[member[x, V]]] = True

In[16]:= (% /. {x -> x_, y -> y_, z -> z_}) /. Equal -> SetDelayed

Main theorem.

In[17]:= equiv[member[x, natadd[natmod[x, z], natmul[y, z]]],
    or[and[member[x, V], not[member[x, omega]]], member[x, natmul[y, z]]]] // not // not

Out[17]= True

In[18]:= member[x_, natadd[natmod[x_, z_], natmul[y_, z_]]] :=
    or[and[member[x, V], not[member[x, omega]]], member[x, natmul[y, z]]]