Theorem ID−A−C

Johan G. F. Belinfante
2004 February 22

```
In[1]:= << goedel54.22b; << tools.m

        :Package Title: goedel54.22b     2004 February 22 at 9:55 p.m.

        It is now:  2004 Feb 23 at 12:31

        Loading Simplification Rules

        TOOLS.M                      Revised 2004 February 11

        weightlimit = 40
```

summary

Theorem **ID−A−C** is a curious theorem that was proved 1998 January 15 using **Otter**. The derivation of this theorem in this notebook follows my handwritten notes and not the proof that **Otter** found. The two set variables **u** and **v** that are eliminated in the final statement appear as Skolem functions in the **Otter** proof.

derivation

Lemma 1.

```
In[2]:= SubstTest[implies, and[member[u, v], member[v, x]], member[u, U[x]], v -> singleton[u]]

Out[2]= or[member[u, U[x]], not[member[u, V]], not[member[singleton[u], x]]] == True

In[3]:= (% /. {u -> u_, x -> x_}) /. Equal -> SetDelayed
```

Lemma 2.

```
In[4]:= SubstTest[implies, and[member[v, A[z]], member[y, z]],
          member[v, y], {y -> singleton[u], z -> complement[x]}] // MapNotNot

Out[4]= or[equal[u, v], member[singleton[u], x], not[member[v, A[complement[x]]]]] == True

In[5]:= (% /. {u -> u_, v -> v_, x -> x_}) /. Equal -> SetDelayed
```

These two lemmas are combined as follows:

```
In[6]:= Map[not, SubstTest[and, implies[p1, p2],
          implies[and[p2, p3], p4], not[implies[and[p1, p3], p4]],
          {p1 -> member[u, complement[U[x]]], p2 -> not[member[singleton[u], x]],
           p3 -> member[v, A[complement[x]]], p4 -> equal[u, v]}]]

Out[6]= or[equal[u, v], member[u, U[x]],
          not[member[u, V]], not[member[v, A[complement[x]]]]] == True
```

```
In[7]:= (% /. {u -> u_, v -> v_, x -> x_}) /. Equal -> SetDelayed
```

The two set variables **u** and **v** are eliminated to derive the main theorem:

```
In[8]:= Map[equal[0, composite[Id, complement[#]]] &,
          SubstTest[class, pair[u, v], or[equal[u, v], not[member[u, y]], not[member[v, z]]],
            {y -> complement[U[x]], z -> A[complement[x]]}]] // Reverse

Out[8]= subclass[image[Di, A[complement[x]]], U[x]] == True
```

```
In[9]:= subclass[image[Di, A[complement[x_]]], U[x_]] := True
```

The original statement of the theorem is equivalent to the above:

```
In[10]:= subclass[cart[complement[U[x]], A[complement[x]]], Id]

Out[10]= True
```

an example

For most classes, either **complement[U[x]]** or **A[complement[x]]** is empty, so their cartesian product is empty. An example of a class for which these are both singletons is the class of sets that do not hold the empty set:

```
In[11]:= class[u, not[member[0, u]]]

Out[11]= P[complement[singleton[0]]]
```

In this case one has:

```
In[12]:= cart[complement[U[x]], A[complement[x]]] /. x -> P[complement[singleton[0]]]

Out[12]= cart[singleton[0], singleton[0]]
```