MAXIMAL[x] ⋂ GREATEST[x]

Johan G. F. Belinfante
2009 June 19

```
In[1]:= SetDirectory["l:"]; << goedel.09jun17a;<< tools.m

        :Package Title: goedel.09jun17a              2009 June 17 at 8:20 a.m.

        It is now:  2009 Jun 19 at 10:44

        Loading Simplification Rules

        TOOLS.M                                      Revised 2009 June 1

        weightlimit = 40
```

summary

The relations **MAXIMAL[x]** and **GREATEST[x]** are defined in the **GOEDEL** program for an arbitrary class **x**, not just for partial orders. In general, neither of these relations need be a function, but their intersection is always a function. This intersection is not empty if and only if **fix[x]** is not empty.

examples

Neither **GREATEST[x]** not **MAXIMAL[x]** need be a function.

Lemma.

```
In[2]:= FUNCTION[composite[inverse[E], id[x]]] // AssertTest

Out[2]= FUNCTION[composite[inverse[E], id[x]]] == subclass[x, union[range[SINGLETON], set[0]]]

In[3]:= FUNCTION[composite[inverse[E], id[x_]]] := subclass[x, union[range[SINGLETON], set[0]]]
```

Example. The relation **GREATEST[cart[x, x]]** is a function only if **x** is empty or a singleton.

```
In[4]:= FUNCTION[GREATEST[cart[x, x]]]

Out[4]= or[equal[0, x], member[x, range[SINGLETON]]]
```

Example. The relation **MAXIMAL[id[x]]** is never a function.

```
In[5]:= FUNCTION[MAXIMAL[id[x]]]

Out[5]= False
```

Comment. Since **cart[x, x]** and **id[x]** are symmetric, these examples also suffice for the relations **LEAST[x]** and **MINIMAL[x]**. Neither of these relations need be a function.

a general theorem

Lemma.

```
In[6]:= SubstTest[implies, and[subclass[s, x], subclass[t, y]],
          subclass[composite[s, t], composite[x, y]],
          {s → intersection[x, u], t → intersection[y, v]}] // Reverse

Out[6]= subclass[composite[intersection[u, x], intersection[v, y]], composite[x, y]] == True

In[7]:= subclass[composite[intersection[u_, x_], intersection[v_, y_]],
          composite[x_, y_]] := True
```

Lemma.

```
In[8]:= SubstTest[subclass, composite[t, inverse[t]],
          Id, t → intersection[x, inverse[y]]] // Reverse

Out[8]= subclass[composite[intersection[x, inverse[y]], intersection[y, inverse[x]]], Id] ==
         FUNCTION[intersection[x, inverse[y]]]

In[9]:= subclass[composite[intersection[x_, inverse[y_]], intersection[y_, inverse[x_]]], Id] :=
         FUNCTION[intersection[x, inverse[y]]]
```

Theorem. If $\mathbf{x} \circ \mathbf{y} \subset \mathbf{Id}$, then $\mathbf{x} \cap$ **inverse[y]** is a function.

```
In[10]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u -> composite[intersection[x, inverse[y]], intersection[inverse[x], y]],
            v -> composite[x, y], w → Id}] // Reverse

Out[10]= or[FUNCTION[intersection[x, inverse[y]]], not[subclass[composite[x, y], Id]]] == True

In[11]:= or[FUNCTION[intersection[x_, inverse[y_]]],
           not[subclass[composite[x_, y_], Id]]] := True
```

derivation

In this section it is shown that the intersection of the relations **MAXIMAL[x]** and **GREATEST[x]** is always a function.

Theorem.

```
In[12]:= SubstTest[subclass, composite[t, intersection[u, v]],
           intersection[composite[t, u], composite[t, v]],
           {t → MAXIMAL[x], u → E, v → inverse[UB[x]]}] // Reverse

Out[12]= subclass[composite[MAXIMAL[x], inverse[GREATEST[x]]], id[fix[x]]] == True

In[13]:= subclass[composite[MAXIMAL[x_], inverse[GREATEST[x_]]], id[fix[x_]]] := True
```

