unions of chains of binary-fixed sets

Johan G. F. Belinfante
2010 July 26

In[1]:= **SetDirectory["l:"]; << goedel.10jul24a;<< tools.m**

```
:Package Title: goedel.10jul24a              2010 July 24 at 3:30 p.m.

It is now:  2010 Jul 26 at 8:17

Loading Simplification Rules

TOOLS.M                                     Revised 2010 July 24

weightlimit = 40
```

introduction and summary

In dealing with a binary operation **b ∈ map[s × s, s]** on a set **s**, it is standard practice to use the shorthand **u v** for the result of applying **b** to a pair of elements **u, v ∈ s** instead of the more correct notation **APPLY[b, pair[u, v]]**. For subsets **x, y ⊂ s**, the same shorthand notation **x y** is commonly used for the set of all elements obtained by applying **b** to all possible pairs of elements **u ∈ x** and **v ∈ y**. A more correct notation for this set is **image[b, x × y]**. The set **x y** can also be viewed as the result of applying the binary operation **IMAGE[b] ∘ CART** to **pair[x, y]**.

A subset **x ⊂ s** is said to be **closed** under the binary operation **b** if **x x ⊂ x**. The class of sets closed under a binary operation is closed under unions of chains. In this notebook sets satisfying the conditions **x x = x** and **x ⊂ x x** are considered. The class of sets satisfying the latter condition is shown to be closed under arbitrary unions. In general the intersection of two classes closed under unions of chains is again closed under unions of chains. It therefore follows that the class of sets satisfying the condition **x x = x** has this property. Another way to say this is that the set of idempotent elements for the binary operation **IMAGE[b] ∘ CART** is closed under unions of chains.

definitions

It is now time to be more precise. Attention need not be restricted to binary operations. In general one one can consider arbitrary classes. A class **y** is said to be **binary closed** under **x** if **image[x, y × y] ⊂ y**. The class of all sets that are binary closed under **x** is called **binclosed[x]**. This class is closed under arbitrary intersections.

In[2]:= **fix[HULL[binclosed[x]]]**

Out[2]= binclosed[x]

The class **binclosed[x]** is also closed under unions of chains.

```
In[3]:=  Uchains[binclosed[x]]

Out[3]=  binclosed[x]
```

A class **y** is said to be **binary fixed** under **x** if the stronger condition **image[x, y × y] = y** holds. The class of all sets that are binary fixed by **x** is **fix[IMAGE[x] ∘ CART ∘ DUP]**. The class of sets **y** that satisfy **y ⊂ image[x, y × y]** is:

```
In[4]:=  class[y, subclass[y, image[x, cart[y, y]]]]

Out[4]=  complement[fix[composite[E, UB[complement[x]], CART, DUP]]]
```

When **x** is thin, this expression is rewritten as follows:

```
In[5]:=  complement[fix[composite[E, UB[complement[thinpart[x]]], CART, DUP]]]

Out[5]=  fix[composite[inverse[S], IMAGE[thinpart[x]], CART, DUP]]
```

It will now be shown that the intersection of this class with **binclosed[x]** is the class of binary fixed sets.

Lemma.

```
In[6]:=  symdif[intersection[binclosed[x],
            complement[fix[composite[E, UB[complement[x]], CART, DUP]]]],
           fix[composite[IMAGE[x], CART, DUP]]] // Normality

Out[6]=  union[intersection[complement[binclosed[x]], fix[composite[IMAGE[x], CART, DUP]]],
           intersection[fix[composite[IMAGE[x], CART, DUP]],
            fix[composite[E, UB[complement[x]], CART, DUP]]],
           intersection[binclosed[x], complement[fix[composite[IMAGE[x], CART, DUP]]],
            complement[fix[composite[E, UB[complement[x]], CART, DUP]]]]] == 0

In[7]:=  (% /. x → x_) /. Equal → SetDelayed
```

