relating binclosed[CAP] and binclosed[CUP] via RC[x]

Johan G. F. Belinfante
2004 January 13

```
In[1]:= << goedel53.11c; << tools.m

        :Package Title: goedel53.11c      2004 January 11 at 10:25 p.m.

        It is now:  2004 Jan 14 at 13:51

        Loading Simplification Rules

        TOOLS.M                       Revised 2004 January 3

        weightlimit = 40
```

summary

The relative complementation process takes binary unions to binary intersections:

```
In[2]:= composite[RC[x], CUP]

Out[2]= composite[CAP, cross[RC[x], RC[x]]]
```

In this notebook, this result is used to show that collections closed under binary intersections are mapped to collections closed under binary unions, and conversely. These results are formulated with and without variables.

implication in one direction

The implication in one direction is easy:

```
In[3]:= Map[implies[subclass[image[CUP, cart[x, x]], x], subclass[#, image[RC[y], x]]] &,
          ImageComp[RC[y], CUP, cart[x, x]]]

Out[3]= or[not[subclass[image[CUP, cart[x, x]], x]],
           subclass[image[CAP, cart[image[RC[y], x], image[RC[y], x]]], image[RC[y], x]]] == True

In[4]:= or[not[subclass[image[CUP, cart[x_, x_]], x_]], subclass[
            image[CAP, cart[image[RC[y_], x_], image[RC[y_], x_]]], image[RC[y_], x_]]] := True
```

One of the two variables is easily eliminated:

```
In[5]:= Map[equal[V, #] &, SubstTest[class, y, implies[member[y, u], member[image[w, y], v]],
             {u -> binclosed[CUP], v -> binclosed[CAP], w -> RC[x]}]] // Reverse

Out[5]= subclass[image[IMAGE[RC[x]], binclosed[CUP]], binclosed[CAP]] == True

In[6]:= subclass[image[IMAGE[RC[x_]], binclosed[CUP]], binclosed[CAP]] := True
```

The other variable can also be eliminated:

```
In[7]:= Map[equal[0, #] &, SubstTest[reify, x,
          dif[image[F[x], binclosed[CUP]], binclosed[CAP]], F[x] -> IMAGE[RC[x]]]] // Reverse

Out[7]= subclass[image[IMG, cart[range[RCF], binclosed[CUP]]], binclosed[CAP]] == True

In[8]:= subclass[image[IMG, cart[range[RCF], binclosed[CUP]]], binclosed[CAP]] := True
```

implication in the other direction

The same basic idea can be used in the other direction, but a few additional steps are needed to finish the job.

```
In[9]:= Map[implies[subclass[image[CAP, cart[x, x]], x], subclass[#, x]] &,
          ImageComp[RC[y], CUP, cart[image[RC[y], x], image[RC[y], x]]]] // Reverse

Out[9]= or[not[subclass[image[CAP, cart[x, x]], x]],
          subclass[image[RC[y], image[CUP, cart[image[RC[y], x], image[RC[y], x]]]], x]] == True

In[10]:= (% /. {x -> x_, y -> y_}) /. Equal -> SetDelayed
```

Lemma.

```
In[11]:= equal[intersection[image[CUP, cart[image[RC[u], v], image[RC[u], w]]], P[u]],
         image[CUP, cart[image[RC[u], v], image[RC[u], w]]]]

Out[11]= True

In[12]:= intersection[image[CUP, cart[image[RC[u_], v_], image[RC[u_], w_]]], P[u_]] :=
          image[CUP, cart[image[RC[u], v], image[RC[u], w]]]
```

Lemma.

```
In[13]:= SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
            {u -> image[RC[y], image[CUP, cart[image[RC[y], x], image[RC[y], x]]]],
           v -> x, w -> RC[y]}]

Out[13]= or[not[subclass[image[RC[y], image[CUP, cart[image[RC[y], x], image[RC[y], x]]]], x]],
          subclass[image[CUP, cart[image[RC[y], x], image[RC[y], x]]], image[RC[y], x]]] == True

In[14]:= (% /. {x -> x_, y -> y_}) /. Equal -> SetDelayed
```

The final result:

```
In[15]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], not[implies[p1, p3]],
                {p1 -> subclass[image[CAP, cart[x, x]], x],
           p2 -> subclass[image[RC[y], image[CUP, cart[image[RC[y], x], image[RC[y], x]]]], x],
                 p3 -> subclass[image[CUP, cart[image[RC[y], x], image[RC[y], x]]],
             image[RC[y], x]]}]]

Out[15]= or[not[subclass[image[CAP, cart[x, x]], x]],
           subclass[image[CUP, cart[image[RC[y], x], image[RC[y], x]]], image[RC[y], x]]] == True

In[16]:= or[not[subclass[image[CAP, cart[x_, x_]], x_]], subclass[
            image[CUP, cart[image[RC[y_], x_], image[RC[y_], x_]]], image[RC[y_], x_]]] := True
```

The elimination of variables proceeds as before:

In[17]:=
```
Map[equal[V, #] &, SubstTest[class, y, implies[member[y, u], member[image[w, y], v]],
    {u -> binclosed[CAP], v -> binclosed[CUP], w -> RC[x]}]] // Reverse
```

Out[17]=
```
subclass[image[IMAGE[RC[x]], binclosed[CAP]], binclosed[CUP]] == True
```

In[18]:=
```
subclass[image[IMAGE[RC[x_]], binclosed[CAP]], binclosed[CUP]] := True
```

In[19]:=
```
Map[equal[0, #] &, SubstTest[reify, x,
   dif[image[F[x], binclosed[CAP]], binclosed[CUP]], F[x] -> IMAGE[RC[x]]]] // Reverse
```

Out[19]=
```
subclass[image[IMG, cart[range[RCF], binclosed[CAP]]], binclosed[CUP]] == True
```

In[20]:=
```
subclass[image[IMG, cart[range[RCF], binclosed[CAP]]], binclosed[CUP]] := True
```