# theorems about inverses

Johan G. F. Belinfante and Tiffany D. Goble
2004 February 26

```
In[1]:= << goedel54.25a; << tools.m

        :Package Title: goedel54.25a     2004 February 25 at 8:11 p.m.

        It is now:  2004 Feb 27 at 15:25

        Loading Simplification Rules

        TOOLS.M                      Revised 2004 February 21

        weightlimit = 40
```

## summary

Three theorems proved using **Otter** are rederived in this notebook in order to make these basic results available in the **GOEDEL** program.

## Theorem IN−DJ−1

The following lemma is of interest in its own right:

```
In[2]:= SubstTest[implies, equal[u, v],
         equal[range[u], range[v]], {u → 0, v → intersection[x, y]}]

Out[2]= or[equal[0, fix[composite[x, inverse[y]]]], not[equal[0, intersection[x, y]]]] == True

In[3]:= or[equal[0, fix[composite[x_, inverse[y_]]]],
          not[equal[0, intersection[x_, y_]]]] := True
```

Theorem **IN−DJ−1** follows:

```
In[4]:= or[not[disjoint[x, y]], disjoint[inverse[x], inverse[y]]] // AssertTest

Out[4]= or[equal[0, intersection[inverse[x], inverse[y]]],
          not[equal[0, intersection[x, y]]]] == True

In[5]:= or[equal[0, intersection[inverse[x_], inverse[y_]]],
          not[equal[0, intersection[x_, y_]]]] := True
```

Restatement:

```
In[6]:= or[not[disjoint[x, y]], disjoint[inverse[x], inverse[y]]]

Out[6]= True
```

The following result is similar to the lemma:

```
In[7]:= SubstTest[implies, equal[u, v],
         equal[domain[u], domain[v]], {u → 0, v → intersection[x, y]}]

Out[7]= or[equal[0, fix[composite[inverse[x], y]]], not[equal[0, intersection[x, y]]]] == True

In[8]:= or[equal[0, fix[composite[inverse[x_], y_]]],
           not[equal[0, intersection[x_, y_]]]] := True
```

## Theorem IN−DJ−2

Theorem **IN−DJ−2** is of a simlar nature:

```
In[9]:= or[equal[0, intersection[x, inverse[y]]],
           not[equal[0, intersection[y, inverse[x]]]]] // AssertTest

Out[9]= or[equal[0, intersection[x, inverse[y]]],
           not[equal[0, intersection[y, inverse[x]]]]] == True

In[10]:= or[equal[0, intersection[x_, inverse[y_]]],
            not[equal[0, intersection[y_, inverse[x_]]]]] := True
```

Restatement:

```
In[11]:= implies[disjoint[x, inverse[y]], disjoint[y, inverse[x]]]

Out[11]= True
```

## A replacement for Theorem IN−SU

Lemma.

```
In[12]:= SubstTest[implies, and[subclass[u, y], equal[u, x]],
          subclass[x, y], u → inverse[inverse[x]]]

Out[12]= or[not[subclass[x, cart[V, V]]],
            not[subclass[composite[Id, x], y]], subclass[x, y]] == True

In[13]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem **IN−SU**, which was proved 1998 December 8 using **Otter**, follows as a corollary:

```
In[14]:= implies[and[subclass[x, cart[V, V]], subclass[inverse[x], inverse[y]]], subclass[x, y]]

Out[14]= True
```

The following is similar to the lemma, but with a more general hypothesis:

```
In[15]:= Map[not, SubstTest[and, implies[p1, p2],
            implies[and[p2, p3], p4], not[implies[and[p1, p3], p4]],
            {p1 → subclass[x, cart[u, v]], p2 → subclass[x, cart[V, V]],
             p3 → subclass[composite[Id, x], y], p4 → subclass[x, y]}]]

Out[15]= or[not[subclass[x, cart[u, v]]],
            not[subclass[composite[Id, x], y]], subclass[x, y]] == True
```

```
In[16]:= or[not[subclass[x_, cart[u_, v_]]],
           not[subclass[composite[Id, x_], y_]], subclass[x_, y_]] := True
```