maximal cliques

Johan G. F. Belinfante
2007 November 25

```
In[1]:= SetDirectory["l:"]; << goedel99.23b; << tools.m

        :Package Title: goedel99.23b                    2007 November 23 at 11:05 p.m.

        It is now:  2007 Nov 25 at 5:41

        Loading Simplification Rules

        TOOLS.M                                  Revised 2007 September 19

        weightlimit = 40
```

summary

It is shown in this notebook that the axiom of choice is equivalent to the statement that there are maximal cliques for any small relation. The axiom of choice is also shown to be equivalent with the stronger statement that for any small relation, every clique is contained in a maximal clique.

derivation

Theorem.

```
In[2]:= SubstTest[implies, and[subclass[u, v], disjoint[v, w]], disjoint[u, w],
          {u → range[CLIQUES], v → fix[UCHAINS], w → subvar[inverse[PS]]}] // Reverse

Out[2]= or[equal[0, intersection[range[CLIQUES], subvar[inverse[PS]]]], not[axch]] == True

In[3]:= % /. Equal → SetDelayed
```

In practice, it is convenient to introduce a variable:

```
In[8]:= SubstTest[implies, and[member[t, u], implies[p, disjoint[u, v]]],
          implies[p, not[member[t, v]]], {p → axch, t → cliques[setpart[x]],
           u → range[CLIQUES], v → subvar[inverse[PS]]}] // Reverse

Out[8]= or[not[axch],
          not[subclass[cliques[setpart[x]], image[inverse[PS], cliques[setpart[x]]]]]] == True

In[9]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. The axiom of choice implies that for any set **x**, the set **cliques[x]** has a maximal element.

```
In[11]:= Map[implies[member[x, y], #] &, SubstTest[implies, equal[x, setpart[t]], or[not[axch],
           not[subclass[cliques[x], image[inverse[PS], cliques[x]]]]], t → x]] // Reverse

Out[11]= or[not[axch], not[member[x, y]],
           not[subclass[cliques[x], image[inverse[PS], cliques[x]]]]] == True

In[12]:= or[not[axch], not[member[x_, y_]],
           not[subclass[cliques[x_], image[inverse[PS], cliques[x_]]]]] := True
```

Corollary. The axiom of choice implies that for any set **x**, the set **chains[x]** has a maximal element.

```
In[15]:= Map[implies[member[x, y], #] &, SubstTest[or, not[axch],
           not[member[t, V]], not[subclass[cliques[t], image[inverse[PS], cliques[t]]]],
           t → union[x, inverse[x]]]] // Reverse

Out[15]= or[not[axch], not[member[x, y]],
           not[subclass[chains[x], image[inverse[PS], chains[x]]]]] == True

In[16]:= or[not[axch], not[member[x_, y_]],
           not[subclass[chains[x_], image[inverse[PS], chains[x_]]]]] := True
```

negative forms

Theorem. (A negative form for a special case of Zorn's lemma.)

```
In[34]:= and[axch, equal[x, Uchains[x]], member[x, y],
           subclass[x, image[inverse[PS], x]]] // NotNotTest

Out[34]= and[axch, equal[x, Uchains[x]],
           member[x, y], subclass[x, image[inverse[PS], x]]] == False

In[35]:= and[axch, equal[x_, Uchains[x_]],
           member[x_, y_], subclass[x_, image[inverse[PS], x_]]] := False
```

Theorem.

```
In[40]:= and[axch, member[x, y],
           subclass[cliques[x], image[inverse[PS], cliques[x]]]] // NotNotTest

Out[40]= and[axch, member[x, y], subclass[cliques[x], image[inverse[PS], cliques[x]]]] == False

In[41]:= and[axch, member[x_, y_],
           subclass[cliques[x_], image[inverse[PS], cliques[x_]]]] := False
```

Theorem.

```
In[42]:= and[axch, member[x, y],
           subclass[chains[x], image[inverse[PS], chains[x]]]] // NotNotTest

Out[42]= and[axch, member[x, y], subclass[chains[x], image[inverse[PS], chains[x]]]] == False
```

```
In[43]:= and[axch, member[x_, y_], subclass[chains[x_], image[inverse[PS], chains[x_]]]] := False
```

cross-sections as maximal cliques

Lemma.

