inv[cat[x]]

Johan G. F. Belinfante
2011 May 24

```
In[1]:= SetDirectory["l:"]; << goedel.11may23a
```

```
:Package Title: goedel.11may23a              2011 May 23 at 7:00 p.m.

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

It is now:  2011 May 24 at 8:7

Loading Simplification Rules

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

weightlimit = 40

Loading completed.

It is now:  2011 May 24 at 8:18
```

summary

Rewrite rules are derived for **inv[cat[x]]**. These rules apply in particular to monoids and groups. It is shown that if **x** is a submonoid of a group **y**, and if **range[x]** is invariant under **inv[y]**, then **x** is a group.

derivation

Theorem.

```
SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
  {t → inv[cat[x]], u → y, v → image[inv[cat[x]], y]}] // Reverse
```

```
or[not[subclass[y, image[inv[cat[x]], y]]], subclass[image[inv[cat[x]], y], y]] == True
```

```
or[not[subclass[y_, image[inv[cat[x_]], y_]]],
  subclass[image[inv[cat[x_]], y_], y_]] := True
```

Corollary.

```
In[30]:= SubstTest[implies, equal[x, cat[t]], or[
           not[subclass[y, image[inv[x], y]]], subclass[image[inv[x], y], y]], t → x] // Reverse
```

```
Out[30]= or[not[category[x]],
           not[subclass[y, image[inv[x], y]]], subclass[image[inv[x], y], y]] == True
```

```
In[31]:= or[not[category[x_]], not[subclass[y_, image[inv[x_], y_]]],
           subclass[image[inv[x_], y_], y_]] := True
```

Theorem.

```
In[33]:= Map[not, SubstTest[and, implies[p2, p3],
            not[implies[p1, p3]], {p1 → member[x, MONOIDS], p2 → category[x], p3 ->
              or[not[subclass[y, image[inv[x], y]]], subclass[image[inv[x], y], y]]}]] // Reverse

Out[33]= or[not[member[x, MONOIDS]],
           not[subclass[y, image[inv[x], y]]], subclass[image[inv[x], y], y]] == True

In[34]:= or[not[member[x_, MONOIDS]], not[subclass[y_, image[inv[x_], y_]]],
           subclass[image[inv[x_], y_], y_]] := True
```

Theorem.

```
In[36]:= SubstTest[implies, equal[x, cat[t]], equal[
            intersection[invar[inv[x]], P[domain[inv[x]]]], subvar[inv[x]]], t → x] // Reverse

Out[36]= or[equal[intersection[invar[inv[x]], P[domain[inv[x]]]], subvar[inv[x]]],
           not[category[x]]] == True

In[37]:= or[equal[intersection[invar[inv[x_]], P[domain[inv[x_]]]], subvar[inv[x_]]],
           not[category[x_]]] := True
```

Theorem.

```
In[44]:= SubstTest[or, not[subclass[y, domain[funpart[t]]]],
           not[subclass[image[funpart[t], y], y]],
           subclass[y, image[inverse[funpart[t]], y]], t → inv[cat[x]]] // Reverse

Out[44]= or[not[subclass[y, domain[inv[cat[x]]]]],
           not[subclass[image[inv[cat[x]], y], y]], subclass[y, image[inv[cat[x]], y]]] == True

In[46]:= or[not[subclass[image[inv[cat[x_]], y_], y_]],
           not[subclass[y_, domain[inv[cat[x_]]]]],
           subclass[y_, image[inv[cat[x_]], y_]]] := True
```

Corollary.

```
In[48]:= SubstTest[implies, equal[x, cat[t]], or[not[subclass[y, domain[inv[x]]]],
            not[subclass[image[inv[x], y], y]], subclass[y, image[inv[x], y]]], t → x] // Reverse

Out[48]= or[not[category[x]], not[subclass[y, domain[inv[x]]]],
           not[subclass[image[inv[x], y], y]], subclass[y, image[inv[x], y]]] == True

In[49]:= or[not[category[x_]], not[subclass[y_, domain[inv[x_]]]],
           not[subclass[image[inv[x_], y_], y_]], subclass[y_, image[inv[x_], y_]]] := True
```

