# binary homomorphisms preserve binclosed sets

## Johan G. F. Belinfante and Ming Li
## 2006 September 27

```
In[1]:= SetDirectory["l:"]; << goedel85.26a; << tools.m

        :Package Title: goedel85.26a          2006 September 26 at 3:00 p.m.

        It is now:  2006 Sep 27 at 21:31

        Loading Simplification Rules

        TOOLS.M                      Revised 2006 September 24

        weightlimit = 40
```

## summary

A standard theorem in group theory says that group homomorphisms take subgroups to subgroups. In this notebook a generalization of this is derived, but without the special hypotheses of group theory. For any binary homomorphism from **x** to **y**, sets closed under **x** are taken to sets closed under **y**.

## derivation

Lemma.

```
In[2]:= SubstTest[implies, equal[u, v], equal[image[u, z], image[v, z]],
          {u -> composite[t, x], v -> composite[y, cross[t, t]], z → cart[w, w]}]

Out[2]= or[equal[image[t, image[x, cart[w, w]]], image[y, cart[image[t, w], image[t, w]]]],
          not[equal[composite[t, x], composite[y, cross[t, t]]]]] == True

In[3]:= (% /. {t → t_, x → x_, w → w_, y → y_}) /. Equal → SetDelayed
```

Main theorem.

```
In[4]:= Map[not, SubstTest[and, implies[p3, p4], not[implies[and[p1, p2], p5]],
          {p1 → member[t, binhom[x, y]], p2 -> subclass[image[x, cart[w, w]], w],
           p3 -> equal[composite[t, x], composite[y, cross[t, t]]],
           p4 -> equal[image[t, image[x, cart[w, w]]], image[y, cart[image[t, w], image[t, w]]]],
           p5 -> subclass[image[y, cart[image[t, w], image[t, w]]], image[t, w]]}]]

Out[4]= or[not[member[t, binhom[x, y]]], not[subclass[image[x, cart[w, w]], w]],
          subclass[image[y, cart[image[t, w], image[t, w]]], image[t, w]]] == True
```

```
In[5]:= or[not[member[t_, binhom[x_, y_]]], not[subclass[image[x_, cart[w_, w_]], w_]],
          subclass[image[y_, cart[image[t_, w_], image[t_, w_]]], image[t_, w_]]] := True
```

## removing variables

Removing one of the variables yields:

```
In[6]:= Map[equal[V, #] &,
          SubstTest[class, v, implies[and[member[v, r], member[x, s]], member[image[x, v], t]],
           {r → binclosed[y], s → binhom[y, z], t → binclosed[z]}]] // Reverse

Out[6]= or[not[member[x, binhom[y, z]]],
          subclass[binclosed[y], image[inverse[IMAGE[x]], binclosed[z]]]] == True

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

Lemma.

```
In[8]:= SubstTest[implies, subclass[y, t], subclass[image[s, y], image[s, t]],
         {s → IMAGE[x], t -> image[inverse[IMAGE[x]], z]}]

Out[8]= or[not[subclass[y, image[inverse[IMAGE[x]], z]]],
          subclass[image[IMAGE[x], y], z]] == True

In[9]:= or[not[subclass[y_, image[inverse[IMAGE[x_]], z_]]],
          subclass[image[IMAGE[x_], y_], z_]] := True
```

Corollary.

```
In[10]:= Map[not, SubstTest[and, implies[p1, p2], not[implies[p1, p3]],
           {p1 -> member[x, binhom[y, z]],
            p2 -> subclass[binclosed[y], image[inverse[IMAGE[x]], binclosed[z]]],
            p3 → subclass[image[IMAGE[x], binclosed[y]], binclosed[z]]}]]

Out[10]= or[not[member[x, binhom[y, z]]],
           subclass[image[IMAGE[x], binclosed[y]], binclosed[z]]] == True

In[11]:= or[not[member[x_, binhom[y_, z_]]],
           subclass[image[IMAGE[x_], binclosed[y_]], binclosed[z_]]] := True
```

Removing a second variable yields an even more succinct result:

```
In[12]:= Map[equal[0, composite[Id, complement[#]]] &, SubstTest[class, pair[u, v],
            implies[and[member[v, r], member[u, s]], member[image[u, v], t]],
            {r → binclosed[x], s → binhom[x, y], t → binclosed[y]}]] // Reverse

Out[12]= subclass[image[IMG, cart[binhom[x, y], binclosed[x]]], binclosed[y]] == True

In[13]:= subclass[image[IMG, cart[binhom[x_, y_], binclosed[x_]]], binclosed[y_]] := True
```