intersection[RFX,TRV]

Johan G. F. Belinfante
2004 December 18

In[1]:= SetDirectory["i:"]; << goedel164.17b; << tools.m

:Package Title: goedel164.17b 2004 December 17 at 9:30 p.m.

It is now: 2004 Dec 18 at 6:32

Loading Simplification Rules

TOOLS.M Revised 2004 December 16

weightlimit = 40

summary

Reflexive transitive relations, also known as quasi-orders, satisfy some properties analogous to those of equivalence relations. In this notebook, variable-free formulations of facts about the class intersection[RFX,TRV] of small reflexive symmetric relations are derived, culminating in a formula for a canonical factorization of such relations.

reflexive core of a transitive relation

If \( x \) is transitive, so is its reflexive core.

In[2]:= SubstTest[implies, TRANSITIVE[x], TRANSITIVE[restrict[x, y, y]], y \rightarrow fix[x]]

Out[2]= or[not[TRANSITIVE[x]], TRANSITIVE[rfx[x]]] := True

In[3]:= or[not[TRANSITIVE[x___]], TRANSITIVE[rfx[x___]]] := True

The class TRV is invariant under CORE[RFX].

In[4]:= Map[equal[0, composite[Id, complement[#]]] &, SubstTest[class, pair[x, z],

implies[member[z, image[u, singleton[setpart[x]]]], member[z, v]],

{u \rightarrow composite[HULL[TRV], IMAGE[cart[{V, V}]]],

v \rightarrow image[inverse[CORE[RFX], TRV]]}] // Reverse


In[5]:= % /. Equal \rightarrow SetDelayed
The image is also contained in `RFX`.

\[
\text{In}[6]:= \quad \text{Map[subclass[#}, \text{RFX}] \&, \text{ImageComp[id[RFX}, \text{CORE[RFX}, \text{TRV}]])}
\]

\[
\text{Out}[6]= \quad \text{subclass[image[CORE[RFX}, \text{TRV}], \text{RFX}] = True}
\]

\[
\text{In}[7]= \quad \% /. \text{Equal} \rightarrow \text{SetDelayed}
\]

The reverse inclusion also holds:

\[
\text{In}[8]= \quad \text{SubstTest[implies, subclass[u, v],}
\]

\[
\quad \text{subclass[image[u, w], image[v, w]], \{u \rightarrow \text{id[RFX}, v \rightarrow \text{CORE[RFX}, w \rightarrow \text{TRV}]\}}
\]

\[
\text{Out}[8]= \quad \text{subclass[intersection[RFX, TRV], image[CORE[RFX}, \text{TRV] = True}
\]

\[
\text{In}[9]= \quad \% /. \text{Equal} \rightarrow \text{SetDelayed}
\]

The following equation follows:

\[
\text{In}[10]= \quad \text{equal[image[CORE[RFX}, \text{TRV], intersection[RFX, TRV]]} \quad /\quad \text{AssertTest}
\]

\[
\text{Out}[10]= \quad \text{equal[image[CORE[RFX}, \text{TRV], intersection[RFX, TRV]] = True}
\]

\[
\text{In}[11]= \quad \text{image[CORE[RFX}, \text{TRV} := \text{intersection[RFX, TRV]}}
\]

---

an antitone property of vertical sections

The truth of the following statement is currently recognized by the GOEDEL program. It says that the vertical sections of a transitive relation satisfy an antitone property with respect to inclusions: if \( u \) is related to \( v \) by a transitive relation \( x \), then the vertical section of \( v \) is contained in that of \( u \).

\[
\text{In}[12]= \quad \text{or[not[member[pair[u, v], x]], not[TRANSITIVE[x]],}
\]

\[
\quad \text{subclass[image[x, singleton[v]], image[x, singleton[u]]}]}
\]

\[
\text{Out}[12]= \quad \text{True}
\]

A version of this fact which eliminates the variables \( u \) and \( v \) is readily derived in the case that \( x \) is thin:

\[
\text{In}[13]= \quad \text{Map[subclass[x, #] \&,}
\]

\[
\quad \text{composite[inverse[VERTSECT[x]], inverse[S], VERTSECT[x]]} \quad /\quad \text{RelnNormality}
\]

\[
\text{Out}[13]= \quad \text{subclass[x, composite[inverse[VERTSECT[x]], inverse[S], VERTSECT[x]] =}
\]

\[
\quad \text{and[equal[V, domain[VERTSECT[x]]], TRANSITIVE[x]]}
\]
The following corollary requires nothing further:

In[15] := subclass[x, composite[id[fix[x]],
   inverse[VERTSECT[x]], inverse[S], VERTSECT[x], id[fix[x]]]]

Out[15] := and[equal[V, domain[VERTSECT[x]]], REFLEXIVE[x], TRANSITIVE[x]]

It will be shown below that this inclusion can be replaced with an equation.