Theorem.

```
In[8]:=  SubstTest[empty, symdif[u, v], {u -> intersection[binclosed[x],
             complement[fix[composite[E, UB[complement[x]], CART, DUP]]]],
           v -> fix[composite[IMAGE[x], CART, DUP]]}]

Out[8]=  equal[fix[composite[IMAGE[x], CART, DUP]], intersection[binclosed[x],
            complement[fix[composite[E, UB[complement[x]], CART, DUP]]]]] == True

In[9]:=  intersection[binclosed[x_],
           complement[fix[composite[E, UB[complement[x_]], CART, DUP]]]] :=
          fix[composite[IMAGE[x], CART, DUP]]
```

Comment. There is also a related result that coincides with the above when **x** is thin:

```
In[12]:=  intersection[binclosed[x], fix[composite[inverse[S], IMAGE[x], CART, DUP]]]

Out[12]=  fix[composite[IMAGE[x], CART, DUP]]
```

unions of chains of binary fixed sets

Lemma.

```
In[13]:= Map[or[#, subclass[u, image[x, cart[u, u]]]] &,
           SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
            w → complement[fix[composite[E, UB[complement[x]], CART, DUP]]]]] // Reverse

Out[13]= or[not[equal[0, intersection[v, fix[composite[E, UB[complement[x]], CART, DUP]]]]],
           not[member[u, v]], subclass[u, image[x, cart[u, u]]]] == True

In[14]:= (% /. {u → u_, v → v_, x → x_}) /. Equal → SetDelayed
```

Theorem.

```
In[15]:= SubstTest[implies, subclass[u, v],
           subclass[image[x, u], image[x, v]], {u → cart[y, y], v → cart[z, z]}] // Reverse

Out[15]= or[not[subclass[y, z]], subclass[image[x, cart[y, y]], image[x, cart[z, z]]]] == True

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

Theorem.

```
In[17]:= Map[not, SubstTest[and, implies[and[p1, p2], p3], implies[p1, p4], implies[p4, p5],
            implies[and[p3, p5], p6], not[implies[and[p1, p2], p6]], {p1 → member[u, v],
             p2 → equal[0, intersection[v, fix[composite[E, UB[complement[x]], CART, DUP]]]],
             p3 → subclass[u, image[x, cart[u, u]]], p4 → subclass[u, U[v]],
             p5 → subclass[image[x, cart[u, u]], image[x, cart[U[v], U[v]]]],
             p6 → subclass[u, image[x, cart[U[v], U[v]]]]}]] // Reverse

Out[17]= or[not[equal[0, intersection[v, fix[composite[E, UB[complement[x]], CART, DUP]]]]],
           not[member[u, v]], subclass[u, image[x, cart[U[v], U[v]]]]] == True

In[18]:= (% /. {u → u_, v → v_, x → x_}) /. Equal → SetDelayed
```

Lemma.

```
In[19]:= Map[equal[V, #] &, SubstTest[class, u,
           or[not[subclass[v, t]], not[member[u, v]], subclass[u, image[x, cart[U[v], U[v]]]]],
           t → complement[fix[composite[E, UB[complement[x]], CART, DUP]]]]]

Out[19]= or[not[equal[0, intersection[v, fix[composite[E, UB[complement[x]], CART, DUP]]]]],
           subclass[U[v], image[x, cart[U[v], U[v]]]]] == True

In[20]:= (% /. {v → v_, x → x_}) /. Equal → SetDelayed
```

Lemma.

```
In[21]:= (Map[equal[V, #] &, SubstTest[class, v,
            implies[subclass[v, t], subclass[U[v], image[u, cart[U[v], U[v]]]]], t →
             complement[fix[composite[E, UB[complement[u]], CART, DUP]]]]]) /. u → complement[x]

Out[21]= equal[0, intersection[fix[composite[E, UB[x], CART, DUP]],
            Uclosure[complement[fix[composite[E, UB[x], CART, DUP]]]]]] == True

In[22]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. The complement of the class **fix[E ∘ UB[x] ∘ CART ∘ DUP]** is closed under arbitrary unions.

```
In[23]:= SubstTest[implies, subclass[Uclosure[t], t], equal[Uclosure[t], t],
           t -> complement[fix[composite[E, UB[x], CART, DUP]]]] // Reverse

Out[23]= equal[complement[fix[composite[E, UB[x], CART, DUP]]],
           Uclosure[complement[fix[composite[E, UB[x], CART, DUP]]]]] == True

In[24]:= Uclosure[complement[fix[composite[E, UB[x_], CART, DUP]]]] :=
          complement[fix[composite[E, UB[x], CART, DUP]]]
```

