erasing

Johan G. F. Belinfante
2007 October 18

In[1]: =  SetDirectory["1:" ]; << goedel98.16a; << tools.m

:Package Title: goedel98.16a        2007 October 16 at 8:40 a.m.

It is now: 2007 Oct 18 at 23:55

Loading Simplification Rules

TOOLS.M Revised 2007 September 19

weightlimit = 40

summary

Given a set $x$, one can erase parts of it in various ways to obtain a subset $y$. For each type of erasure it is interesting to study its erasure relation consisting of all ordered pairs $\text{pair}(x, y)$ such that $y$ can be obtained from $x$ by erasure in a specified way. If no restriction whatsoever is placed on the erasure process, so that any subset can be obtained, the erasure relation is just the inverse of the subset relation $S$.

In[2]: =  class[ pair[x,y], subclass[y, x] ]


An important type of erasure in the theory of relations is restriction. By erasing vertically, one can obtain from a given relation $x$ its restrictions, that is, subrelations of the form $\text{composite}[x, \text{id}[v]]$. The erasure relation for this type of erasure has a special name in the GOEDEL program:

In[3]: =  class[ pair[x, y], assert[exists[v, equal[y, composite[x, id[v]]]]]]

Out[3]=  RESTRICT

Erasing both horizontally and vertically yields relations of the form $\text{composite}[\text{id}[u], x, \text{id}[v]]$. For the theory of partial orders, the symmetric case $u = v$ is important. The erasure relations for each of these types of erasures may have formal properties such as transitivity, for which one can derive rewrite rules. It would also be desirable to have rewrite rules to help reduce the myriad formulas for specific erasure processes to obtain standard forms by which one can refer to them. Many types of erasure have special names, but it is probably best to be conservative about the number of new names to be introduced into the GOEDEL program, partly because the names are not completely standard, but also because there may be rewrite rules common to several forms of erasure. Introducing separate names for all the various forms of erasure would introduce unnecessary duplication.
introduction

Lemma.

\begin{verbatim}
In[4]:= composite[CAP, id[cart[V, x]], inverse[FIRST]] // VSNormality // Reverse
Out[4]= intersection[fix[composite[inverse[DIF], DISJOINT, id[x], S, SECOND]], inverse[S]] :=
            composite[CAP, id[cart[V, x]], inverse[FIRST]]
In[5]:= intersection[fix[composite[inverse[DIF], DISJOINT, id[x_], S, SECOND]], inverse[S]] :=
            composite[CAP, id[cart[V, x]], inverse[FIRST]]
\end{verbatim}

The types of erasure considered in the present notebook are mainly of the general form \( y = \text{intersection}[x, t] \) where \( t \) can be any member of some specified class \( z \). The corresponding erasure relation is:

\begin{verbatim}
In[6]:= class[pair[x, y], exists[t, and[member[t, z], equal[y, intersection[x, t]]]]]
\end{verbatim}

A slight generalization of this is obtained by requiring the set \( x \) to belong to a specified class \( w \).

\begin{verbatim}
In[7]:= class[pair[x, y],
           and[member[x, w], exists[t, and[member[t, z], equal[y, intersection[x, t]]]]]]
Out[7]= composite[CAP, id[cart[w, z]], inverse[FIRST]]
\end{verbatim}

One could of course consider still more general erasure relations of the form \( \text{composite}[CAP, id[x], inverse[FIRST]] \), but most of the rewrite rules derived below fail at this higher level of generality. The rules derived in the next section are an exception to this.

antisymmetry

Lemma.

\begin{verbatim}
In[8]:= fix[composite[inverse[CAP], inverse[S], FIRST]] // InvertFixTest
Out[8]= fix[composite[inverse[CAP], inverse[S], FIRST]] = cart[V, V]
In[9]:= fix[composite[inverse[CAP], inverse[S], FIRST]] := cart[V, V]
\end{verbatim}

Theorem.

\begin{verbatim}
In[10]:= subclass[composite[FIRST, id[x], inverse[CAP]], S] // AssertTest
In[11]:= subclass[composite[FIRST, id[x_], inverse[CAP]], S] := True
\end{verbatim}
Corollary. Any subclass of $S$ is an antisymmetric relation. The following rewrite rule is slightly more general than necessary:

$$\text{In [12]}: \quad \text{SubstTest[subclass, w, intersection[u, v],}
\quad \{w \rightarrow \text{intersection[composite[CAP, id[x], inverse[FIRST]],}
\quad \text{composite[FIRST, id[y], inverse[CAP]]], u \rightarrow S, v \rightarrow inverse[S]]} \quad \text{// Reverse}
$$

$$\text{Out[12]}: \quad \text{subclass[intersection[composite[CAP, id[x], inverse[FIRST]],}
\quad \text{composite[FIRST, id[y], inverse[CAP]]], Id] = True}
$$

$$\text{In[13]}: \quad \text{subclass[intersection[composite[CAP, id[x_], inverse[FIRST]],}
\quad \text{composite[FIRST, id[y_], inverse[CAP]]], Id] := True}
$$

