duality in group theory

Johan G. F. Belinfante
2009 February 10

In[1]:= SetDirectory["l:"]; << goedel.09feb09a; << tools.m

Package Title: goedel.09feb09a 2009 February 9 at 4:00 p.m.

It is now: 2009 Feb 10 at 18:1

Loading Simplification Rules

TOOLS.M Revised 2009 February 9

weightlimit = 40

summary

If \(x\) is a group, so is \(\text{flip}[x]\), called the opposite group. For every theorem in group theory a dual theorem is obtained by replacing the group with its opposite. When \(\text{gp}\) wrappers are used, some special rewrite rules are useful to facilitate the application of duality. A simple example illustrating the principle of duality is presented in the final section.

simplification rules for \(\text{gp}[x]\)

In this section, some general simplification rules for the \(\text{gp}\) wrapper are derived by using the fact that groups are binary operations.

Theorem.

\[
\text{In}[2] := \text{SubstTest}[\text{composite}, \text{binog}[t], \text{id}[\text{cart}[V, V]], t \rightarrow \text{gp}[x]] \quad \text{// Reverse}
\]

\[
\text{Out}[2] = \text{composite}[\text{gp}[x], \text{id}[\text{cart}[V, V]]] := \text{gp}[x]
\]

\[
\text{In}[3] := \text{composite}[\text{gp}[x_\_], \text{id}[\text{cart}[V, V]]] := \text{gp}[x]
\]

Corollary.

\[
\text{In}[4] := \text{ImageComp}[\text{gp}[x], \text{id}[\text{cart}[V, V]], V] \quad \text{// Reverse}
\]

\[
\text{Out}[4] = \text{image}[\text{gp}[x], \text{cart}[V, V]] := \text{range}[\text{gp}[x]]
\]

\[
\text{In}[5] := \text{image}[\text{gp}[x], \text{cart}[V, V]] := \text{range}[\text{gp}[x]]
\]

Corollary. Inverse images of \(\text{gp}[x]\) are relations.
The special case that \( y \) is \( V \) requires a separate rule.

Corollary. The domain of \( \text{gp}[x] \) is a relation.

\[
\text{In}[8]:= \text{ImComp}[\text{gp}[x], \text{id[cart[V, V]}], V] // \text{Reverse}
\]

\[
\text{Out}[8]= \text{composite}[\text{Id, image[inverse[gp[x]]}], y] = \text{image[inverse[gp[x]]], y]
\]

\[
\text{In}[9]:= \text{composite}[\text{Id, image[inverse[gp[x_]}], y_}] = \text{image[inverse[gp[x]]}, y]
\]

\textbf{quasigroup rules}

By definition, a group is a nonempty associative quasigroup. Accordingly, one can obtain many rewrite rules for groups by specializing known facts about quasigroups.

Theorem.

\[
\text{In}[10]:= \text{SubstTest[implies, member[t, GROUPS], member[t, QUASIGPS], t \rightarrow \text{gp[x]}]} // \text{Reverse}
\]

\[
\text{Out}[10]= \text{member[gp[x], QUASIGPS]} = \text{True}
\]

\[
\text{In}[11]:= \text{member[gp[x_], QUASIGPS] := True}
\]

The final corollary of the preceding section can be sharpened:

Theorem. The relation \( \text{domain[gp[x]}] \) is the cartesian square of \( \text{range[gp[x]}] \).

\[
\text{In}[13]:= \text{SubstTest[cartsq, range[quasigp[t]], t \rightarrow \text{gp[x]}]} // \text{Reverse}
\]

\[
\text{Out}[13]= \text{cart[range[gp[x]], range[gp[x]]]} = \text{domain[gp[x]]}
\]

\[
\text{In}[14]:= \text{cart[range[gp[x_]], range[gp[x_]]]} = \text{domain[gp[x]]}
\]

Corollary. The domain of the domain of \( \text{gp}[x] \) is its range.

\[
\text{In}[15]:= \text{SubstTest[domain, domain[quasigp[t]], t \rightarrow \text{gp[x]}]} // \text{Reverse}
\]

\[
\text{Out}[15]= \text{domain[domain[gp[x]]]} = \text{range[gp[x]]}
\]

\[
\text{In}[16]:= \text{domain[domain[gp[x_]]]} = \text{range[gp[x]]}
\]

Corollary. The fixed-point class of the domain of \( \text{gp}[x] \) is its range.
empty gp[x] rules

Theorem. If \( x \) is not a group, then \( gp[x] \) is empty.

\[
\text{In[21]} := \text{Map}[\text{implies}[#, \text{empty}[gp[x]]], \&,
\text{SubstTest}[\text{empty, intersection}[x, \text{image}[V, \text{intersection}[\text{set}[x], t]]], t \rightarrow \text{GROUPS}]]
\]

\[
\text{Out[21]} = \text{or}[\text{equal}[0, gp[x]], \text{member}[x, \text{GROUPS}]] = \text{True}
\]

\[
\text{In[22]} := \text{or}[\text{equal}[0, gp[x]], \text{member}[x, \text{GROUPS}]] := \text{True}
\]

