inverse elements in groups, part 2

Johan G. F. Belinfante
2009 February 7

```
In[1]:= << l:goedel.09feb06a; << l:tools.m

        :Package Title: goedel.09feb06a            2009 February 6 at 9:10 a.m.

        It is now:  2009 Feb 7 at 22:38

        Loading Simplification Rules

        TOOLS.M                                    Revised 2009 February 4

        weightlimit = 40
```

summary

This second notebook on inverse elements in groups features a direct translation into the language of the **GOEDEL** program of the usual proof that the inverse of a product of two elements **u**, **v ∈ range[x]** of a group **x** is the product of their inverses in the reverse order. To prepare for this, one needs to know that if a product of two elements is the identity element **e[x]**, then each of the two elements is the inverse of the other. Since the symmetric inverse-pair relation **inv[x]** for a group **x** is a function, the statement that **pair[u, v] ∈ inv[x]** is equivalent to various others, including the statement that **v = APPLY[inv[x], u]** and the statement **e[x] = APPLY[x, PAIR[u, v]]**. Once these various equivalences are made available, the proof of the main theorem is attacked, using the lemma $\mathbf{u^{-1} \cdot (u \cdot v) = v}$ and a similar one with the roles of **u** and its inverse interchanged. These theorems, though quite straight-forward applications of associativity and basic properties of identity elements and inverses, have to be carefully laid out to keep execution time to a minimum. Another problem with this pedestrian approach is the presence of expressions of the form **APPLY[x, PAIR[u, v]]** without any wrapper on **x** to indicate to the **GOEDEL** program that the variable **x** is a function, which makes it quite tricky to eliminate the variables **u** and **v**.

inverse pairs in a group

In this section various ways of saying that a pair of elements of a group are inverses of each other are shown to be equivalent.

Lemma. If **x** is a group and if **pair[u, v] ∈ inv[x]**, then **u ∈ range[x]**.

```
In[2]:= Map[not, SubstTest[and, implies[p1, p4], implies[p2, p3],
           implies[and[p3, p4], p5], not[implies[and[p1, p2], p5]], {p1 → member[x, GROUPS],
            p2 → member[pair[u, v], inv[x]], p3 → member[pair[u, v], domain[x]],
            p4 → equal[domain[x], cartsq[range[x]]], p5 → member[u, range[x]]}]] // Reverse

Out[2]= or[member[u, range[x]], not[member[x, GROUPS]], not[member[pair[u, v], inv[x]]]] == True
```

```
In[3]:= or[member[u_, range[x_]], not[member[x_, GROUPS]],
          not[member[pair[u_, v_], inv[x_]]]] := True
```

Lemma. If **x** is a group and if **pair[u, v] ∈ inv[x]**, then **v ∈ range[x]**.

```
In[4]:= Map[not, SubstTest[and, implies[p1, p4], implies[p2, p3],
           implies[and[p3, p4], p5], not[implies[and[p1, p2], p5]], {p1 → member[x, GROUPS],
            p2 → member[pair[u, v], inv[x]], p3 → member[pair[u, v], domain[x]],
            p4 → equal[domain[x], cartsq[range[x]]], p5 → member[v, range[x]]}]] // Reverse

Out[4]= or[member[v, range[x]], not[member[x, GROUPS]], not[member[pair[u, v], inv[x]]]] == True
```

```
In[5]:= or[member[v_, range[x_]], not[member[x_, GROUPS]],
          not[member[pair[u_, v_], inv[x_]]]] := True
```

Theorem. If **x** is a group and if **pair[u, v] ∈ inv[x]**, then **u · v = e[x]**.

```
In[6]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3],
           not[implies[p1, p3]], {p1 → and[member[x, GROUPS], member[pair[u, v], inv[x]]],
            p2 → member[x, MONOIDS], p3 → equal[APPLY[x, PAIR[u, v]], e[x]]}]] // Reverse

Out[6]= or[equal[APPLY[x, PAIR[u, v]], e[x]],
          not[member[x, GROUPS]], not[member[pair[u, v], inv[x]]]] == True
```

```
In[7]:= or[equal[APPLY[x_, PAIR[u_, v_]], e[x_]],
          not[member[x_, GROUPS]], not[member[pair[u_, v_], inv[x_]]]] := True
```

