finite changes to a countably infinite set

Johan G. F. Belinfante
2006 August 17

```
In[1]:= SetDirectory["l:"]; << goedel84.16c; << tools.m

        :Package Title: goedel84.16c       2006 August 16 at 12:35 noon

        It is now:  2006 Aug 17 at 7:37

        Loading Simplification Rules

        TOOLS.M                        Revised 2006 August 15

        weightlimit = 40
```

summary

If one inserts a finite number of elements into a countably infinite set or if one removes a finite number of elements from a countably infinite set, the resulting set remains countably infinite.

a special case

For the special case of **omega**, removing all numbers up to some point yields an infinite set.

```
In[2]:= SubstTest[member, union[u, v], FINITE, {u → nat[x], v -> dif[omega, nat[x]]}] // Reverse

Out[2]= member[intersection[omega, complement[nat[x]]], FINITE] == False

In[3]:= member[intersection[omega, complement[nat[x_]]], FINITE] := False
```

If a subset of omega is not finite, it is countably infinite.

```
In[4]:= SubstTest[implies, subclass[w, omega],
         or[member[w, FINITE], equal[omega, card[w]]], w -> dif[omega, nat[x]]]

Out[4]= equal[omega, card[intersection[omega, complement[nat[x]]]]] == True

In[5]:= card[intersection[omega, complement[nat[x_]]]] := omega
```

lemmas

Lemma.

```
In[6]:=  SubstTest[implies, member[pair[u, v], Q],
          equal[image[Q, set[u]], image[Q, set[v]]], {u → omega, v → dif[omega, nat[x]]}]

Out[6]=  equal[image[Q, set[omega]],
           image[Q, set[intersection[omega, complement[nat[x]]]]]] == True

In[7]:=  image[Q, set[intersection[omega, complement[nat[x_]]]]] := image[Q, set[omega]]
```

Corollary.

```
In[8]:=  SubstTest[member, y, image[Q, set[z]], z -> dif[omega, nat[x]]] // Reverse

Out[8]=  member[pair[intersection[omega, complement[nat[x]]], y], Q] == equal[omega, card[y]]

In[9]:=  member[pair[intersection[omega, complement[nat[x_]]], y_], Q] := equal[omega, card[y]]
```

The following rewrite rule will also be needed shortly.

```
In[10]:=  equal[union[omega, nat[x]], omega]

Out[10]=  True

In[11]:=  union[omega, nat[x_]] := omega
```

---

adding new elements

Cardinality is preserved for disjoint unions. The special case of disjoint unions is considered in this section. Later on, the restriction to disjoint unions will be removed.

```
In[12]:=  SubstTest[implies,
           and[member[pair[u, x], Q], member[pair[v, y], Q], disjoint[u, v], disjoint[x, y]],
           member[pair[union[u, v], union[x, y]], Q], {u → nat[z], v → dif[omega, nat[z]]}]

Out[12]=  or[equal[omega, card[union[x, y]]], not[equal[0, intersection[x, y]]],
            not[equal[omega, card[y]]], not[equal[card[x], nat[z]]]] == True

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

The variable **z** is removed:

```
In[14]:=  Map[implies[member[x, FINITE], equal[V, #]] &,
              SubstTest[class, z, or[equal[omega, u], not[equal[0, v]], not[equal[omega, w]],
                not[equal[t, nat[z]]]], {t → card[x], u -> card[union[x, y]],
                v -> intersection[x, y], w -> card[y]}]] // MapNotNot // Reverse

Out[14]=  or[equal[omega, card[union[x, y]]], not[equal[0, intersection[x, y]]],
            not[equal[omega, card[y]]], not[member[x, FINITE]]] == True

In[15]:=  (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

the general case

In this section the restriction to disjointness unions will be removed. The disjointness literal is removed as follows:

```
In[16]:= SubstTest[or, equal[omega, card[union[w, y]]], not[equal[0, intersection[w, y]]],
          not[equal[omega, card[y]]], not[member[w, FINITE]], w → dif[x, y]]

Out[16]= or[equal[omega, card[union[x, y]]], not[equal[omega, card[y]]],
           not[member[intersection[x, complement[y]], FINITE]]] == True

In[17]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. The union of a finite set and a countably infinite set is countably infinite.

```
In[18]:= Map[not, SubstTest[and, implies[p1, p3], implies[and[p2, p3], p4],
           not[implies[and[p1, p2], p4]], {p1 → member[x, FINITE],
            p2 -> equal[omega, card[y]], p3 -> member[intersection[x, complement[y]], FINITE],
            p4 → equal[omega, card[union[x, y]]]}]]

Out[18]= or[equal[omega, card[union[x, y]]],
           not[equal[omega, card[y]]], not[member[x, FINITE]]] == True

In[19]:= or[equal[omega, card[union[x_, y_]]],
           not[equal[omega, card[y_]]], not[member[x_, FINITE]]] := True
```

variable-free formulation

Lemma.