Theorem. A function is empty if its domain is empty.

\[
\text{In[23]} := \text{SubstTest}[\text{empty, domain}[\text{funpart}[t]], t \rightarrow \text{gp}[x]] // \text{Reverse}
\]

\[
\text{Out[23]} = \text{equal}[0, \text{domain}[gp[x]]] = \text{equal}[0, gp[x]]
\]

\[
\text{In[24]} := \text{equal}[0, \text{domain}[gp[x]]] := \text{equal}[0, gp[x]]
\]

Observation. A similar rule holds for the domain of the domain, but no new rewrite rules are needed, thanks to the results derived in the preceding section:

\[
\text{In[25]} := \text{equal}[0, \text{domain}[\text{domain}[gp[x]]]]
\]

\[
\text{Out[25]} = \text{equal}[0, gp[x]]
\]


duality rules for gp[x]

Theorem. Groups are sets.

\[
\text{In[26]} := \text{SubstTest}[\text{member, binop}[t], V, t \rightarrow \text{gp}[x]] // \text{Reverse}
\]

\[
\text{Out[26]} = \text{member}[gp[x], V] = \text{True}
\]

\[
\text{In[27]} := \text{member}[gp[x], V] := \text{True}
\]
Theorem. The flip\(gp[x]\) is a group if and only if \(gp[x]\) is a group.

\[\text{In}[28]:= \text{member}[\text{composite}[gp[x], SWAP], \text{GROUPS}] \text{// AssertTest} \]

\[\text{Out}[28]= \text{member}[\text{composite}[gp[x], SWAP], \text{GROUPS}] = \text{not[equal[0, gp[x]]]}\]

\[\text{In}[29]:= \text{member}[\text{composite}[gp[x_], SWAP], \text{GROUPS}] := \text{not[equal[0, gp[x]]]}\]

---

**special wrapper removal rules**

Special wrapper removal rules for the \(gp\) wrapper are derived in this section.

Theorem. Iterated wrapper rule.

\[\text{In}[30]:= \text{equal}[gp[gp[x]], gp[x]] \]

\[\text{Out}[30]= \text{True}\]

\[\text{In}[31]:= \text{gp[gp[x_]]} := \text{gp[x]}\]

Theorem. A similar special wrapper-removal rule for opposite groups.

\[\text{In}[32]:= \text{equal}[\text{gp[flip[gp[x]]]}, \text{flip[gp[x]]}] \]

\[\text{Out}[32]= \text{True}\]

\[\text{In}[33]:= \text{gp[composite}[gp[x_], \text{SWAP}]] := \text{composite}[gp[x], \text{SWAP}]\]

---

**dual associative laws**

The associative law for groups can be formulated without variables for group elements. Two rewrite rules are derived in this section, related by duality.

Theorem. A rewrite rule for the associative law.

\[\text{In}[34]:= \text{SubstTest}[\text{composite}, \text{cat[t], cross[Id, cat[t]], ASSOC, t \rightarrow gp[x]}] \text{// Reverse} \]

\[\text{Out}[34]= \text{composite}[gp[x], \text{cross[Id, gp[x]]}, \text{ASSOC}] = \text{composite}[gp[x], \text{cross[gp[x], Id]}] \]

\[\text{In}[35]:= \text{composite}[gp[x_], \text{cross[Id, gp[x_]]}, \text{ASSOC}] := \text{composite}[gp[x], \text{cross[gp[x], Id]}] \]

One can use duality to derive a similar result. To obtain a clean rewrite rule, the following lemma is needed.

Lemma.
Application of duality now yields the following:

```
In[49]:= Map[composite#, cross[SWAP, Id], SWAP] &,
   SubstTest[composite, assoc[t], cross[assoc[t], Id], inverse[ASSOC], t \[Rule] flip[gp[x]]]] // Reverse
```

```
Out[49]= composite[gp[x], cross[gp[x], Id], inverse[ASSOC]] \[Equal] composite[gp[x], cross[Id, gp[x]]]
```

The same result can also obtained more directly without explicitly using duality.

Theorem. A dual rewrite rule for the associative law. (Here the same result is obtained from the general theory of associative relations.)

```
In[50]:= SubstTest[composite, assoc[t], cross[assoc[t], Id], inverse[ASSOC], t \[Rule] gp[x]] // Reverse
```

```
Out[50]= composite[gp[x], cross[gp[x], Id], inverse[ASSOC]] \[Equal] composite[gp[x], cross[Id, gp[x]]]
```

```
In[51]:= composite[gp[x_], cross[gp[x_], Id], inverse[ASSOC]] :=
   composite[gp[x], cross[Id, gp[x]]]
```