Corollary. If **x** is a group and if **pair[u, v] ∈ inv[x]**, then **v · u = e[x]**.

```
In[8]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3],
           not[implies[p1, p3]], {p1 → and[member[x, GROUPS], member[pair[u, v], inv[x]]],
            p2 → member[pair[v, u], inv[x]], p3 → equal[APPLY[x, PAIR[v, u]], e[x]]}]] // Reverse

Out[8]= or[equal[APPLY[x, PAIR[v, u]], e[x]],
          not[member[x, GROUPS]], not[member[pair[u, v], inv[x]]]] == True
```

```
In[9]:= or[equal[APPLY[x_, PAIR[v_, u_]], e[x_]],
          not[member[x_, GROUPS]], not[member[pair[u_, v_], inv[x_]]]] := True
```

Theorem. If **x** is a group and if **pair[u, v] ∈ inv[x]**, then **v = u^{-1}**.

```
In[10]:= Map[not, SubstTest[and, implies[p1, p3],
            implies[and[p2, p3], p4], not[implies[and[p1, p2], p4]],
            {p1 → member[x, GROUPS], p2 -> member[pair[u, v], inv[x]],
             p3 → FUNCTION[inv[x]], p4 → equal[v, APPLY[inv[x], u]]}]] // Reverse

Out[10]= or[equal[v, APPLY[inv[x], u]],
           not[member[x, GROUPS]], not[member[pair[u, v], inv[x]]]] == True
```

```
In[11]:= or[equal[v_, APPLY[inv[x_], u_]],
           not[member[x_, GROUPS]], not[member[pair[u_, v_], inv[x_]]]] := True
```

Corollary. If **x** is a group and if **pair[u, v] ∈ inv[x]**, then **u = v^{-1}**.

```
In[12]:= Map[not, SubstTest[and, implies[p2, p3],
           implies[and[p1, p3], p4], not[implies[and[p1, p2], p4]],
           {p1 → member[x, GROUPS], p2 -> member[pair[u, v], inv[x]],
            p3 → member[pair[v, u], inv[x]], p4 → equal[u, APPLY[inv[x], v]]}]] // Reverse

Out[12]= or[equal[u, APPLY[inv[x], v]],
           not[member[x, GROUPS]], not[member[pair[u, v], inv[x]]]] == True

In[13]:= or[equal[u_, APPLY[inv[x_], v_]],
           not[member[x_, GROUPS]], not[member[pair[u_, v_], inv[x_]]]] := True
```

Theorem. If **x** is a group and if $\mathbf{u, v \in range[x]}$ satisfy $\mathbf{u \cdot v = e[x]}$, then also $\mathbf{v \cdot u = e[x]}$.

```
In[14]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3], not[implies[p1, p3]],
           {p1 -> and[member[x, GROUPS], equal[e[x], APPLY[x, PAIR[u, v]]]],
            p2 → member[pair[u, v], inv[x]], p3 → equal[e[x], APPLY[x, PAIR[v, u]]]}]] // Reverse

Out[14]= or[equal[APPLY[x, PAIR[v, u]], e[x]],
           not[equal[APPLY[x, PAIR[u, v]], e[x]]], not[member[x, GROUPS]]] == True

In[15]:= or[equal[APPLY[x_, PAIR[v_, u_]], e[x_]],
           not[equal[APPLY[x_, PAIR[u_, v_]], e[x_]]], not[member[x_, GROUPS]]] := True
```

Theorem. If **x** is a group and if $\mathbf{u \cdot v = e[x]}$, then $\mathbf{u = v^{-1}}$.

```
In[16]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3], not[implies[p1, p3]],
           {p1 -> and[member[x, GROUPS], equal[APPLY[x, PAIR[u, v]], e[x]]], p2 ->
             equal[APPLY[x, PAIR[v, u]], e[x]], p3 -> equal[u, APPLY[inv[x], v]]}]] // Reverse

Out[16]= or[equal[u, APPLY[inv[x], v]],
           not[equal[APPLY[x, PAIR[u, v]], e[x]]], not[member[x, GROUPS]]] == True

In[17]:= or[equal[u_, APPLY[inv[x_], v_]],
           not[equal[APPLY[x_, PAIR[u_, v_]], e[x_]]], not[member[x_, GROUPS]]] := True
```