Corollary.

```
In[14]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u -> composite[MAXIMAL[x], inverse[GREATEST[x]]], v -> id[fix[x]], w → Id}] // Reverse

Out[14]= subclass[composite[MAXIMAL[x], inverse[GREATEST[x]]], Id] == True

In[15]:= subclass[composite[MAXIMAL[x_], inverse[GREATEST[x_]]], Id] := True
```

Theorem. The relation **GREATEST[x] ⋂ MAXIMAL[x]** is always a function.

```
In[16]:= SubstTest[implies, subclass[composite[u, v], Id],
           FUNCTION[intersection[u, inverse[v]]],
           {u → MAXIMAL[x], v → inverse[GREATEST[x]]}] // Reverse

Out[16]= FUNCTION[intersection[GREATEST[x], MAXIMAL[x]]] == True

In[17]:= FUNCTION[intersection[GREATEST[x_], MAXIMAL[x_]]] := True
```

Corollary. The relation **LEAST[x] ⋂ MINIMAL[x]** is always a function.

```
In[18]:= SubstTest[FUNCTION, intersection[GREATEST[t], MAXIMAL[t]], t → inverse[x]] // Reverse

Out[18]= FUNCTION[intersection[LEAST[x], MINIMAL[x]]] == True

In[19]:= FUNCTION[intersection[LEAST[x_], MINIMAL[x_]]] := True
```

---

a lower bound

In general, the intersection **GREATEST[x] ⋂ MAXIMAL[x]** may be empty. This is the case, for example, when **x = Di**.

```
In[20]:= intersection[GREATEST[x], MAXIMAL[x]] /. x → Di

Out[20]= 0
```

In this section it is shown that if **fix[x]** is not empty, then neither is **GREATEST[x] ⋂ MAXIMAL[x]**.

Lemma.

```
In[21]:= member[set[t], fix[composite[inverse[GREATEST[x]], MAXIMAL[x]]]] // AssertTest

Out[21]= member[set[t], fix[composite[inverse[GREATEST[x]], MAXIMAL[x]]]] == member[t, fix[x]]
```

```
In[22]:= member[set[t_], fix[composite[inverse[GREATEST[x_]], MAXIMAL[x_]]]] :=
          member[t, fix[x]]
```

Lemma.

```
In[23]:= SubstTest[implies, member[u, domain[v]], not[empty[v]],
            {u -> set[t], v -> intersection[GREATEST[x], MAXIMAL[x]]}] // Reverse

Out[23]= or[not[equal[0, intersection[GREATEST[x], MAXIMAL[x]]]], not[member[t, fix[x]]]] == True

In[24]:= (% /. {t → t_, x → x_}) /. Equal → SetDelayed
```

The variable **t** can be eliminated.

Theorem.

```
In[25]:= Map[equal[V, #] &, SubstTest[class, t, implies[member[t, u], not[empty[v]]],
            {u → fix[x], v -> intersection[GREATEST[x], MAXIMAL[x]]}]]

Out[25]= or[equal[0, fix[x]], not[equal[0, intersection[GREATEST[x], MAXIMAL[x]]]]] == True

In[26]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[27]:= equal[composite[Id, x], intersection[Di, x]] // AssertTest

Out[27]= equal[composite[Id, x], intersection[Di, x]] == equal[0, fix[x]]

In[28]:= equal[composite[Id, x_], intersection[Di, x_]] := equal[0, fix[x]]
```

Lemma.

```
In[29]:= SubstTest[implies, equal[t, intersection[Di, x]],
            disjoint[GREATEST[t], MAXIMAL[t]], t → composite[Id, x]] // Reverse

Out[29]= or[equal[0, intersection[GREATEST[x], MAXIMAL[x]]], not[equal[0, fix[x]]]] == True

In[30]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem.

```
In[31]:= equiv[equal[0, intersection[GREATEST[x], MAXIMAL[x]]], equal[0, fix[x]]]

Out[31]= True

In[32]:= equal[0, intersection[GREATEST[x_], MAXIMAL[x_]]] := equal[0, fix[x]]
```

Corollary.

```
In[33]:= SubstTest[disjoint, GREATEST[t], MAXIMAL[t], t → inverse[x]] // Reverse

Out[33]= equal[0, intersection[LEAST[x], MINIMAL[x]]] == equal[0, fix[x]]
```

```
In[34]:= equal[0, intersection[LEAST[x_], MINIMAL[x_]]] := equal[0, fix[x]]
```