finite chains

Johan G. F. Belinfante
2004 February 2

```
In[1]:= << goedel53.29a; << tools.m

        :Package Title: goedel53.29a      2004 January 29 at 8:35 p.m.

        It is now:  2004 Feb 2 at 8:51

        Loading Simplification Rules

        TOOLS.M                      Revised 2004 January 3

        weightlimit = 40
```

summary

Any finite nonempty chain of subsets has a least member and a greatest member.

existence of a least member

Lemma 1.

```
In[2]:= Map[implies[member[y, x], #] &,
         SubstTest[implies, and[member[s, x], member[pair[s, y], composite[Id, w]]],
          member[y, image[w, x]], w -> PS]]

Out[2]= or[equal[s, y], member[y, image[PS, x]],
          not[member[s, x]], not[member[y, x]], not[subclass[s, y]]] == True

In[3]:= (% /. {s -> s_, x -> x_, y -> y_}) /. Equal -> SetDelayed
```

Lemma 2.

```
In[4]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
            {u -> pair[y, s], v -> cart[x, x], w -> union[S, inverse[S]]}] // MapNotNot

Out[4]= or[not[member[s, x]], not[member[y, x]],
          not[subclass[cart[x, x], union[S, inverse[S]]]], subclass[s, y], subclass[y, s]] == True

In[5]:= (% /. {s -> s_, x -> x_, y -> y_}) /. Equal -> SetDelayed
```

Lemma 3.

```
In[6]:= Map[not, SubstTest[and, implies[and[p1, p3], or[p4, p5]],
          implies[and[p1, p6, not[p7]], p3], implies[not[p7], not[p5]],
                not[implies[and[p1, p6, not[p7]], p4]],
          {p1 -> and[member[s, x], member[y, x]], p3 -> subclass[s, y],
                p4 -> member[y, image[PS, x]], p5 -> equal[y, s],
           p6 -> subclass[cart[x, x], union[S, inverse[S]]], p7 -> subclass[y, s]}]]

Out[6]= or[member[y, image[PS, x]], not[member[s, x]], not[member[y, x]],
          not[subclass[cart[x, x], union[S, inverse[S]]]], subclass[y, s]] == True

In[7]:= (% /. {s -> s_, x -> x_, y -> y_}) /. Equal -> SetDelayed
```

Lemma 4. Eliminating the variable **s**.

```
In[8]:= Map[equal[V, #] &, SubstTest[class, s, or[member[y, u],
            not[member[s, x]], not[member[y, x]], not[subclass[v, w]], subclass[y, s]],
          {u -> image[PS, x], v -> cart[x, x], w -> union[S, inverse[S]]}]] // Reverse

Out[8]= or[member[y, image[PS, x]], not[member[y, x]],
          not[subclass[cart[x, x], union[S, inverse[S]]]], subclass[y, A[x]]] == True

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

Lemma 5. If a there is a member contained in all other members, it must be the least member.

```
In[10]:= Map[not, SubstTest[and, implies[q1, q2], not[implies[and[q1, q3], q4]],
          {q1 -> member[y, x], q2 -> subclass[A[x], y],
             q3 -> subclass[y, A[x]], q4 -> member[A[x], x]}]]

Out[10]= or[member[A[x], x], not[member[y, x]], not[subclass[y, A[x]]]] == True

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

Lemma 6.

```
In[12]:= Map[not, SubstTest[and, implies[and[p1, p2], p3], implies[and[p1, p4], or[p2, p5]],
                   not[implies[and[p1, p4], or[p3, p5]]],
           {p1 -> member[y, x], p2 -> subclass[y, A[x]], p3 -> member[A[x], x],
                   p4 -> subclass[cart[x, x], union[S, inverse[S]]],
            p5 -> member[y, image[PS, x]]}]]

Out[12]= or[member[y, image[PS, x]], member[A[x], x],
           not[member[y, x]], not[subclass[cart[x, x], union[S, inverse[S]]]]] == True

In[13]:= (% /. {x -> x_, y -> y_}) /. Equal -> SetDelayed
```