---

inverse of a product

Lemma. If **x** is a group and $\mathbf{u, v \in range[x]}$, then $\mathbf{u^{-1} \cdot (u \cdot v) = v}$.

```
In[18]:= (Map[not, SubstTest[and, implies[p1, p2], implies[p1, p3], not[implies[p1, p4]],
            {p1 → and[equal[s, APPLY[inv[x], u]], equal[t, APPLY[x, PAIR[u, v]]],
               member[x, GROUPS], member[u, range[x]], member[v, range[x]]],
             p2 -> equal[APPLY[x, PAIR[APPLY[x, PAIR[s, u]], v]], APPLY[x, PAIR[s, t]]],
             p3 → equal[e[x], APPLY[x, PAIR[s, u]]], p4 → equal[v, APPLY[x, PAIR[s, t]]]}] /.
           {s -> APPLY[inv[x], u], t -> APPLY[x, PAIR[u, v]]}) // Reverse

Out[18]= or[equal[v, APPLY[x, PAIR[APPLY[inv[x], u], APPLY[x, PAIR[u, v]]]]],
           not[member[u, range[x]]], not[member[v, range[x]]], not[member[x, GROUPS]]] == True
```

```
In[19]:= or[equal[v_, APPLY[x_, PAIR[APPLY[inv[x_], u_], APPLY[x_, PAIR[u_, v_]]]]],
           not[member[u_, range[x_]]], not[member[v_, range[x_]]],
           not[member[x_, GROUPS]]] := True
```

Corollary. If **x** is a group and **u**, **v** ∈ **range[x],** then $\mathbf{u \cdot (u^{-1} \cdot v) = v}$.

```
In[20]:= (Map[not, SubstTest[and, implies[p1, p2],
              implies[p1, p3], not[implies[p1, p4]], {p1 → and[member[x, GROUPS],
                 member[u, range[x]], member[v, range[x]], equal[t, APPLY[inv[x], u]]],
               p2 → member[t, range[x]],
               p3 → equal[u, APPLY[inv[x], t]],
               p4 -> equal[v, APPLY[x, PAIR[u, APPLY[x, PAIR[t, v]]]]]}]] /.
            t -> APPLY[inv[x], u]) // Reverse
```

```
Out[20]= or[equal[v, APPLY[x, PAIR[u, APPLY[x, PAIR[APPLY[inv[x], u], v]]]]],
           not[member[u, range[x]]], not[member[v, range[x]]], not[member[x, GROUPS]]] == True
```

```
In[21]:= or[equal[v_, APPLY[x_, PAIR[u_, APPLY[x_, PAIR[APPLY[inv[x_], u_], v_]]]]],
           not[member[u_, range[x_]]], not[member[v_, range[x_]]],
           not[member[x_, GROUPS]]] := True
```

Theorem. If **x** is a group and **u**, **v** ∈ **range[x]**, then $\mathbf{(u \cdot v)^{-1} = v^{-1} \cdot u^{-1}}$.

```
In[22]:= Map[not, SubstTest[and, implies[p1, p2], implies[p1, p3], implies[and[p2, p3], p4],
             implies[and[p1, p4], p5], implies[and[p1, p5], p6], not[implies[p1, p6]],
             {p1 → and[member[x, GROUPS], member[u, range[x]], member[v, range[x]]],
              p2 → equal[v, APPLY[x, PAIR[APPLY[inv[x], u], APPLY[x, PAIR[u, v]]]]],
              p3 → equal[APPLY[x, PAIR[APPLY[inv[x], v], v]], e[x]],
              p4 → equal[APPLY[x, PAIR[APPLY[inv[x], v],
                  APPLY[x, PAIR[APPLY[inv[x], u], APPLY[x, PAIR[u, v]]]]]], e[x]],
              p5 → equal[APPLY[x, PAIR[APPLY[x, PAIR[APPLY[inv[x], v], APPLY[inv[x], u]]],
                  APPLY[x, PAIR[u, v]]]], e[x]],
              p6 → equal[APPLY[x, PAIR[APPLY[inv[x], v], APPLY[inv[x], u]]],
                APPLY[inv[x], APPLY[x, PAIR[u, v]]]]}]] // Reverse
```

```
Out[22]= or[equal[APPLY[x, PAIR[APPLY[inv[x], v], APPLY[inv[x], u]]],
              APPLY[inv[x], APPLY[x, PAIR[u, v]]]], not[member[u, range[x]]],
           not[member[v, range[x]]], not[member[x, GROUPS]]] == True
```

```
In[23]:= or[equal[APPLY[x_, PAIR[APPLY[inv[x_], v_], APPLY[inv[x_], u_]]],
              APPLY[inv[x_], APPLY[x_, PAIR[u_, v_]]]], not[member[u_, range[x_]]],
           not[member[v_, range[x_]]], not[member[x_, GROUPS]]] := True
```