cliques of an equivalence relation

Johan G. F. Belinfante
2014 April 26

In[1]:=
```
SetDirectory["l:"]; << goedel.14apr24a
```

```
:Package Title: goedel.14apr24a                    2014 April 24 at 5:40 p.m.

Loading takes about seventeen minutes, half that time due to builtin pauses.

It is now:  2014 Apr 26 at 17:23

Loading Simplification Rules

TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

weightlimit = 40

Loading completed.

It is now:  2014 Apr 26 at 17:40
```

summary

If a collection of cliques of an equivalence relation has a point in common, then its sum class is a clique.

derivation

Lemma.

In[39]:=
```
Map[not, SubstTest[and, implies[p2, p3],
   implies[and[p1, p3], p4], not[implies[and[p1, p2], p4]],
   {p1 -> member[pair[u, w], eqv[x]], p2 -> member[pair[v, w], eqv[x]],
    p3 -> member[pair[w, v], eqv[x]], p4 -> member[pair[u, v], eqv[x]]}]] // Reverse
```

Out[39]=
```
or[member[pair[u, v], eqv[x]],
  not[member[pair[u, w], eqv[x]]], not[member[pair[v, w], eqv[x]]]] == True
```

In[41]:=
```
or[member[pair[u_, v_], eqv[x_]],
  not[member[pair[u_, w_], eqv[x_]]], not[member[pair[v_, w_], eqv[x_]]]] := True
```

Theorem.

```
In[26]:= Map[not, SubstTest[and, implies[and[p1, p4], p7], implies[and[p2, p4], p8],
           (*implies[and[p3,p7,p8],p9],*) not[implies[and[p1, p2, p3, p4], p9]],
           {p1 → member[v, A[w]], p2 → subclass[w, cliques[eqv[x]]],
            p3 → member[u, t], p4 -> member[t, w], p7 → member[v, t],
            p8 → subclass[cart[t, t], eqv[x]], p9 → member[pair[u, v], eqv[x]]}]] // Reverse

Out[26]= or[member[pair[u, v], eqv[x]], not[member[t, w]], not[member[u, t]],
          not[member[v, A[w]]], not[subclass[w, cliques[eqv[x]]]]] == True

In[28]:= or[member[pair[u_, v_], eqv[x_]], not[member[t_, w_]], not[member[u_, t_]],
          not[member[v_, A[w_]]], not[subclass[w_, cliques[eqv[x_]]]]] := True
```

Corollary. (Eliminate the variable **t**.)

```
In[32]:= Map[equal[V, domain[#]] &, SubstTest[reify, t,
          case[or[member[p, q], not[member[t, w]], not[member[u, t]], not[member[v, A[w]]],
            not[subclass[w, z]]]], {p → pair[u, v], q → eqv[x], z -> cliques[eqv[x]]}]]

Out[32]= or[member[pair[u, v], eqv[x]], not[member[u, U[w]]],
          not[member[v, A[w]]], not[subclass[w, cliques[eqv[x]]]]] == True

In[34]:= or[member[pair[u_, v_], eqv[x_]], not[member[u_, U[w_]]],
          not[member[v_, A[w_]]], not[subclass[w_, cliques[eqv[x_]]]]] := True
```

Lemma.

```
In[45]:= Map[not, SubstTest[and, implies[and[p1, p2, p3], p5], implies[and[p1, p2, p4], p6],
           implies[and[p5, p6], p7], not[implies[and[p1, p2, p3, p4], p7]],
           {p1 → member[t, A[w]], p2 → subclass[w, cliques[eqv[x]]],
            p3 → member[u, U[w]], p4 → member[v, U[w]], p5 → member[pair[u, t], eqv[x]],
            p6 -> member[pair[v, t], eqv[x]], p7 → member[pair[u, v], eqv[x]]}]] // Reverse

Out[45]= or[member[pair[u, v], eqv[x]], not[member[t, A[w]]], not[member[u, U[w]]],
          not[member[v, U[w]]], not[subclass[w, cliques[eqv[x]]]]] == True

In[46]:= (% /. {t → t_, u → u_, v → v_, w → w_, x → x_}) /. Equal → SetDelayed
```

Lemma. (Eliminate the variables **t**, **u** and **v**.)

```
In[48]:= Map[empty[composite[complement[#], id[cart[V, V]]]] &,
         SubstTest[class, pair[pair[u, v], t], or[member[pair[u, v], q],
           not[member[t, y]], not[member[pair[u, v], z]], not[subclass[w, s]]],
          {q → eqv[x], s → cliques[eqv[x]], y → A[w], z → cart[U[w], U[w]]}]]

Out[48]= or[equal[0, A[w]], not[subclass[w, cliques[eqv[x]]]],
          subclass[cart[U[w], U[w]], eqv[x]]] == True

In[49]:= or[equal[0, A[w_]], not[subclass[w_, cliques[eqv[x_]]]],
          subclass[cart[U[w_], U[w_]], eqv[x_]]] := True
```

Restatement.

```
In[50]:= implies[and[member[w, P[cliques[eqv[x]]]], not[empty[A[w]]]],
          member[U[w], cliques[eqv[x]]]]

Out[50]= True
```

Theorem. (Eliminate the variable **w**.)

```
In[52]:= Map[equal[V, #] &, SubstTest[class, w, implies[
            and[member[w, P[t]], not[empty[A[w]]]], member[U[w], t]], t → cliques[eqv[x]]]]

Out[52]= subclass[image[BIGCUP, intersection[complement[image[inverse[BIGCAP], set[0]]],
             P[cliques[eqv[x]]]]], cliques[eqv[x]]] == True

In[53]:= subclass[image[BIGCUP, intersection[complement[image[inverse[BIGCAP], set[0]]],
             P[cliques[eqv[x_]]]]], cliques[eqv[x_]]] := True
```

Comment. The class **cliques[eqv[x]]** is not closed under arbitrary unions except in the trivial case that **eqv[x]** is a cartesian square.

```
In[54]:= subclass[Uclosure[cliques[eqv[x]]], cliques[eqv[x]]]

Out[54]= subclass[cart[fix[eqv[x]], fix[eqv[x]]], eqv[x]]
```

Restatement.

```
In[55]:= member[cliques[eqv[setpart[x]]],
          allclosed[composite[BIGCUP, id[complement[image[inverse[BIGCAP], set[0]]]]]]]

Out[55]= True
```

Theorem. Eliminate the variable **x**.

```
In[60]:= Map[subclass[image[CLIQUES, EQV], image[composite[CLIQUES, EQUIV], domain[#]]] &,
          SubstTest[reify, x, case[member[cliques[eqv[setpart[x]]], w]],
           w -> allclosed[composite[BIGCUP, id[complement[image[inverse[BIGCAP], set[0]]]]]]]]

Out[60]= subclass[image[CLIQUES, EQV],
           allclosed[composite[BIGCUP, id[complement[image[inverse[BIGCAP], set[0]]]]]]] == True

In[61]:= subclass[image[CLIQUES, EQV],
           allclosed[composite[BIGCUP, id[complement[image[inverse[BIGCAP], set[0]]]]]]] := True
```

Comment. The above result could be viewed as a partial converse for the following result.

```
In[62]:= implies[member[x,
           allclosed[composite[BIGCUP, id[complement[image[inverse[BIGCAP], set[0]]]]]]],
          EQUIVALENCE[composite[inverse[E], id[x], E]]]

Out[62]= True
```