co-restrictions of binary homomorphisms

Johan G. F. Belinfante
2013 July 7

```
In[1]:= SetDirectory["l:"]; << goedel.13jul06a

:Package Title: goedel.13jul06a                    2013 July 6 at 4:30 p.m.

Loading takes about seventeen minutes, half that time due to builtin pauses.

It is now:  2013 Jul 7 at 10:3

Loading Simplification Rules

TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

weightlimit = 40

Loading completed.

It is now:  2013 Jul 7 at 10:19
```

summary

If **t ∈ binhom[x, y]**, one can replace **y** with its restriction to the cartesian square of the range of **t**.

derivation

Theorem. A simplification rule for mappings.

```
In[2]:= SubstTest[member, t, intersection[u, v], {u -> map[x, y], v -> map[x, range[t]]}] // Reverse

Out[2]= member[t, map[x, intersection[y, range[t]]]] == member[t, map[x, y]]

In[3]:= member[t_, map[x_, intersection[y_, range[t_]]]] := member[t, map[x, y]]
```

An analogous simplification rule will be derived for binary homomorphisms.

Lemma. One can restrict **y** to the cartesian square of **range[t]**.

```
In[4]:= Map[implies[member[t, binhom[x, y]], #] &, (member[t, binhom[x, z]] // AssertTest) /.
          z → composite[y, id[cartsq[range[t]]]]] // MapNotNot

Out[4]= or[member[t, binhom[x, composite[y, id[cart[range[t], range[t]]]]]],
          not[member[t, binhom[x, y]]]] == True

In[5]:= (% /. {t → t_, x → x_, y → y_}) /. Equal → SetDelayed
```

Converse.

```
In[6]:=  Map[implies[#, member[t, binhom[x, y]]] &, (member[t, binhom[x, z]] // AssertTest) /.
            z → composite[y, id[cartsq[range[t]]]]] // MapNotNot

Out[6]=  or[member[t, binhom[x, y]],
           not[member[t, binhom[x, composite[y, id[cart[range[t], range[t]]]]]]]] == True

In[7]:=  (% /. {t → t_, x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. A simplification rule for binary homomorphisms.

```
In[8]:=  equiv[member[t, binhom[x, composite[y, id[cart[range[t], range[t]]]]]],
          member[t, binhom[x, y]]]

Out[8]=  True

In[9]:=  member[t_, binhom[x_, composite[y_, id[cart[range[t_], range[t_]]]]]] :=
          member[t, binhom[x, y]]
```

Any **t ∈ binhom[x, y]** can be factored as the product of itself with the inclusion binary homomorphism **id[range[t]]** from the restriction of **y** to the cartesian square of **range[t]** into **y** itself.

Theorem. An inclusion binary homomorphism associated with the range of a given binary homomorphism.

```
In[10]:=  Map[implies[member[t, binhom[x, y]], #] &,
            member[id[range[t]], binhom[composite[y, id[cartsq[range[t]]]], y]] //
             AssertTest] // MapNotNot

Out[10]=  or[member[id[range[t]], binhom[composite[y, id[cart[range[t], range[t]]]], y]],
            not[member[t, binhom[x, y]]]] == True

In[11]:=  or[member[id[range[t_]], binhom[composite[y_, id[cart[range[t_], range[t_]]]], y_]],
            not[member[t_, binhom[x_, y_]]]] := True
```