finitely generated binary closure theorem

Johan G. F. Belinfante
2011 January 27

```
In[1]:= SetDirectory["l:"]; << goedel.11jan26a

        :Package Title: goedel.11jan26a          2011 January 26 at 8:00 p.m.

        It is now:  2011 Jan 27 at 10:12

        Loading Simplification Rules

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

        weightlimit = 40
```

summary

Every set **setpart[y]** is a subset of a set **hull[binclosed[thinpart[x]], setpart[y]]** that is binary closed under any given thin relation **thinpart[x]**. It is often said that the binary closed set **hull[binclosed[thinpart[x]], setpart[y]]** is **generated** by the set **setpart[y]**. A binary closed set is **finitely generated** if it is generated by a finite set. In this notebook it is shown that every element of the binary closed set generated by a set **setpart[y]** is a member of a binary closed set that is generated by some finite subset of **setpart[y]**.

The theorem derived in this notebook is a special case of Corollary 1.10 on page 25 of the following reference.

```
In[2]:= "Ralph N. McKenzie, George E. McNulty and Walter F. Taylor, Algebras, Lattices,
         Varieties, volume 1, Wadsworth and Brooks/Cole, Belmont, California, 1967.";
```

The proof given in this reference uses induction. The derivation to be presented here is similar, except that induction is not used. Nonetheless, analogs of both the base case and the induction step of their proof are used in the derivation below.

inclusion in one direction

The theorem to be derived is an equation. The following lemma establishes an inclusion in one direction.

Lemma.

```
In[3]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
          {t → composite[inverse[E], HULL[binclosed[thinpart[x]]]],
           u → intersection[FINITE, P[setpart[y]]], v → P[setpart[y]]}] // Reverse

Out[3]= subclass[U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]],
          hull[binclosed[thinpart[x]], setpart[y]]] == True
```

```
In[4]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

base case

In this section an analog of the base case for the induction proof given in the cited reference is derived.

Lemma.

```
In[5]:= Map[implies[member[x, y], #] &,
          SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
           {t → composite[inverse[E], HULL[binclosed[thinpart[z]]]],
            u → set[set[x]], v → intersection[FINITE, P[y]]}]] // Reverse

Out[5]= or[not[member[x, y]], subclass[hull[binclosed[thinpart[z]], set[x]],
           U[image[HULL[binclosed[thinpart[z]]], intersection[FINITE, P[y]]]]]] == True

In[6]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[7]:= Map[implies[member[x, y], #] &, SubstTest[implies, and[member[x, u], subclass[u, v]],
           member[x, v], {u -> hull[binclosed[thinpart[z]], set[x]],
            v -> U[image[HULL[binclosed[thinpart[z]]], intersection[FINITE, P[y]]]]}]] // Reverse

Out[7]= or[member[x, U[image[HULL[binclosed[thinpart[z]]], intersection[FINITE, P[y]]]]],
          not[member[x, y]], not[subclass[hull[binclosed[thinpart[z]], set[x]],
            U[image[HULL[binclosed[thinpart[z]]], intersection[FINITE, P[y]]]]]]] == True

In[8]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[17]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3], not[implies[p1, p3]],
            {p1 → member[x, y], p2 -> subclass[hull[binclosed[thinpart[z]], set[x]],
               U[image[HULL[binclosed[thinpart[z]]], intersection[FINITE, P[y]]]]],
             p3 → member[x, U[image[HULL[binclosed[thinpart[z]]],
                 intersection[FINITE, P[y]]]]]}]] // Reverse

Out[17]= or[member[x, U[image[HULL[binclosed[thinpart[z]]], intersection[FINITE, P[y]]]]],
           not[member[x, y]]] == True

In[18]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Theorem. Analog of the base case for the induction proof presented in the cited reference.

```
In[19]:= Map[equal[V, #] &, SubstTest[class, u, implies[member[u, y], member[u, v]],
           v -> U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[y]]]]]]

Out[19]= subclass[y, U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[y]]]]] == True

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

induction step

In this section an analog of the induction step in the proof given in the cited reference is derived.

Lemma.

```
In[21]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
           subclass[u, w], {u → y, v → union[y, z], w → hull[x, union[y, z]]}] // Reverse

Out[21]= subclass[y, hull[x, union[y, z]]] == True

In[22]:= subclass[y_, hull[x_, union[y_, z_]]] := True
```

Lemma. **hull[binclosed[thinpart[x]], fin[y] ⋃ fin[z]] ∈ binclosed[thinpart[x]]**.

```
In[23]:= SubstTest[subclass,
           image[thinpart[x], cartsq[hull[binclosed[thinpart[x]], setpart[t]]]],
           hull[binclosed[thinpart[x]], setpart[t]], t → union[fin[y], fin[z]]] // Reverse

Out[23]= subclass[image[thinpart[x], cart[hull[binclosed[thinpart[x]], union[fin[y], fin[z]]],
             hull[binclosed[thinpart[x]], union[fin[y], fin[z]]]]],
           hull[binclosed[thinpart[x]], union[fin[y], fin[z]]]] == True

In[24]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Lemma. (Analog of the induction step in the proof given in the cited reference.)

```
In[25]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u -> image[thinpart[x], cart[hull[binclosed[thinpart[x]], fin[y]],
               hull[binclosed[thinpart[x]], fin[z]]]], v ->
             image[thinpart[x], cartsq[hull[binclosed[thinpart[x]], union[fin[y], fin[z]]]]],
            w -> hull[binclosed[thinpart[x]], union[fin[y], fin[z]]]}] // Reverse

Out[25]= subclass[image[thinpart[x],
            cart[hull[binclosed[thinpart[x]], fin[y]], hull[binclosed[thinpart[x]], fin[z]]]],
           hull[binclosed[thinpart[x]], union[fin[y], fin[z]]]] == True

In[26]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Lemma.