```
In[17]:= AssInt[FUNS, P[cart[V, V]], P[x]]

Out[17]= intersection[FUNS, P[composite[Id, x]]] == intersection[FUNS, P[x]]

In[18]:= intersection[FUNS, P[composite[Id, x_]]] := intersection[FUNS, P[x]]
```

Technical Lemma.

```
In[19]:= AssInt[image[inverse[IMAGE[id[cart[V, V]]]], FUNS], P[cart[V, V]], P[x]]

Out[19]= intersection[image[inverse[IMAGE[id[cart[V, V]]]], FUNS], P[composite[Id, x]]] ==
          intersection[FUNS, P[x]]

In[20]:= intersection[image[inverse[IMAGE[id[cart[V, V]]]], FUNS], P[composite[Id, x_]]] :=
          intersection[FUNS, P[x]]
```

Lemma.

```
In[23]:= SubstTest[member, cliques[setpart[t]], range[CLIQUES], t -> intersection[
             cartsq[composite[Id, setpart[x]]], complement[cross[Id, Di]]]] // Reverse

Out[23]= member[intersection[FUNS, P[setpart[x]]], range[CLIQUES]] == True

        (% /. x → x_) /. Equal → SetDelayed
```

Theorem. Cross-sections of a class **x** are maximal elements of the class of functions contained in **x**.

```
In[25]:= SubstTest[empty, dif[u, v],
          {u -> intersection[FUNS, P[x]], v → image[inverse[PS], intersection[FUNS, P[x]]]}]

Out[25]= subclass[intersection[FUNS, P[x]],
           image[inverse[PS], intersection[FUNS, P[x]]]] == equal[0, X[x]]

In[26]:= subclass[intersection[FUNS, P[x_]],
           image[inverse[PS], intersection[FUNS, P[x_]]]] := equal[0, X[x]]
```

converse theorem

Lemma.

```
In[27]:= SubstTest[implies, and[member[t, u], disjoint[u, v]], not[member[t, v]],
           {t -> intersection[FUNS, P[setpart[x]]],
            u → range[CLIQUES], v → subvar[inverse[PS]]}] // Reverse

Out[27]= or[not[equal[0, intersection[range[CLIQUES], subvar[inverse[PS]]]]],
           not[equal[0, X[setpart[x]]]]] == True

         (% /. x → x_) /. Equal → SetDelayed
```

Converse Theorem.

```
In[29]:= Map[equal[V, #] &, SubstTest[class, x, implies[empty[u], member[setpart[x], v]],
           {u → intersection[range[CLIQUES], subvar[inverse[PS]]], v → SELECT}]]

Out[29]= or[axch, not[equal[0, intersection[range[CLIQUES], subvar[inverse[PS]]]]]] == True

In[30]:= % /. Equal → SetDelayed
```

Main Theorem. The axiom of choice is equivalent to the statement that **cliques[x]** has a maximal element for every set **x**.

```
In[32]:= equiv[equal[0, intersection[range[CLIQUES], subvar[inverse[PS]]]], axch]

Out[32]= True

In[33]:= equal[0, intersection[range[CLIQUES], subvar[inverse[PS]]]] := axch
```

---

a stronger version

Theorem. The axiom of choice implies that every clique of a small relation is contained in a maximal clique.

```
In[44]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u -> composite[inverse[E], id[range[CLIQUES]]], v ->
             composite[inverse[E], id[fix[composite[inverse[IMAGE[inverse[S]]], S, UCHAINS]]]],
            w -> composite[inverse[S], MAXIMAL[S]]}] // Reverse

Out[44]= or[not[axch], subclass[composite[inverse[E], id[range[CLIQUES]]],
            composite[inverse[S], MAXIMAL[S]]]] == True

In[45]:= % /. Equal → SetDelayed
```

Lemma.

```
In[46]:= Map[empty,
          dif[intersection[range[CLIQUES], subvar[inverse[PS]]], set[0]] // Renormality]

Out[46]= subclass[intersection[range[CLIQUES], subvar[inverse[PS]]], set[0]] == axch

In[47]:= % /. Equal → SetDelayed
```

Converse theorem. The statement that all cliques of small relations are contained in maximal cliques implies **axch**.

