

restrictions of associative relations

Johan G. F. Belinfante
2003 August 21

```
In[1]:= << goedel52.s78; << tools.m

:Package Title: goedel52.s78      2003 August 14 at 4:45 p.m.

It is now: 2003 Aug 21 at 12:5

Loading Simplification Rules

TOOLS.M                          Revised 2003 August 9

weightlimit = 40
```

summary

It is shown in this notebook that if x is an associative relation, and if z is closed under x , that is, if $\text{image}[x, \text{cart}[z, z]]$ is a subclass of z , then the restriction $\text{composite}[x, \text{id}[\text{cart}[z, z]]]$ is associative.

derivation

Lemmas:

```
In[2]:= SubstTest[composite, ASSOC, cross[cross[x, x], x], x → id[z]]

Out[2]= composite[ASSOC, id[cart[cart[z, z], z]]] = composite[id[cart[z, cart[z, z]]], ASSOC]

In[3]:= composite[ASSOC, id[cart[cart[z_, z_], z_]]] := composite[id[cart[z, cart[z, z]]], ASSOC]

In[4]:= (associative[x] // AssertTest) /. x → composite[x, id[cart[z, z]]] // Reverse

Out[4]= equal[composite[x, cross[composite[id[z], x, id[cart[z, z]]], id[z]]],
  composite[x, cross[id[z], composite[id[z], x, id[cart[z, z]]]], ASSOC] ==
  associative[composite[x, id[cart[z, z]]]]

In[5]:= equal[composite[x_, cross[composite[id[z_], x_, id[cart[z_, z_]]], id[z_]]],
  composite[x_, cross[id[z_], composite[id[z_], x_, id[cart[z_, z_]]]], ASSOC] :=
  associative[composite[x, id[cart[z, z]]]]

In[6]:= SubstTest[implies, equal[u, v], equal[composite[u, w], composite[v, w]],
  {u → composite[x, cross[x, Id]],
  v → composite[x, cross[Id, x], ASSOC], w → id[cart[cart[z, z], z]]}

Out[6]= or[equal[composite[x, cross[composite[x, id[cart[z, z]]], id[z]]],
  composite[x, cross[id[z], composite[x, id[cart[z, z]]]], ASSOC],
  not[equal[composite[x, cross[x, Id]], composite[x, cross[Id, x], ASSOC]]] == True

In[7]:= (% /. {x → x_, z → z_}) /. Equal → SetDelayed
```

```

In[8]:= SubstTest[implies, equal[u, v], equal[cross[w, u], cross[w, v]],
  {u → composite[id[z], x, id[cart[z, z]]], v → composite[x, id[cart[z, z]]], w → id[z]}]

Out[8]= or[equal[cross[id[z], composite[x, id[cart[z, z]]]],
  cross[id[z], composite[id[z], x, id[cart[z, z]]]],
  not[subclass[image[x, cart[z, z]], z]] = True

In[9]:= (% /. {x → x_, z → z_}) /. Equal → SetDelayed

In[10]:= SubstTest[implies, equal[u, v], equal[cross[u, w], cross[v, w]],
  {u → composite[id[z], x, id[cart[z, z]]], v → composite[x, id[cart[z, z]]], w → id[z]}]

Out[10]= or[equal[cross[composite[x, id[cart[z, z]]], id[z]],
  cross[composite[id[z], x, id[cart[z, z]]], id[z]]],
  not[subclass[image[x, cart[z, z]], z]] = True

In[11]:= (% /. {x → x_, z → z_}) /. Equal → SetDelayed

In[12]:= SubstTest[implies, and[equal[u, v], equal[v, w]], equal[u, w],
  {u → composite[x, cross[composite[id[z], x, id[cart[z, z]]], id[z]]],
  v → composite[x, cross[composite[x, id[cart[z, z]]], id[z]]],
  w → composite[x, cross[id[z], composite[id[z], x, id[cart[z, z]]]], ASSOC}}]

Out[12]= or[associative[composite[x, id[cart[z, z]]]],
  not[equal[composite[x, cross[composite[x, id[cart[z, z]]], id[z]]],
  composite[x, cross[composite[id[z], x, id[cart[z, z]]], id[z]]]],
  not[equal[composite[x, cross[composite[x, id[cart[z, z]]], id[z]]],
  composite[x, cross[id[z], composite[id[z], x, id[cart[z, z]]]], ASSOC]]] = True

In[13]:= (% /. {x → x_, z → z_}) /. Equal → SetDelayed

Main result: (This takes a while.)

In[14]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], implies[p4, p5],
  implies[p5, p6], implies[p4, p7], implies[p7, p8], implies[and[p3, p6], p9],
  implies[and[p8, p9], p10], not[implies[and[p1, p4], p10]], {p1 → associative[x],
  p2 → equal[composite[x, cross[x, Id]], composite[x, cross[Id, x], ASSOC]],
  p3 → equal[composite[x, cross[composite[x, id[cart[z, z]]], id[z]]],
  composite[x, cross[id[z], composite[x, id[cart[z, z]]]], ASSOC]],
  p4 → subclass[image[x, cart[z, z]], z],
  p5 → equal[cross[id[z], composite[x, id[cart[z, z]]]],
  cross[id[z], composite[id[z], x, id[cart[z, z]]]],
  p6 → equal[composite[x, cross[id[z], composite[id[z], x, id[cart[z, z]]]], ASSOC],
  composite[x, cross[id[z], composite[x, id[cart[z, z]]]], ASSOC]],
  p7 → equal[cross[composite[id[z], x, id[cart[z, z]]], id[z]],
  cross[composite[x, id[cart[z, z]]], id[z]],
  p8 → equal[composite[x, cross[composite[id[z], x, id[cart[z, z]]], id[z]],
  composite[x, cross[composite[x, id[cart[z, z]]], id[z]]],
  p9 → equal[composite[x, cross[composite[x, id[cart[z, z]]], id[z]],
  composite[x, cross[id[z], composite[id[z], x, id[cart[z, z]]]], ASSOC]],
  p10 → associative[composite[x, id[cart[z, z]]]]]]

Out[14]= or[associative[composite[x, id[cart[z, z]]]],
  not[associative[x]], not[subclass[image[x, cart[z, z]], z]] = True

In[15]:= or[associative[composite[x_, id[cart[z_, z_]]]],
  not[associative[x_]], not[subclass[image[x_, cart[z_, z_]], z_]] := True

```