---

**normalization**

The GOEDEL program already contains the following rewrite rule for standardizing one variant expression for the type of erasure under consideration:

$$\text{In [14]}: \quad \text{composite[CAP, id[cart[x, y]], inverse[SECOND]]}
$$

$$\text{Out[14]}: \quad \text{composite[CAP, id[cart[y, x]], inverse[FIRST]]}
$$

Other formulas can be derived using various Normality tests. Some of these may strike one as somewhat strange; for example:

$$\text{In [15]}: \quad (\text{composite[CAP, id[x], inverse[FIRST]]} \quad \text{// ReifNormality} \quad \text{// Reverse}) / . x \rightarrow V
$$

$$\text{Out[15]}: \quad \text{composite[IMG, inverse[FIRST], IMAGE[DUP]] = inverse[S]}
$$

$$\text{In[16]}: \quad \text{composite[IMG, inverse[FIRST], IMAGE[DUP]] := inverse[S]}
$$

The use of ReifNormality yields this rewrite rule:

$$\text{In [17]}: \quad \text{composite[CAP, id[cart[V, x]], inverse[FIRST]]} \quad \text{// ReifNormality} \quad \text{// Reverse}
$$

$$\text{Out[17]}: \quad \text{composite[IMG, id[cart[V, x]], inverse[FIRST], IMAGE[DUP]] =}
\quad \text{composite[CAP, id[cart[V, x]], inverse[FIRST]]}
$$

$$\text{In[18]}: \quad \text{composite[IMG, id[cart[V, x_]], inverse[FIRST], IMAGE[DUP]] :=}
\quad \text{composite[CAP, id[cart[V, x]], inverse[FIRST]]}
$$

Yet another such expression is this:

$$\text{In [19]}: \quad \text{intersection[composite[SECOND, id[composite[inverse[S], id[x]]]],}
\quad \text{inverse[DIF], DISJOINT, inverse[S]]} \quad \text{// ReifNormality}
$$

$$\text{Out[19]}: \quad \text{intersection[composite[SECOND, id[composite[inverse[S], id[x]]], inverse[DIF], DISJOINT],}
\quad \text{inverse[S]] = composite[CAP, id[cart[V, x]], inverse[FIRST]]}
$$
In general one could perform two types of erasures, one after the other, say with parameters \( x \) and \( y \). For the types of erasure under consideration, the composite is again an erasure of the type under consideration with parameter \( \text{image}[\text{CAP}, \text{cart}[x, y]] \). Note that the order is not important because the symmetry of the function \( \text{CAP} \) implies that this expression is symmetric in \( x \) and \( y \). A slightly more general rule can be derived:

\[
\text{In}[20]:= \text{intersection[composite}[\text{SECOND}, \text{id[composite}[\text{inverse}[S], \text{id}[x_]]], \text{inverse}[\text{DIF}], \text{DISJOINT}], \text{inverse}[S]] := \text{composite[CAP, id[cart[V, x]], inverse[FIRST]]}
\]

And here is still one more:

\[
\text{In}[21]:= \text{composite}[\text{IMG}, \text{id[cart[image][\text{IMAGE}[\text{DUP}], x], V]], inverse[SECOND]] \text{ // ReifNormality}
\]

\[
\text{Out}[21]= \text{composite}[\text{CAP, id[cart[\text{V}, x]], inverse[SECOND]]} = \text{composite[CAP, id[cart[V, x]], inverse[FIRST]]}
\]

\[
\text{In}[22]:= \text{composite}[\text{IMG}, \text{id[cart[image][\text{IMAGE}[\text{DUP}], x_], V]], inverse[SECOND]] := \text{composite[CAP, id[cart[V, x]], inverse[FIRST]]}
\]

### composite erasures

Theorem.

In general one could perform two types of erasures, one after the other, say with parameters \( x \) and \( y \). For the types of erasure under consideration, the composite is again an erasure of the type under consideration with parameter \( \text{image}[\text{CAP}, \text{cart}[x, y]] \). Note that the order is not important because the symmetry of the function \( \text{CAP} \) implies that this expression is symmetric in \( x \) and \( y \). A slightly more general rule can be derived:

\[
\text{In}[23]:= \text{Assoc[CAP, composite[cross[Id, CAP], \text{ASSOC}], composite[}
\text{composite[Id[cart[\text{cart}[x, y], z]], inverse[\text{FIRST]}], inverse[\text{FIRST}]]] \text{ // Reverse}
\]

\[
\text{Out}[23]= \text{composite[CAP, id[cart[V, z]], inverse[FIRST], CAP, id[cart[x, y]], inverse[FIRST]} = \text{composite[CAP, id[cart[x, image[\text{CAP}, \text{cart[y, z]]]}], inverse[\text{FIRST}]]}
\]

\[
\text{In}[24]:= \text{composite[CAP, id[cart[V, z_]], inverse[\text{FIRST}], CAP, id[cart[x_, y_]],}
\text{inverse[\text{FIRST}]]} := \text{composite[CAP, id[cart[x, image[\text{CAP}, \text{cart[y, z]]]}], inverse[\text{FIRST}]]}
\]

