intersections of subgroups of a group

Johan G. F. Belinfante
2011 June 6

In[1]:= **SetDirectory["l:"]; << goedel.11jun05a**

```
:Package Title: goedel.11jun05a                 2011 June 5 at 2:30 a.m.

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

It is now:  2011 Jun 6 at 14:52

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 Jun 6 at 15:3
```

summary

The intersection of any nonenpty collection of subgroups of a group is a subgroup. The key result needed to derive this fact is a rewrite rule that currently explicitly involves **setpart**.

In[2]:= **Aclosure[image[IMAGE[oopart[x]], setpart[y]]]**

Out[2]= image[IMAGE[oopart[x]], Aclosure[setpart[y]]]

a new conditional rewrite rule

To facilitate the derivation of **Aclosure** rules without having to always remember to use a **setpart** wrapper, a new conditional rewrite rule will be introduced.

Lemma. (Eliminating the **setpart** wrapper.)

In[6]:= **Map[implies[member[y, z], #] &,
SubstTest[implies, equal[y, setpart[t]], equal[Aclosure[image[IMAGE[oopart[x]], y]],
image[IMAGE[oopart[x]], Aclosure[y]]], t → y]] // Reverse**

Out[6]= or[equal[Aclosure[image[IMAGE[oopart[x]], y]], image[IMAGE[oopart[x]], Aclosure[y]]],
not[member[y, z]]] == True

In[7]:= **or[equal[Aclosure[image[IMAGE[oopart[x_]], y_]],
image[IMAGE[oopart[x_]], Aclosure[y_]]], not[member[y_, z_]]] := True**

Lemma. (Eliminating the **oopart** wrapper.)

```
In[11]:= SubstTest[implies, equal[x, oopart[t]],
           or[equal[Aclosure[image[IMAGE[x], y]], image[IMAGE[x], Aclosure[y]]],
            not[member[y, z]]], t → x] // Reverse

Out[11]= or[equal[Aclosure[image[IMAGE[x], y]], image[IMAGE[x], Aclosure[y]]],
           not[FUNCTION[x]], not[FUNCTION[inverse[x]]], not[member[y, z]]] == True

In[12]:= or[equal[Aclosure[image[IMAGE[x_], y_]], image[IMAGE[x_], Aclosure[y_]]],
           not[FUNCTION[x_]], not[FUNCTION[inverse[x_]]], not[member[y_, z_]]] := True
```

Theorem. A conditional rewrite rule.

```
In[13]:= implies[and[ONEONE[x], member[y, V]],
          equal[Aclosure[image[IMAGE[x], y]], image[IMAGE[x], Aclosure[y]]]]

Out[13]= True

In[15]:= Aclosure[image[IMAGE[x_], y_]] :=
          image[IMAGE[x], Aclosure[y]] /; and[ONEONE[x], member[y, V]]
```

derivation of an Aclosure rule for subgroups

Theorem. Any collection of subsets of a group is determined by the collection of their domains.

```
In[16]:= SubstTest[image, IMAGE[composite[id[funpart[t]], inverse[FIRST]]],
           image[IMAGE[FIRST], intersection[y, P[funpart[t]]]], t → gp[x]] // Reverse

Out[16]= image[IMAGE[composite[id[gp[x]], inverse[FIRST]]],
           image[IMAGE[FIRST], intersection[y, P[gp[x]]]]] == intersection[y, P[gp[x]]]

In[17]:= image[IMAGE[composite[id[gp[x_]], inverse[FIRST]]],
           image[IMAGE[FIRST], intersection[y_, P[gp[x_]]]]] := intersection[y, P[gp[x]]]
```

Theorem. The class of ranges of subgroups of a group is closed under arbitrary intersections.

```
In[18]:= SubstTest[Aclosure,
           intersection[binclosed[t], complement[P[complement[set[e[gp[x]]]]]]],
            fix[IMAGE[inv[gp[x]]]]], t → gp[x]] // Reverse

Out[18]= Aclosure[image[IMAGE[SECOND], intersection[GROUPS, P[gp[x]]]]] ==
          image[IMAGE[SECOND], intersection[GROUPS, P[gp[x]]]]

In[19]:= Aclosure[image[IMAGE[SECOND], intersection[GROUPS, P[gp[x_]]]]] :=
          image[IMAGE[SECOND], intersection[GROUPS, P[gp[x]]]]
```

Theorem. The class of domains of subgroups of a group is closed under arbitrary intersections.

```
In[20]:= SubstTest[Aclosure, image[CART, id[t]],
           t -> image[IMAGE[SECOND], intersection[GROUPS, P[gp[x]]]]] // Reverse

Out[20]= Aclosure[image[IMAGE[FIRST], intersection[GROUPS, P[gp[x]]]]] ==
          image[IMAGE[FIRST], intersection[GROUPS, P[gp[x]]]]

In[21]:= Aclosure[image[IMAGE[FIRST], intersection[GROUPS, P[gp[x_]]]]] :=
          image[IMAGE[FIRST], intersection[GROUPS, P[gp[x]]]]
```

Theorem. The class of subgroups of a group is closed under arbitrary intersections.

```
In[23]:= SubstTest[Aclosure, image[IMAGE[composite[id[gp[x]], inverse[FIRST]]], setpart[t]],
           t -> image[IMAGE[FIRST], intersection[GROUPS, P[gp[x]]]]] // Reverse

Out[23]= Aclosure[intersection[GROUPS, P[gp[x]]]] == intersection[GROUPS, P[gp[x]]]

In[24]:= Aclosure[intersection[GROUPS, P[gp[x_]]]] := intersection[GROUPS, P[gp[x]]]
```