Theorem. (Result of eliminating the variable **y**.)

```
In[14]:= Map[equal[V, #] &, SubstTest[class, y,
            or[member[y, u], member[v, x], not[member[y, x]], not[subclass[w, z]]],
             {u -> image[PS, x], v -> A[x], w -> cart[x, x], z -> union[S, inverse[S]]}]] // Reverse

Out[14]= or[member[A[x], x],
           not[subclass[cart[x, x], union[S, inverse[S]]]], subclass[x, image[PS, x]]] == True

In[15]:= or[member[A[x_], x_], not[subclass[cart[x_, x_], union[S, inverse[S]]]],
           subclass[x_, image[PS, x_]]] := True
```

Corollary. Any finite nonempty chain of subsets has a least member. This is Theorem **FIN−DCC** for which a proof of length 27 was found by **Otter** on 2000 October 10.

```
In[16]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], not[implies[p1, p3]],
           {p1 -> and[not[member[A[x], x]], subclass[cart[x, x], union[S, inverse[S]]]],
              p2 -> subvariant[PS, x], p3 -> or[not[member[x, FINITE]], equal[0, x]]}]]

Out[16]= or[equal[0, x], member[A[x], x], not[member[x, FINITE]],
           not[subclass[cart[x, x], union[S, inverse[S]]]]] == True

In[17]:= or[equal[0, x_], member[A[x_], x_], not[member[x_, FINITE]],
           not[subclass[cart[x_, x_], union[S, inverse[S]]]]] := True
```

variable−free versions

Theorem.

```
In[18]:= Map[equal[0, #] &, dif[cliques[union[S, inverse[S]]],
            union[subvar[PS], fix[composite[E, BIGCAP]]]] // Renormality]

Out[18]= subclass[cliques[union[S, inverse[S]]],
           union[fix[composite[E, BIGCAP]], subvar[PS]]] == True

In[19]:= subclass[cliques[union[S, inverse[S]]],
           union[fix[composite[E, BIGCAP]], subvar[PS]]] := True
```

Corollary about least members of nonempty finite chains.

```
In[20]:= Map[equal[0, #] &, dif[intersection[FINITE, cliques[union[S, inverse[S]]]],
            union[singleton[0], fix[composite[E, BIGCAP]]]] // Renormality]

Out[20]= subclass[intersection[FINITE, cliques[union[S, inverse[S]]]],
           union[fix[composite[E, BIGCAP]], singleton[0]]] == True

In[21]:= subclass[intersection[FINITE, cliques[union[S, inverse[S]]]],
           union[fix[composite[E, BIGCAP]], singleton[0]]] := True
```

comment

A class is said to satisfy the descending chain condition if every nonempty chain of subsets has a least member. Since any subset of a finite set is finite, finiteness implies the descending chain condition. The converse does not hold: any nonempty subset of the class of ordinals is a chain of sets with a least member.

existence of greatest elements

The strategy is to use relative complements to relate least elements to greatest elements.

```
In[22]:= SubstTest[implies, and[member[y, FINITE], ONEONE[z]],
          member[image[z, y], FINITE], z -> RC[x]]

Out[22]= or[member[image[RC[x], y], FINITE], not[member[y, FINITE]]] == True

In[23]:= or[member[image[RC[x_], y_], FINITE], not[member[y_, FINITE]]] := True
```

