definitions for GLB[x] and LUB[x]

Johan G. F. Belinfante
2005 September 27

The concepts of greatest lower bound and least upper bound are basic, and have many important applications. For example, for the case of the subset relation \( S \), the functions \texttt{BIGCAP} and \texttt{BIGCUP} are related to greatest lower bounds and least upper bounds, respectively:

\begin{verbatim}
In[2]:= class[pair[u, v], member[v, greatest[z, lb[z, u]]]] /. z \rightarrow S
Out[2]= BIGCAP

In[3]:= class[pair[u, v], member[v, least[z, ub[z, u]]]] /. z \rightarrow S
Out[3]= BIGCUP
\end{verbatim}

For many applications these literal concepts of greatest lower bounds and least upper bounds produce complicated formulas. For example, if one simply replaces the subset relation \( S \) in these formulas with a cartesian product, one finds:

\begin{verbatim}
In[4]:= class[pair[u, v], member[v, greatest[z, lb[z, u]]]] /. z \rightarrow \text{cart}[x, y]
Out[4]= union[cart[intersection[complement[set[0]], P[y]], intersection[x, y]],
  cart[P[y], intersection[x, y, complement[image[V, complement[x]]]]],
  cart[set[0], intersection[y, complement[image[V, complement[x]]]]]]
\end{verbatim}

A much simpler result is obtained by making a minor modification, replacing \( lb[z,u] \) by its intersection with \texttt{domain[z]}.

\begin{verbatim}
In[5]:= class[pair[u, v], member[v, greatest[z, intersection[domain[z], lb[z, u]]]]] /. 
  z \rightarrow \text{cart}[x, y]
Out[5]= \text{cart}[P[y], \text{intersection}[x, y]]
\end{verbatim}

Intersecting with \texttt{domain[z]} only affects how the empty set is treated:
AssertTest[implies, andsubclass[u, v], subclass[v, w], subclass[u, w], \{u \rightarrow lb[x, y], v \rightarrow image[\text{inverse}[x], y], w \rightarrow domain[x]\}]

Out[6]= \text{or}[\text{equal}[0, y], \text{subclass}[lb[x, y], domain[x]]] := \text{True}

Out[7]= \text{or}[\text{equal}[0, y_\text{\_}], \text{subclass}[lb[x_\text{\_}, y_\text{\_}], domain[x_\text{\_}]]] := \text{True}

In principle, one could redefine the lower bound constructor lb[x,y] by including an intersection with domain[x], but doing so does not appear to be particularly useful. In this notebook, a fresh start is made. The old definitions of GLB and LUB have been removed from the GOEDEL program, along with all rewrite rules involving GLB and LUB. New membership rules wrapped with \texttt{class} have been added to the GOEDEL program, providing a starting point from which various basic properties of GLB[x] and LUB[x] will be derived.

\textbf{wrapped and unwrapped membership rules}

The following membership rules for GLB[x] and LUB[x], wrapped with \texttt{class}, serve as new definitions of these constructors.

In[8]= \text{Begin}["Goedel\`Private`"];

In[9]= \text{FirstMatch}[\text{class}[w_\text{\_}, \text{member}[x_\text{\_}, \text{HoldPattern}[\text{GLB}[y_\text{\_}]]]]]

Out[9]= \text{class}[w_\text{\_}, \text{member}[x_\text{\_}, \text{GLB}[y_\text{\_}]]] := \text{Module}[\{u = \text{Unique}[\text{}, v = \text{Unique}[\text{}], \text{class}[w, \text{exists}[u, v, \text{and}[\text{equal}[x, \text{pair}[u, v]]], \text{member}[u, v], \text{member}[v, \text{domain}[y]], \text{subclass}[u, \text{image}[y, \text{set}[v]]], \text{subclass}[\text{intersection}[\text{domain}[y], \text{lb}[y, u]], \text{image}[\text{inverse}[y, \text{set}[v]]]]]])

In[10]= \text{FirstMatch}[\text{class}[w_\text{\_}, \text{member}[x_\text{\_}, \text{HoldPattern}[\text{LUB}[y_\text{\_}]]]]]

Out[10]= \text{class}[w_\text{\_}, \text{member}[x_\text{\_}, \text{LUB}[y_\text{\_}]]] := \text{Module}[\{u = \text{Unique}[\text{}], v = \text{Unique}[\text{}], \text{class}[w, \text{exists}[u, v, \text{and}[\text{equal}[x, \text{pair}[u, v]]], \text{member}[u, v], \text{member}[v, \text{range}[y]], \text{subclass}[u, \text{image}[\text{inverse}[y, \text{set}[v]]], \text{subclass}[\text{intersection}[\text{range}[y], \text{ub}[y, u]], \text{image}[y, \text{set}[v]]]]))