Corollary. The complement of the class **fix[E ∘ UB[x] ∘ CART ∘ DUP]** is closed under unions of chains.

```
In[25]:= SubstTest[Uchains, Uclosure[t],
           t -> complement[fix[composite[E, UB[x], CART, DUP]]]] // Reverse

Out[25]= Uchains[complement[fix[composite[E, UB[x], CART, DUP]]]] ==
          complement[fix[composite[E, UB[x], CART, DUP]]]

In[26]:= Uchains[complement[fix[composite[E, UB[x_], CART, DUP]]]] :=
          complement[fix[composite[E, UB[x], CART, DUP]]]
```

Theorem. The class of binary fixed sets is closed under unions of chains.

```
In[27]:= SubstTest[implies, and[equal[Uchains[u], u], equal[Uchains[v], v]],
           equal[Uchains[intersection[u, v]], intersection[u, v]], {u → binclosed[x],
            v → complement[fix[composite[E, UB[complement[x]], CART, DUP]]]}] // Reverse

Out[27]= equal[fix[composite[IMAGE[x], CART, DUP]],
           Uchains[fix[composite[IMAGE[x], CART, DUP]]]] == True

In[28]:= Uchains[fix[composite[IMAGE[x_], CART, DUP]]] := fix[composite[IMAGE[x], CART, DUP]]
```

Comment. For the special case **x = SWAP ∘ RIF**, this reduces to the statement **Uchains[IDEM] = IDEM**, a result that is already available in the **GOEDEL** program.

```
In[30]:= fix[composite[IMAGE[x], CART, DUP]] /. x → composite[SWAP, RIF]

Out[30]= IDEM
```

related Uclosure results

In this final section, some additional rewrite rules are derived that are similar to the **Uclosure** rule derived earlier. For example, one can obviously replace the upper bound relation **UB[x]** with the lower bound relation **LB[x]**.

Theorem. The complement of the class **fix[E ∘ LB[x] ∘ CART ∘ DUP]** is closed under arbitrary unions.

```
In[31]:= SubstTest[Uclosure,
           complement[fix[composite[E, UB[t], CART, DUP]]], t → inverse[x]] // Reverse

Out[31]= Uclosure[complement[fix[composite[E, LB[x], CART, DUP]]]] ==
          complement[fix[composite[E, LB[x], CART, DUP]]]

In[32]:= Uclosure[complement[fix[composite[E, LB[x_], CART, DUP]]]] :=
          complement[fix[composite[E, LB[x], CART, DUP]]]
```

If **x** is a thin relation, the class of sets satisfying **y ⊂ image[x, y × y]** is rewritten.

Theorem.

```
In[33]:= SubstTest[Uclosure, complement[fix[composite[E, UB[t], CART, DUP]]],
           t → complement[thinpart[x]]] // Reverse

Out[33]= Uclosure[fix[composite[inverse[S], IMAGE[thinpart[x]], CART, DUP]]] ==
          fix[composite[inverse[S], IMAGE[thinpart[x]], CART, DUP]]

In[34]:= Uclosure[fix[composite[inverse[S], IMAGE[thinpart[x_]], CART, DUP]]] :=
          fix[composite[inverse[S], IMAGE[thinpart[x]], CART, DUP]]
```

In particular, this is the case for any binary operation.

Corollary.

```
In[35]:= SubstTest[Uclosure,
           fix[composite[inverse[S], IMAGE[thinpart[t]], CART, DUP]], t → binop[x]] // Reverse

Out[35]= Uclosure[fix[composite[inverse[S], IMAGE[binop[x]], CART, DUP]]] ==
          fix[composite[inverse[S], IMAGE[binop[x]], CART, DUP]]

In[36]:= Uclosure[fix[composite[inverse[S], IMAGE[binop[x_]], CART, DUP]]] :=
          fix[composite[inverse[S], IMAGE[binop[x]], CART, DUP]]
```