transformations of cliques

Lemma.

```
In[24]:= Map[equal[V, class[t, #]] &,
          SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
            {u -> cart[t, t], v -> y, w -> cross[x, x]}]]

Out[24]= subclass[cliques[y], cliques[complement[
            composite[inverse[x], complement[composite[x, y, inverse[x]]], x]]]] == True

In[25]:= (% /. {x -> x_, y -> y_}) /. Equal -> SetDelayed

In[26]:= subclass[cliques[y],
           image[inverse[IMAGE[x]], cliques[composite[x, y, inverse[x]]]]] // AssertTest

Out[26]= subclass[cliques[y], image[inverse[IMAGE[x]], cliques[composite[x, y, inverse[x]]]]] ==
          subclass[fix[y], domain[VERTSECT[x]]]

In[27]:= subclass[cliques[y_],
           image[inverse[IMAGE[x_]], cliques[composite[x_, y_, inverse[x_]]]]] :=
          subclass[fix[y], domain[VERTSECT[x]]]

In[28]:= Map[implies[thin[x], #] &,
           SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
             {u -> cliques[y],
             v -> image[inverse[IMAGE[x]], cliques[composite[x, y, inverse[x]]]],
               w -> IMAGE[x]}]] // MapNotNot

Out[28]= or[not[equal[V, domain[VERTSECT[x]]]],
           subclass[image[IMAGE[x], cliques[y]], cliques[composite[x, y, inverse[x]]]]] == True

In[29]:= or[not[equal[V, domain[VERTSECT[x_]]]], subclass[
            image[IMAGE[x_], cliques[y_]], cliques[composite[x_, y_, inverse[x_]]]]] := True
```

application

Lemma 1. (application of the lemma in the preceding section)

```
In[30]:= SubstTest[implies, thin[w],
          subclass[image[IMAGE[w], cliques[y]], cliques[composite[w, y, inverse[w]]]],
            {w -> RC[x], y -> union[S, inverse[S]]}]

Out[30]= subclass[image[IMAGE[RC[x]], cliques[union[S, inverse[S]]]],
           cliques[union[composite[inverse[S], id[intersection[image[V, singleton[x]], P[x]]]],
             composite[id[P[x]], S, id[intersection[image[V, singleton[x]], P[x]]]]]]] == True

In[31]:= (% /. x -> x_) /. Equal -> SetDelayed
```

Lemma 2. (application of monotonicity of cliques)

```
In[32]:= SubstTest[implies, subclass[w, x], subclass[cliques[w], cliques[x]],
            {x -> union[S, inverse[S]],
          w -> union[composite[inverse[S], id[intersection[image[V, singleton[x]], P[x]]]],
            composite[id[P[x]], S, id[intersection[image[V, singleton[x]], P[x]]]]]}]

Out[32]= subclass[
          cliques[union[composite[inverse[S], id[intersection[image[V, singleton[x]], P[x]]]],
            composite[id[P[x]], S, id[intersection[image[V, singleton[x]], P[x]]]]]],
          cliques[union[S, inverse[S]]]] == True

In[33]:= (% /. x -> x_) /. Equal -> SetDelayed
```

Combining the two lemmas yields the fact that the class of chains is invariant under **IMAGE[RC[x]]**.

```
In[34]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
            {u -> image[IMAGE[RC[x]], cliques[union[S, inverse[S]]]],
                v -> cliques[
            union[composite[inverse[S], id[intersection[image[V, singleton[x]], P[x]]]],
             composite[id[P[x]], S, id[intersection[image[V, singleton[x]], P[x]]]]]],
            w -> cliques[union[S, inverse[S]]]}]

Out[34]= subclass[image[IMAGE[RC[x]], cliques[union[S, inverse[S]]]],
          cliques[union[S, inverse[S]]]] == True

In[35]:= subclass[image[IMAGE[RC[x_]], cliques[union[S, inverse[S]]]],
          cliques[union[S, inverse[S]]]] := True

In[36]:= Map[implies[member[x, FINITE], #] &,
         SubstTest[implies, and[member[x, y], subclass[y, z]], member[x, z],
            {y -> cliques[union[S, inverse[S]]],
           z -> image[inverse[IMAGE[RC[U[x]]]], cliques[union[S, inverse[S]]]]}]]

Out[36]= or[not[member[x, FINITE]], not[subclass[cart[x, x], union[S, inverse[S]]]],
          subclass[cart[image[RC[U[x]], x], image[RC[U[x]], x]], union[S, inverse[S]]]] == True

In[37]:= (% /. x -> x_) /. Equal -> SetDelayed
```