From the wrapped membership rule one can derive the following unwrapped membership rule for GLB[z] for the special case of ordered pairs:

In[11]= \text{member}[\text{pair}[x, y], \text{GLB}[z]] // \text{AssertTest}

Out[11]= \text{member}[\text{pair}[x, y], \text{GLB}[z]] := \text{and}[\text{member}[x, V], \text{member}[y, \text{domain}[z]], \text{subclass}[x, \text{image}[z, \text{set}[y]]], \text{subclass}[\text{intersection}[\text{domain}[z], \text{lb}[z, x]], \text{image}[\text{inverse}[z, \text{set}[y]]]]

In[12]= \text{member}[\text{pair}[x_\text{\_}, y_\text{\_}], \text{GLB}[z_\text{\_}]] := \text{and}[\text{member}[x, V], \text{member}[y, \text{domain}[z]], \text{subclass}[x, \text{image}[z, \text{set}[y]]], \text{subclass}[\text{intersection}[\text{domain}[z], \text{lb}[z, x]], \text{image}[\text{inverse}[z, \text{set}[y]]]]

A similar result holds for LUB[z].
These definitions can be restated as follows:

\[\text{In}[15] := \text{member}[\text{pair}[x, y], \text{GLB}[z]] \Rightarrow \text{and}[\text{member}[x, V], \text{member}[y, \text{greatest}[z, \text{intersection}[\text{domain}[z], \text{lb}[z, x]]]]] \]

\[\text{Out}[15] = \text{True} \]

\[\text{In}[16] := \text{member}[\text{pair}[x, y], \text{LUB}[z]] \Rightarrow \text{and}[\text{member}[x, V], \text{member}[y, \text{least}[z, \text{intersection}[\text{range}[z], \text{ub}[z, x]]]]] \]

\[\text{Out}[16] = \text{True} \]

Our main concerns in this notebook are with normalization rules for these constructors.

---

**normalization for GLB[x]**

The basic **Normality** result for the GLB[x] constructor is this:

\[\text{In}[17] := \text{GLB}[x] \Rightarrow \text{Normality} \Rightarrow \text{Reverse} \]

\[\text{Out}[17] = \text{composite}[\text{id}[\text{domain}[x]], \text{intersection}[\text{complement}[\text{composite}[\text{complement}[x], \text{id}[\text{domain}[x]], \text{LB}[x]]], \text{LB}[x]]] = \text{GLB}[x] \]

\[\text{In}[18] := \text{composite}[\text{id}[\text{domain}[x_]]], \text{intersection}[\text{complement}[\text{composite}[\text{complement}[x_], \text{id}[\text{domain}[x_]], \text{LB}[x_]]], \text{LB}[x_]]] := \text{GLB}[x] \]

A second application of **Normality** yields a basic rewrite rule:

\[\text{In}[19] := \text{GLB}[x] \Rightarrow \text{Normality} \Rightarrow \text{Reverse} \]

\[\text{Out}[19] = \text{composite}[\text{Id}, \text{GLB}[x]] = \text{GLB}[x] \]

\[\text{In}[20] := \text{composite}[\text{Id}, \text{GLB}[x_]] := \text{GLB}[x] \]

The GLB constructor ignores any elements that are not ordered pairs.

\[\text{In}[21] := \text{GLB}[\text{composite}[\text{Id}, x]] \Rightarrow \text{RelnNormality} \]

\[\text{Out}[21] = \text{GLB}[\text{composite}[\text{Id}, x]] = \text{GLB}[x] \]

\[\text{In}[22] := \text{GLB}[\text{composite}[\text{Id}, x_]] := \text{GLB}[x] \]
normalization for LUB[x]

Similar results hold for the LUB[x] constructor. The basic Normality result is this:

In[23]:=  LUB[x] // Normality // Reverse

Out[23]=  composite[id[range[x]], intersection[
            complement[composite[complement[inverse[x]], id[range[x]], UB[x]]], UB[x]]] = LUB[x]

In[24]:=  composite[id[range[x_]], intersection[complement[
            composite[complement[inverse[x_]], id[range[x_]], UB[x_]]], UB[x_]]] := LUB[x]

A second application yields:

In[25]:=  LUB[x] // Normality // Reverse

Out[25]=  composite[Id, LUB[x]] = LUB[x]

In[26]:=  composite[Id, LUB[x_]] := LUB[x]

The LUB constructor also ignores any elements that are not ordered pairs.