---

a factorization result

Recall that \texttt{composite[inverse[funpart[x]], inverse[S], funpart[x]]} is both reflexive and transitive. Variable-free restatements of these facts can be obtained as follows. For the reflexive property, one obtains:

In[16] := Map[equal[V, #] &,
   SubstTest[class, x, subclass[image[u, singleton[setpart[x]]]], v],
   {u -> composite[COMPOSE, intersection[composite[inverse[FIRST], INVERSE],
      composite[inverse[SECOND], IMAGE[cross[Id, inverse[S]]]]],
     FUNPART, v \rightarrow \text{RFX}]}) // Reverse

Out[16] := subclass[FUNS, fix[composite[INVERSE, inverse[image[inverse[COMPOSE], RFX]],
   IMAGE[cross[Id, inverse[S]]]]]] = True

In[17] := \% /. Equal \rightarrow \text{SetDelayed}

A similar result holds for the transitive property:

In[18] := Map[equal[V, #] &,
   SubstTest[class, x, subclass[image[u, singleton[setpart[x]]]], v],
   {u -> composite[COMPOSE, intersection[composite[inverse[FIRST], INVERSE],
      composite[inverse[SECOND], IMAGE[cross[Id, inverse[S]]]]],
     FUNPART, v \rightarrow \text{TRV}]}) // Reverse

Out[18] := subclass[FUNS, fix[composite[INVERSE, inverse[image[inverse[COMPOSE], TRV]],
   IMAGE[cross[Id, inverse[S]]]]]] = True

In[19] := \% /. Equal \rightarrow \text{SetDelayed}

These two results will now be solved to derive an lower bound for \texttt{intersection[RFX,-\text{TRV}]. Note that the following is a function:
Consequently, one has:

\[
\begin{align*}
\text{In[22]} & : \quad \text{abstract}[x, \\
& \quad \text{image}[\text{COMPOSE}, \text{composite}[\text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]], \text{id}[x], \text{INVERSE}]]) \\
\text{Out[22]} & : \quad \text{composite}[\text{COMPOSE}, \text{intersection}[\text{composite}[\text{inverse}[\text{FIRST}], \text{INVERSE}], \\
& \quad \text{composite}[\text{inverse}[\text{SECOND}], \text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]]]]) \\
\text{In[23]} & : \quad \text{SubstTest}[\text{composite}, \text{funpart}[x], \text{inverse}[\text{funpart}[x]], \\
& \quad x -> \text{composite}[\text{COMPOSE}, \text{intersection}[\text{composite}[\text{inverse}[\text{FIRST}], \text{INVERSE}], \\
& \quad \text{composite}[\text{inverse}[\text{SECOND}], \text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]]]]) \\
\text{Out[23]} & : \quad \text{composite}[\text{COMPOSE}, \text{intersection}[\text{composite}[\text{inverse}[\text{FIRST}], \text{INVERSE}], \\
& \quad \text{composite}[\text{inverse}[\text{SECOND}], \text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]]]), \\
& \quad \text{intersection}[\text{composite}[\text{INVERSE}, \text{FIRST}], \text{composite}[ \\
& \quad \text{inverse}[\text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]]], \text{SECOND}], \text{inverse}[\text{COMPOSE}]] = \\
& \quad \text{id}[\text{image}[\text{COMPOSE}, \text{composite}[\text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]], \text{INVERSE}]]] \\
\text{In[24]} & : \quad \%/. \text{Equal} \rightarrow \text{SetDelayed} \\
\text{Corollary:} \\
\text{In[25]} & : \quad \text{ImageComp}[\text{composite}[\text{COMPOSE}, \text{intersection}[\text{composite}[\text{inverse}[\text{FIRST}], \text{INVERSE}], \\
& \quad \text{composite}[\text{inverse}[\text{SECOND}], \text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]]]]) \\
& \quad \text{inverse}[\text{composite}[\text{COMPOSE}, \text{intersection}[ \\
& \quad \text{composite}[\text{inverse}[\text{FIRST}], \text{INVERSE}], \text{composite}[\text{inverse}[\text{SECOND}], \\
& \quad \text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]]]]), x] /\text{Reverse} \\
\text{Out[24]} & : \quad \text{image}[\text{COMPOSE}, \text{composite}[\text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]]], \\
& \quad \text{id}[\text{fix}[\text{composite}[\text{INVERSE}, \text{inverse}[\text{image}[\text{inverse}[\text{COMPOSE}], x]], \\
& \quad \text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]])], \text{INVERSE}]] = \text{intersection}[x, \\
& \quad \text{image}[\text{COMPOSE}, \text{composite}[\text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]], \text{INVERSE}]]] \\
\text{This is made into a temporary rewrite rule:} \\
\text{In[26]} & : \quad \text{image}[\text{COMPOSE}, \text{composite}[\text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]]], \\
& \quad \text{id}[\text{fix}[\text{composite}[\text{INVERSE}, \text{inverse}[\text{image}[\text{inverse}[\text{COMPOSE}], x]], \\
& \quad \text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]])], \text{INVERSE}]] = \text{intersection}[x, \\
& \quad \text{image}[\text{COMPOSE}, \text{composite}[\text{IMAGE}[\text{cross}[\text{Id}, \text{inverse}[S]]], \text{INVERSE}]]] \\
\text{The statement obtained for the reflexive property is then transformed as follows:}
A similar restatement is obtained for the transitive property:

