maximal cliques for equivalence relations

Johan G. F. Belinfante
2011 October 4

```
In[1]:= SetDirectory["l:"]; << goedel.11oct04a

:Package Title: goedel.11oct04a                2011 October 4 at 12:45 noon

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

It is now:  2011 Oct 4 at 13:51

Loading Simplification Rules

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

weightlimit = 40

Loading completed.

It is now:  2011 Oct 4 at 14:3
```

summary

Every clique of a thin equivalence relation is a subset of a maximal clique.

derivation

The proof is not complicated, but some fussy details need to be addressed. A brief sketch of the idea behind the proof may perhaps be helpful. Let **y** be a non-empty clique of an equivalence relation **eqv[x]**, and let **w ∈ y**. The vertical section **z = image[eqv[x], {w}]** is the equivalence class of **w**. One has **y ⊂ z** and **z** is a maximal clique. Thus, if the clique **y** is not empty, it is a subclass of a maximal clique. If **y = 0**, then **y** is a subclass of any maximal clique, and for any small equivalence relation there always is at least one maximal clique. If **eqv[x] = 0** then **0** is a maximal clique.

Theorem. Equivalence classes are maximal cliques.

```
In[2]:= Map[equal[V, domain[#]] &,
         SubstTest[reify, y, case[or[equal[y, image[t, set[w]]], not[member[w, fix[t]]],
            not[subclass[cart[y, y], t]], not[subclass[image[t, set[w]], y]]]], t → eqv[x]]]

Out[2]= or[not[member[w, fix[eqv[x]]]],
          not[member[image[eqv[x], set[w]], image[inverse[PS], cliques[eqv[x]]]]]] == True

In[3]:= or[not[member[w_, fix[eqv[x_]]]],
          not[member[image[eqv[x_], set[w_]], image[inverse[PS], cliques[eqv[x_]]]]]] := True
```

Lemma.

```
In[4]:= SubstTest[implies, and[member[u, v], subclass[v, x]],
          member[u, x], {u → pair[w, w], v → cart[y, y]}] // Reverse

Out[4]= or[member[pair[w, w], x], not[member[w, y]], not[subclass[cart[y, y], x]]] == True

In[5]:= (% /. {w → w_, x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. Every member of a clique of **x** is a fixed point of **x**.

```
In[6]:= Map[not, SubstTest[and, implies[p1, p2],
            not[implies[p1, p3]], {p1 → and[member[w, y], subclass[cart[y, y], x]],
             p2 -> member[pair[w, w], x], p3 → member[w, fix[x]]}]] // Reverse

Out[6]= or[member[w, fix[x]], not[member[w, y]], not[subclass[cart[y, y], x]]] == True

In[7]:= or[member[w_, fix[x_]], not[member[w_, y_]], not[subclass[cart[y_, y_], x_]]] := True
```

Corollary. The equivalence class of any member of a clique is a maximal clique.

```
In[8]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3],
            not[implies[p1, p3]], {p1 → and[member[w, y], subclass[cart[y, y], eqv[x]]],
             p2 → member[w, fix[eqv[x]]], p3 → not[
               member[image[eqv[x], set[w]], image[inverse[PS], cliques[eqv[x]]]]]}]] // Reverse

Out[8]= or[not[member[w, y]],
          not[member[image[eqv[x], set[w]], image[inverse[PS], cliques[eqv[x]]]]],
          not[subclass[cart[y, y], eqv[x]]]] == True

In[9]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Lemma.

```
In[10]:= SubstTest[implies, and[member[z, t], subclass[y, z]],
            member[y, image[inverse[S], t]], {t -> maximal[S, cliques[eqv[thinpart[x]]]],
             z → image[eqv[thinpart[x]], set[w]]}] // Reverse

Out[10]= or[member[image[eqv[thinpart[x]], set[w]],
             image[inverse[PS], cliques[eqv[thinpart[x]]]]],
            not[subclass[y, image[eqv[thinpart[x]], set[w]]]],
            not[subclass[intersection[cliques[eqv[thinpart[x]]], image[S, set[y]]],
              image[inverse[PS], cliques[eqv[thinpart[x]]]]]]] == True

In[11]:= (% /. {w → w_, x → x_, y → y_}) /. Equal → SetDelayed
```