In[27]:=  LUB[composite[Id, x]] // RelnNormality

Out[27]=  LUB[composite[Id, x]] = LUB[x]

In[28]:=  LUB[composite[Id, x_]] := LUB[x]

interrelations between GLB and LUB

Each of the constructors GLB and LUB is related to the other as follows:

In[29]:=  GLB[inverse[x]] // RelnNormality

Out[29]=  GLB[inverse[x]] = LUB[x]

In[30]:=  GLB[inverse[x_]] := LUB[x]

In[31]:=  LUB[inverse[x]] // RelnNormality

Out[31]=  LUB[inverse[x]] = GLB[x]

In[32]:=  LUB[inverse[x_]] := GLB[x]
vertical sections

Vertical section rules provide an important alternative to membership rules, and are needed in order to be able to take full advantage of the VSNormality test, and its relatives. The vertical section rule for GLB can be derived using the more primitive Normality test:

\[
\begin{align*}
\text{In [33]} & : = \text{image}[\text{GLB}[x], \text{set}[y]] \text{ // Normality} \\
\text{Out [33]} & = \text{image}[\text{GLB}[x], \text{set}[y]] = \text{intersection}[\text{domain}[x], \text{image}[V, \text{set}[y]], \text{lb}[x, y], \text{ub}[x, \text{intersection}[\text{domain}[x], \text{lb}[x, y]]]\]
\end{align*}
\]

\[
\begin{align*}
\text{In [34]} & : = \text{image}[\text{GLB}[x_\_], \text{set}[y_\_]] := \text{intersection}[\text{domain}[x], \text{image}[V, \text{set}[y]], \text{lb}[x, y], \text{ub}[x, \text{intersection}[\text{domain}[x], \text{lb}[x, y]]]
\end{align*}
\]

Replacing \(x\) with its inverse yields an analogous result for LUB.

\[
\begin{align*}
\text{In [35]} & : = \text{SubstTest}[\text{image}, \text{GLB}[z], \text{set}[y], z \rightarrow \text{inverse}[x]] \\
\text{Out [35]} & = \text{image}[\text{LUB}[x], \text{set}[y]] = \text{intersection}[\text{image}[V, \text{set}[y]], \text{lb}[x, \text{intersection}[\text{range}[x], \text{ub}[x, y]]], \text{range}[x], \text{ub}[x, y]]
\end{align*}
\]

\[
\begin{align*}
\text{In [36]} & : = \text{image}[\text{LUB}[x_\_], \text{set}[y_\_]] := \text{intersection}[\text{image}[V, \text{set}[y]], \text{lb}[x, \text{intersection}[\text{range}[x], \text{ub}[x, y]]], \text{range}[x], \text{ub}[x, y]]
\end{align*}
\]

examples

The results for the subset relation \(S\) are not affected by the revision of the definitions of GLB and LUB.

\[
\begin{align*}
\text{In [37]} & : = \text{GLB}[S] \text{ // RelinNormality} \\
\text{Out [37]} & = \text{GLB}[S] = \text{BIGCAP} \\
\text{In [38]} & : = \text{GLB}[S] := \text{BIGCAP} \\
\text{In [39]} & : = \text{LUB}[S] \text{ // RelinNormality} \\
\text{Out [39]} & = \text{LUB}[S] = \text{BIGCUP} \\
\text{In [40]} & : = \text{LUB}[S] := \text{BIGCUP}
\end{align*}
\]

For the case of cartesian products, one obtains the simple formulas mentioned in the introduction of this notebook.

\[
\begin{align*}
\text{In [41]} & : = \text{GLB}[\text{cart}[x, y]] \text{ // RelinNormality} \\
\text{Out [41]} & = \text{GLB}[\text{cart}[x, y]] = \text{cart}[P[y], \text{intersection}[x, y]]
\end{align*}
\]

\[
\begin{align*}
\text{In [42]} & : = \text{GLB}[\text{cart}[x_\_, y_\_]] := \text{cart}[P[y], \text{intersection}[x, y]]
\end{align*}
\]

Similarly:
In[43]:= LUB[cart[x, y]] // RelnNormality

Out[43]= LUB[cart[x, y]] = cart[P[x], intersection[x, y]]

In[44]:= LUB[cart[x_, y_]] := cart[P[x], intersection[x, y]]