```
In[27]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
           {t → composite[inverse[E], HULL[binclosed[thinpart[x]]]],
            u → set[union[fin[r], fin[s]]], v -> intersection[FINITE, P[setpart[y]]]}] // Reverse

Out[27]= or[not[subclass[fin[r], setpart[y]]], not[subclass[fin[s], setpart[y]]],
           subclass[hull[binclosed[thinpart[x]], union[fin[r], fin[s]]], U[
             image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]]] == True

In[28]:= (% /. {r → r_, s → s_, x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[29]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3],
            not[implies[p1, p3]], {p1 → subclass[union[fin[r], fin[s]], setpart[y]],
             p2 -> subclass[hull[binclosed[thinpart[x]], union[fin[r], fin[s]]],
               U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]],
             p3 -> subclass[image[thinpart[x], cart[hull[binclosed[thinpart[x]], fin[r]],
                 hull[binclosed[thinpart[x]], fin[s]]]], U[image[HULL[binclosed[thinpart[x]]],
                 intersection[FINITE, P[setpart[y]]]]]]}]] // Reverse

Out[29]= or[not[subclass[fin[r], setpart[y]]],
           not[subclass[fin[s], setpart[y]]], subclass[image[thinpart[x], cart[
              hull[binclosed[thinpart[x]], fin[r]], hull[binclosed[thinpart[x]], fin[s]]]], U[
             image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]]] == True

In[30]:= (% /. {r → r_, s → s_, x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma. (Eliminate the **fin** wrappers.)

```
In[31]:= SubstTest[implies, and[equal[u, fin[r]], equal[v, fin[s]],
            subclass[u, setpart[y]], subclass[v, setpart[y]]], subclass[image[thinpart[x],
             cart[hull[binclosed[thinpart[x]], u], hull[binclosed[thinpart[x]], v]]],
            U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]],
           {r → u, s → v}] // Reverse

Out[31]= or[not[member[u, FINITE]], not[member[v, FINITE]], not[subclass[u, setpart[y]]],
           not[subclass[v, setpart[y]]], subclass[image[thinpart[x],
             cart[hull[binclosed[thinpart[x]], u], hull[binclosed[thinpart[x]], v]]], U[
             image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]]] == True

In[32]:= (% /. {u → u_, v → v_, x → x_, y → y_}) /. Equal → SetDelayed
```

---

main theorem

The variables **u** and **v** can be eliminated as follows. (This takes a minute or so.)

Theorem. **U[image[HULL[binclosed[thinpart[x]]], FINITE ⋂ P[setpart[y]]]] ∈ binclosed[thinpart[x]]**.

```
In[33]:= Map[empty[composite[Id, complement[#]]] &,
          SubstTest[class, pair[u, v], implies[and[member[u, s], member[v, s]],
            subclass[image[w, cart[APPLY[t, u], APPLY[t, v]]], z]],
           {s → intersection[FINITE, P[setpart[y]]], w → thinpart[x],
            t → HULL[binclosed[thinpart[x]]],
            z -> U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]}]]

Out[33]= subclass[image[thinpart[x],
            cart[U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]],
             U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]]],
           U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]] == True
```

```
In[34]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma. (Inclusion in opposite direction.)

```
In[35]:= SubstTest[implies, and[subclass[setpart[y], v], member[v, u]],
           subclass[hull[u, setpart[y]], v],
           {u → binclosed[thinpart[x]], v -> U[image[HULL[binclosed[thinpart[x]]],
               intersection[FINITE, P[setpart[y]]]]]]}] // Reverse

Out[35]= subclass[hull[binclosed[thinpart[x]], setpart[y]],
           U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]] == True

In[36]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Main Theorem.

```
In[37]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> hull[binclosed[thinpart[x]], setpart[y]],
           v -> U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]]}]

Out[37]= equal[hull[binclosed[thinpart[x]], setpart[y]],
           U[image[HULL[binclosed[thinpart[x]]], intersection[FINITE, P[setpart[y]]]]]] == True

In[38]:= U[image[HULL[binclosed[thinpart[x_]]], intersection[FINITE, P[setpart[y_]]]]] :=
          hull[binclosed[thinpart[x]], setpart[y]]
```

The variable **y** can be eliminated using **reify**.

```
In[40]:= Map[VERTSECT, SubstTest[reify, y, image[t, intersection[FINITE, P[setpart[y]]]],
           t → composite[inverse[E], HULL[binclosed[thinpart[x]]]]]]]

Out[40]= composite[BIGCUP, IMAGE[HULL[binclosed[thinpart[x]]]], IMAGE[id[FINITE]], POWER] ==
          HULL[binclosed[thinpart[x]]]

In[41]:= composite[BIGCUP, IMAGE[HULL[binclosed[thinpart[x_]]]], IMAGE[id[FINITE]], POWER] :=
          HULL[binclosed[thinpart[x]]]
```