Main Theorem. Every clique of a thin equivalence relation is a subset of a maximal clique.

```
In[12]:= Map[equal[V, domain[reify[y, case[#]]]] &,
           Map[equal[V, domain[reify[w, case[#]]]] &, Map[not, SubstTest[and,
              implies[and[p1, p2], p3], implies[and[p1, p2], p4], (* implies[and[p3,p4],p5], *)
              not[implies[and[p1, p2], p5]], {p1 → subclass[cart[y, y], eqv[thinpart[x]]],
               p2 → member[w, y], p3 → subclass[y, image[eqv[thinpart[x]], set[w]]],
               p4 → not[member[image[eqv[thinpart[x]], set[w]],
                  image[inverse[PS], cliques[eqv[thinpart[x]]]]]],
               p5 → not[subclass[intersection[cliques[eqv[thinpart[x]]], image[S, set[y]]],
                  image[inverse[PS], cliques[eqv[thinpart[x]]]]]]}]]]] // Reverse

Out[12]= subclass[cliques[eqv[thinpart[x]]],
           image[inverse[S], intersection[cliques[eqv[thinpart[x]]],
             complement[image[inverse[PS], cliques[eqv[thinpart[x]]]]]]]] == True

In[13]:= subclass[cliques[eqv[thinpart[x_]]],
           image[inverse[S], intersection[cliques[eqv[thinpart[x_]]],
             complement[image[inverse[PS], cliques[eqv[thinpart[x_]]]]]]]] := True
```

Corollary. (Restatement without the **eqv** and **setpart** wrappers.)

```
In[14]:= SubstTest[implies, equal[x, eqv[thinpart[t]]],
           subclass[cliques[x], image[inverse[S], maximal[S, cliques[x]]]], t → x] // Reverse

Out[14]= or[not[equal[V, domain[VERTSECT[x]]]],
           not[EQUIVALENCE[x]], subclass[cliques[x], image[inverse[S],
             intersection[cliques[x], complement[image[inverse[PS], cliques[x]]]]]]] == True

In[15]:= or[not[equal[V, domain[VERTSECT[x_]]]],
           not[EQUIVALENCE[x_]], subclass[cliques[x_], image[inverse[S],
             intersection[cliques[x_], complement[image[inverse[PS], cliques[x_]]]]]]] := True
```

Counterexample. One can not omit the thinness hypothesis.

```
In[16]:= or[not[EQUIVALENCE[x]],
           subclass[cliques[x], image[inverse[S], intersection[cliques[x],
              complement[image[inverse[PS], cliques[x]]]]]]] /. x → cart[V, V]

Out[16]= False
```

Theorem. (Variable-free statement.)

```
In[17]:= Map[equal[V, domain[#]] &, SubstTest[reify, x, case[subclass[cliques[eqv[thinpart[x]]],
             image[inverse[t], maximal[S, cliques[eqv[thinpart[x]]]]]]], t → S]]

Out[17]= subclass[composite[inverse[E], id[image[CLIQUES, EQV]]],
           composite[inverse[S], MAXIMAL[S]]] == True

In[18]:= subclass[composite[inverse[E], id[image[CLIQUES, EQV]]],
           composite[inverse[S], MAXIMAL[S]]] := True
```

Note that this theorem closely resembles the following famous equivalent of the axiom of choice, which states that **axch** is equivalent to the assertion that any clique of any small relation is a subset of a maximal clique.

```
In[19]:= subclass[composite[inverse[E], id[range[CLIQUES]]], composite[inverse[S], MAXIMAL[S]]]

Out[19]= axch
```

As has been shown above, the statement that every clique of a thin equivalence relation is a subset of a maximal clique does not require the axiom of choice.