Theorem.

```
In[50]:= Map[not, SubstTest[and, implies[p2, p3], not[implies[p1, p3]],
           {p1 → member[x, MONOIDS], p2 → category[x], p3 -> or[not[subclass[y, domain[inv[x]]]],
             not[subclass[image[inv[x], y], y]], subclass[y, image[inv[x], y]]]}]] // Reverse

Out[50]= or[not[member[x, MONOIDS]], not[subclass[y, domain[inv[x]]]],
           not[subclass[image[inv[x], y], y]], subclass[y, image[inv[x], y]]] == True

In[51]:= or[not[member[x_, MONOIDS]], not[subclass[y_, domain[inv[x_]]]],
           not[subclass[image[inv[x_], y_], y_]], subclass[y_, image[inv[x_], y_]]] := True
```

the case of a group

Lemma.

```
In[53]:= SubstTest[or, not[member[t, MONOIDS]], not[subclass[y, image[inv[t], y]]],
           subclass[image[inv[t], y], y], t → gp[x]] // Reverse

Out[53]= or[equal[0, gp[x]], not[subclass[y, image[inv[gp[x]], y]]],
           subclass[image[inv[gp[x]], y], y]] == True

In[54]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem.

```
In[56]:= SubstTest[implies, equal[x, gp[t]], or[equal[0, x], not[subclass[y, image[inv[x], y]]],
             subclass[image[inv[x], y], y]], t → x] // Reverse // MapNotNot

Out[56]= or[not[member[x, GROUPS]],
           not[subclass[y, image[inv[x], y]]], subclass[image[inv[x], y], y]] == True

In[58]:= or[not[member[x_, GROUPS]], not[subclass[y_, image[inv[x_], y_]]],
           subclass[image[inv[x_], y_], y_]] := True
```

Lemma.

```
In[60]:= SubstTest[or, not[member[t, MONOIDS]],
           not[subclass[y, domain[inv[t]]]], not[subclass[image[inv[t], y], y]],
           subclass[y, image[inv[t], y]], t → gp[x]] // Reverse

Out[60]= or[equal[0, gp[x]], not[subclass[y, range[gp[x]]]],
           not[subclass[image[inv[gp[x]], y], y]], subclass[y, image[inv[gp[x]], y]]] == True

In[61]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem.

```
In[63]:= SubstTest[implies, equal[x, gp[t]],
           or[equal[0, x], not[subclass[y, range[x]]], not[subclass[image[inv[x], y], y]],
            subclass[y, image[inv[x], y]]], t → x] // Reverse // MapNotNot

Out[63]= or[not[member[x, GROUPS]], not[subclass[y, range[x]]],
           not[subclass[image[inv[x], y], y]], subclass[y, image[inv[x], y]]] == True

In[65]:= or[not[member[x_, GROUPS]], not[subclass[image[inv[x_], y_], y_]],
           not[subclass[y_, range[x_]]], subclass[y_, image[inv[x_], y_]]] := True
```

---

subgroups

It was shown in the notebook **SBMD-INV.NB** that if **x** is a submonoid of a group **y**, and if **range[x]** is subvariant under **inv[y]**, then **x** is a group. Here a similar result is derived, but replacing subvariance with invariance.

Theorem. A submonoid **x** of a group **y** is a subgroup if **range[x]** is invariant under **inv[y]**.

```
In[67]:= Map[not, SubstTest[and, implies[p3, p5], implies[and[p2, p4, p5], p6],
            implies[and[p1, p2, p3, p6], p7], not[implies[and[p1, p2, p3, p4], p7]],
            {p1 → member[x, MONOIDS], p2 → member[y, GROUPS], p3 → subclass[x, y],
             p4 → invariant[inv[y], range[x]], p5 → subclass[range[x], range[y]],
             p6 → subvariant[inv[y], range[x]], p7 → member[x, GROUPS]}]] // Reverse

Out[67]= or[member[x, GROUPS], not[member[x, MONOIDS]], not[member[y, GROUPS]],
           not[subclass[x, y]], not[subclass[image[inv[y], range[x]], range[x]]]] == True

In[68]:= or[member[x_, GROUPS], not[member[x_, MONOIDS]], not[member[y_, GROUPS]],
           not[subclass[x_, y_]], not[subclass[image[inv[y_], range[x_]], range[x_]]]] := True
```

Theorem. (Eliminating the variable for the submonoid.)

```
In[69]:= Map[equal[V, #] &,
          SubstTest[class, t, implies[and[member[x, u], member[t, v]], member[t, w]],
           {u → GROUPS, v → intersection[MONOIDS, P[x],
              image[inverse[IMAGE[SECOND]], invar[inv[x]]]], w → GROUPS}]]

Out[69]= or[not[member[x, GROUPS]], subclass[intersection[MONOIDS,
            image[inverse[IMAGE[SECOND]], invar[inv[x]]], P[x]], GROUPS]] == True

In[70]:= or[not[member[x_, GROUPS]], subclass[intersection[MONOIDS,
            image[inverse[IMAGE[SECOND]], invar[inv[x_]]], P[x_]], GROUPS]] := True
```