```
In[20]:= or[and[equal[omega, card[union[x, y]]], member[y, V]],
           not[equal[omega, card[y]]], not[member[x, FINITE]]] // NotNotTest

Out[20]= or[and[equal[omega, card[union[x, y]]], member[y, V]],
           not[equal[omega, card[y]]], not[member[x, FINITE]]] == True

In[21]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Eliminating the variables yields a statement about **K**-invariance. The introduction of the cover relation **K** here is the result of rewrite rules.

```
In[22]:= Map[equal[0, composite[Id, complement[#]]] &, SubstTest[class, pair[x, y],
            implies[and[member[x, u], member[y, v]], member[union[x, y], v]],
            {u → FINITE, v → image[Q, set[omega]]}]] // Reverse

Out[22]= subclass[image[K, image[Q, set[omega]]], image[Q, set[omega]]] == True

In[23]:= subclass[image[K, image[Q, set[omega]]], image[Q, set[omega]]] := True
```

A more natural equational reformulation can be derived as a corollary:

```
In[24]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> image[Q, set[omega]], v -> image[CUP, cart[FINITE, image[Q, set[omega]]]]}]

Out[24]= True == equal[image[CUP, cart[FINITE, image[Q, set[omega]]]], image[Q, set[omega]]]

In[25]:= image[CUP, cart[FINITE, image[Q, set[omega]]]] := image[Q, set[omega]]
```

removing a finite number of elements

Every subset of a countably infinite set is either finite or countably infinite.

```
In[26]:= SubstTest[implies, and[equal[omega, card[x]], subclass[z, x]],
          or[member[z, FINITE], equal[omega, card[z]]], z → intersection[x, y]]

Out[26]= or[equal[omega, card[intersection[x, y]]],
           member[intersection[x, y], FINITE], not[equal[omega, card[x]]]] == True

In[27]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

A union of two sets is finite if and only if both sets are finite.

```
In[28]:= Map[implies[#, member[x, FINITE]] &,
           SubstTest[member, union[u, v], FINITE, {u → dif[x, y], v → y}]] // Reverse

Out[28]= or[member[x, FINITE], not[member[y, FINITE]],
           not[member[intersection[x, complement[y]], FINITE]]] == True

In[29]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. Removing a finite number of elements from a countably infinite set yields a countably infinite subset.

```
In[30]:= Map[not, SubstTest[and, implies[p1, p3], implies[and[p2, p3], p4],
           implies[and[p1, p4], p5], not[implies[and[p1, p2], p5]],
           {p1 -> equal[omega, card[x]], p2 -> member[y, FINITE], p3 → not[member[x, FINITE]],
            p4 → not[member[dif[x, y], FINITE]], p5 -> equal[omega, card[dif[x, y]]]}]]

Out[30]= or[equal[omega, card[intersection[x, complement[y]]]],
           not[equal[omega, card[x]]], not[member[y, FINITE]]] == True

In[31]:= or[equal[omega, card[intersection[complement[y_], x_]]],
           not[equal[omega, card[x_]]], not[member[y_, FINITE]]] := True
```

variable-free formulation

The following observation is used to derive the first lemma.

```
In[36]:= abstract[x, image[DIF, cart[image[Q, set[omega]], x]]]

Out[36]= composite[DIF, id[cart[image[Q, set[omega]], V]], inverse[SECOND]]
```

Since the empty set is finite, one obtains the following general inclusion:

```
In[32]:= SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
          {u → set[0], v → FINITE, w -> composite[DIF, id[cart[x, V]], inverse[SECOND]]}]

Out[32]= subclass[x, image[DIF, cart[x, FINITE]]] == True

In[33]:= subclass[x_, image[DIF, cart[x_, FINITE]]] := True
```

Eliminating the variables from the theorem derived in the preceding section yields an inclusion in the opposite direction:

```
In[34]:= Map[equal[0, composite[Id, complement[#]]] &,
           SubstTest[class, pair[x, y], implies[and[member[x, u], member[y, v]],
             member[dif[x, y], u]], {u → image[Q, set[omega]], v → FINITE}]] // Reverse

Out[34]= subclass[image[DIF, cart[image[Q, set[omega]], FINITE]], image[Q, set[omega]]] == True

In[35]:= % /. Equal → SetDelayed
```

Combining the two inclusions yields an equation:

```
In[37]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> image[Q, set[omega]], v -> image[DIF, cart[image[Q, set[omega]], FINITE]]}]

Out[37]= True == equal[image[DIF, cart[image[Q, set[omega]], FINITE]], image[Q, set[omega]]]

In[38]:= image[DIF, cart[image[Q, set[omega]], FINITE]] := image[Q, set[omega]]
```