Conversely, for any thin reflexive transitive relation \( x \) there is a canonical factorization of the form \( x = \text{composite}[\text{inverse}[\text{funpart}[y]], \text{inverse}[S], \text{funpart}[y]] \) where \( y \) is the restriction of the vertical section function to \( \text{fix}[x] \). For sets, this result is:

The variable \( x \) can be eliminated as follows.
The wrapper-related functions \texttt{CORE[RFX]} and \texttt{composite[HULL[TRV], IMAGE[id[cart[V,V]]]]} can be eliminated using \texttt{ImageComp}:

\begin{verbatim}
In[33]:= Map[subclass[intersection[RFX, TRV], #] &, 
    ImageComp[composite[CORE[RFX], HULL[TRV], IMAGE[id[cart[V, V]]]], 
    inverse[composite[CORE[RFX], HULL[TRV], IMAGE[id[cart[V, V]]]], 
    fix[composite[COMPOSE, intersection[composite[inverse[FIRST], INVERSE], 
    composite[inverse[SECOND], IMAGE[cross[Id, inverse[S]]]], VS]]] 
    ]
\end{verbatim}

\begin{verbatim}
Out[33]= subclass[intersection[RFX, TRV], 
    fix[composite[COMPOSE, intersection[composite[inverse[FIRST], INVERSE], 
    composite[inverse[SECOND], IMAGE[cross[Id, inverse[S]]]], VS]]] ]
\end{verbatim}

The next step uses the fact that \texttt{VS} is contained in \texttt{cart[V, FUNS]}. One needs this relation:

\begin{verbatim}
In[35]:= abstract[x, 
    fix[composite[COMPOSE, intersection[composite[inverse[FIRST], INVERSE], 
    composite[inverse[SECOND], IMAGE[cross[Id, inverse[S]]]], x]]] 
\end{verbatim}

\begin{verbatim}
Out[35]= composite[FIRST, 
    id[composite[intersection[composite[INVERSE, FIRST], composite[
    inverse[IMAGE[cross[Id, inverse[S]]]], SECOND]], inverse[COMPOSE]]]]
\end{verbatim}

The result is this inclusion:
Combining this inclusion with the one derived for \texttt{intersection[RFX, TRV]} yields:

\begin{verbatim}
In[38]:= SubstTest[implies, andsubclass[u, v], subclass[v, w]], subclass[u, w],
    \{u -> intersection[RFX, TRV],
        v -> fix[composite[COMPOSE, intersection[compose[inverse[FIRST], INVERSE],
            compose[inverse[SECOND], IMAGE[cross[Id, inverse[S]]]]], VS]],
    w -> image[COMPOSE, composite[IMAGE[cross[Id, inverse[S]]]],
        id[FUNS], INVERSE]]
Out[38]= subclass[intersection[RFX, TRV], image[COMPOSE,
    composite[IMAGE[cross[Id, inverse[S]]], id[FUNS], INVERSE]]] = True
\end{verbatim}

The reverse inclusion was proved in the preceding section. Combining these, one obtains a variable-free equation which says that (small) reflexive transitive relations can be characterized by the existence of a factorization of the form \texttt{compose[inverse[f], inverse[S], f]} where \texttt{f} is a function.

\begin{verbatim}
In[40]:= SubstTest[and, subclass[u, v], subclass[v, u],
    \{u -> intersection[RFX, TRV], v -> image[COMPOSE,
        composite[IMAGE[cross[Id, inverse[S]]], id[FUNS], INVERSE]]\}]
Out[40]= True = equal[
    image[COMPOSE, composite[IMAGE[cross[Id, inverse[S]]], id[FUNS], INVERSE]],
    intersection[RFX, TRV]]
In[41]:= image[COMPOSE, composite[IMAGE[cross[Id, inverse[S]]], id[FUNS], INVERSE]] :=
    intersection[RFX, TRV]
\end{verbatim}

Note the resemblance of this formula with a corresponding one for the class of equivalence relations:
In[42]:  = image[COMPOSE, composite[id[FUNS], INVERSE]]

out[42]=  EQV