```
In[48]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
           {t → FIRST, u -> composite[inverse[E], id[range[CLIQUES]]],
            v -> composite[inverse[S], MAXIMAL[S]]}] // Reverse

Out[48]= or[axch, not[subclass[composite[inverse[E], id[range[CLIQUES]]],
             composite[inverse[S], MAXIMAL[S]]]]] == True

In[49]:= % /. Equal → SetDelayed
```

The theorem and its converse can be combined into a single rewrite rule.

```
In[50]:= equiv[subclass[composite[inverse[E], id[range[CLIQUES]]],
           composite[inverse[S], MAXIMAL[S]]], axch]

Out[50]= True

In[51]:= subclass[composite[inverse[E], id[range[CLIQUES]]],
           composite[inverse[S], MAXIMAL[S]]] := axch
```

introducing variables

A variable can be introduced as follows, using a **setpart** wrapper:

```
In[56]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
            {u → composite[inverse[E], id[set[cliques[setpart[x]]]]],
             v -> composite[inverse[E], id[range[CLIQUES]]],
             w -> composite[inverse[S], MAXIMAL[S]]}] // Reverse // MapNotNot

Out[56]= or[not[axch],
           subclass[cliques[setpart[x]], image[inverse[S], intersection[cliques[setpart[x]],
              complement[image[inverse[PS], cliques[setpart[x]]]]]]]] == True

In[57]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. If the axiom of choice holds, and if **x** is a set, then any clique in **x** is contained in a maximal clique.

```
In[59]:= Map[implies[member[x, y], #] &, SubstTest[implies, equal[x, setpart[t]],
           or[not[axch], subclass[cliques[x], image[inverse[S], intersection[cliques[x],
              complement[image[inverse[PS], cliques[x]]]]]], t → x]] // Reverse

Out[59]= or[not[axch], not[member[x, y]], subclass[cliques[x], image[inverse[S],
             intersection[cliques[x], complement[image[inverse[PS], cliques[x]]]]]] == True

In[60]:= or[not[axch], not[member[x_, y_]], subclass[cliques[x_], image[inverse[S],
             intersection[cliques[x_], complement[image[inverse[PS], cliques[x_]]]]]]] := True
```

Hausdorff's maximum principle

Theorem. If the axiom of choice holds, and if **x** is a set, then any chain in **x** is contained in a maximal chain.

```
In[62]:= Map[implies[member[x, y], #] &,
           SubstTest[implies, and[axch, member[t, V]], subclass[cliques[t], image[inverse[S],
              intersection[cliques[t], complement[image[inverse[PS], cliques[t]]]]]],
            t → union[x, inverse[x]]]] // Reverse

Out[62]= or[not[axch], not[member[x, y]], subclass[chains[x], image[inverse[S],
            intersection[chains[x], complement[image[inverse[PS], chains[x]]]]]] == True

In[66]:= or[not[axch], not[member[x_, y_]], subclass[chains[x_], image[inverse[S],
            intersection[chains[x_], complement[image[inverse[PS], chains[x_]]]]]] := True
```

Corollary. The axiom of choice implies Hausdorff's Maximum Principle: every chain of sets in a set **x**, ordered by inclusion, is contained in a maximal chain.

```
In[72]:= Map[implies[member[x, y], #] &,
           SubstTest[implies, and[axch, member[t, V]], subclass[chains[t], image[inverse[S],
              intersection[chains[t], complement[image[inverse[PS], chains[t]]]]]],
            t → composite[id[x], S, id[x]]]] // Reverse

Out[72]= or[not[axch], not[member[x, y]],
           subclass[intersection[chains[S], P[x]], image[inverse[S], intersection[chains[S],
              complement[image[inverse[PS], intersection[chains[S], P[x]]]], P[x]]]] == True

In[73]:= or[not[axch], not[member[x_, y_]],
           subclass[intersection[chains[S], P[x_]], image[inverse[S], intersection[chains[S],
              complement[image[inverse[PS], intersection[chains[S], P[x_]]]], P[x_]]]] := True
```