Double negation is used to combine these results:

```
In[38]:= implies[member[x, intersection[FINITE, cliques[union[S, inverse[S]]]]],
            member[image[RC[U[x]], x], intersection[FINITE, cliques[union[S, inverse[S]]]]]] //
         NotNotTest

Out[38]= or[and[member[image[RC[U[x]], x], FINITE],
           subclass[cart[image[RC[U[x]], x], image[RC[U[x]], x]], union[S, inverse[S]]]],
          not[member[x, FINITE]], not[subclass[cart[x, x], union[S, inverse[S]]]]] == True

In[39]:= (% /. x -> x_) /. Equal -> SetDelayed
```

Lemma.

```
In[40]:= SubstTest[implies, and[equal[u, v], member[v, w]],
         member[u, w], {u -> U[x], v -> 0, w -> x}]

Out[40]= or[member[U[x], x], not[equal[x, singleton[0]]]] == True

In[41]:= or[member[U[x_], x_], not[equal[x_, singleton[0]]]] := True
```

This implies:

```
In[42]:= equiv[or[equal[x, singleton[0]], member[U[x], x]], member[U[x], x]]

Out[42]= True
```

A rewrite rule corresponding to this fact is needed:

```
In[43]:= or[equal[x_, singleton[0]], member[U[x_], x_]] := member[U[x], x]
```

Lemma.

```
In[44]:= and[member[x, V], member[intersection[image[V, singleton[x]], U[x]], x],
           not[equal[0, x]]] // AssertTest

Out[44]= and[member[x, V], member[intersection[image[V, singleton[x]], U[x]], x],
           not[equal[0, x]]] == member[U[x], x]

In[45]:= and[member[x_, V], member[intersection[image[V, singleton[x_]], U[x_]], x_],
           not[equal[0, x_]]] := member[U[x], x]
```

The theorem about existence of least elements implies:

```
In[46]:= SubstTest[or, equal[0, y], member[A[y], y],
          not[member[y, FINITE]], not[subclass[cart[y, y], union[S, inverse[S]]]],
             y -> image[RC[U[x]], x]]

Out[46]= or[equal[0, x], member[U[x], x], not[member[x, V]],
           not[member[image[RC[U[x]], x], FINITE]], not[subclass[
             cart[image[RC[U[x]], x], image[RC[U[x]], x]], union[S, inverse[S]]]]] == True

In[47]:= (% /. x -> x_) /. Equal -> SetDelayed
```

Theorem: any nonempty finite chain has a greatest element.

```
In[48]:= Map[not, SubstTest[and, implies[p1, p2], implies[p1, p3],
           implies[and[p2, p3], p4], not[implies[p1, p4]], {p1 -> and[member[x, FINITE],
              not[equal[0, x]], subclass[cart[x, x], union[S, inverse[S]]]],
                 p2 -> and[member[image[RC[U[x]], x], FINITE],
              subclass[cart[image[RC[U[x]], x], image[RC[U[x]], x]], union[S, inverse[S]]]],
            p3 -> member[x, complement[singleton[0]]],
                 p4 -> member[U[x], x]}]]

Out[48]= or[equal[0, x], member[U[x], x], not[member[x, FINITE]],
           not[subclass[cart[x, x], union[S, inverse[S]]]]] == True

In[49]:= or[equal[0, x_], member[U[x_], x_], not[member[x_, FINITE]],
           not[subclass[cart[x_, x_], union[S, inverse[S]]]]] := True
```

variable free formulation

The variable−free form is obtained by simply replacing **BIGCAP** with **BIGCUP**, but note that the fixed−point class is transformed because the inverse of the function **BIGCUP** is thin, whereas the inverse of the function **BIGCAP** is not.

```
In[50]:= Map[equal[0, #] &, dif[intersection[FINITE, cliques[union[S, inverse[S]]]],
            union[singleton[0], fix[composite[E, BIGCUP]]]]] // Renormality]

Out[50]= subclass[intersection[FINITE, cliques[union[S, inverse[S]]]],
           union[fix[composite[inverse[IMAGE[inverse[BIGCUP]]], E]], singleton[0]]] == True
```

```
In[51]:= subclass[intersection[FINITE, cliques[union[S, inverse[S]]]],
           union[fix[composite[inverse[IMAGE[inverse[BIGCUP]]], E]], singleton[0]]] := True
```