### transitive laws

Lemma.

\[
\text{In}[25]:= \text{SubstTest[implies, subclass[x, y], subclass[image[t, x], image[t, y]]},
\text{t -> composite[intersection[composite[\text{inverse[\text{FIRST}], inverse[w]}],}
\text{composite[\text{inverse[\text{SECOND}], u}], id[cart[\text{V}, \text{V}]], inverse[\text{SECOND}]]]] \text{ // Reverse}
\]

\[
\text{Out}[25]= \text{or[not[subclass[x, y]]], subclass[composite[u, id[cart[v, x]], w], composite[u, id[cart[v, y]], w]]} = \text{True}
\]

\[
\text{In}[26]:= \text{or[not[subclass[x_, y_]]}, \text{subclass[}
\text{composite[u_, id[\text{cart}[v_], x_]], w_], \text{composite[u_, id[\text{cart}[v_], y_]], w_]]} = \text{True}
\]

Theorem.
In[27]:= \[\text{Map[implies[subclass[image[CAP, cart[x, x]], x], \# \&, SubstTest[subclass, composite[t, t], t \rightarrow \text{composite[CAP, id[cart[V, x]], inverse[\text{FIRST}]]}]]}
\]

Out[27]= or[not[subclass[image[CAP, cart[x, x]], x]],
\text{TRANSITIVE[composite[CAP, id[cart[V, x]], inverse[\text{FIRST}]]]} = \text{True}

In[28]:= (% /. x \rightarrow x_) /. \text{Equal} \rightarrow \text{SetDelayed}

The following lemma can be used to generalize this result slightly.

In[29]:= SubstTest[implies, \text{TRANSITIVE[t], TRANSITIVE[composite[t, id[x]]]},
\{t \rightarrow \text{composite[CAP, id[cart[V, y]], inverse[\text{FIRST}]]}\} \text{ // Reverse}

Out[29]= or[not[\text{TRANSITIVE[composite[CAP, id[cart[V, y]], inverse[\text{FIRST}]]]}],
\text{TRANSITIVE[composite[CAP, id[cart[x, y]], inverse[\text{FIRST}]]]} = \text{True}

In[30]:= (% /. \{x \rightarrow x_, y \rightarrow y_\}) /. \text{Equal} \rightarrow \text{SetDelayed}

Theorem.

In[31]:= \[\text{Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3]},
\text{not[implies[p1, p3]]}, \{p1 \rightarrow \text{subclass[image[CAP, cart[y, y]], y]},
p2 \rightarrow \text{TRANSITIVE[composite[CAP, id[cart[V, y]], inverse[\text{FIRST}]]]},
p3 \rightarrow \text{TRANSITIVE[composite[CAP, id[cart[x, y]], inverse[\text{FIRST}]]]}\} \text{ // Reverse}
\]

Out[31]= or[not[subclass[image[CAP, cart[y, y]], y]],
\text{TRANSITIVE[composite[CAP, id[cart[x, y]], inverse[\text{FIRST}]]]} = \text{True}

In[32]:= or[not[subclass[image[CAP, cart[y_, y_]], y_]],
\text{TRANSITIVE[composite[CAP, id[cart[x_, y_]], inverse[\text{FIRST}]]]} = \text{True}

Transitive laws for two important special cases are immediate corollaries.

In[33]:= SubstTest[implies, subclass[image[CAP, cart[y, y]], y], \text{TRANSITIVE[}
\\text{composite[CAP, id[cart[x, y]], inverse[\text{FIRST}]]}, y \rightarrow \text{range[\text{CART}]}\} \text{ // Reverse}

Out[33]= \text{TRANSITIVE[composite[CAP, id[cart[x, range[\text{CART}]]], inverse[\text{FIRST}]]} = \text{True}

In[34]:= \text{TRANSITIVE[composite[CAP, id[cart[x_, range[\text{CART}]]], inverse[\text{FIRST}]]} = \text{True}

Corollary.

In[35]:= SubstTest[implies, subclass[image[CAP, cart[y, y]], y], \text{TRANSITIVE[}
\\text{composite[CAP, id[cart[x, y]], inverse[\text{FIRST}]]}, y \rightarrow \text{image[\text{CART, Id}]}\} \text{ // Reverse}

Out[35]= \text{TRANSITIVE[composite[CAP, id[cart[x, image[\text{CART, Id}]]], inverse[\text{FIRST}]]} = \text{True}

In[36]:= \text{TRANSITIVE[composite[CAP, id[cart[x_, image[\text{CART, Id}]]], inverse[\text{FIRST}]]} = \text{True}
reflexivity of erasure relations

A general condition for reflexivity can be derived:

\[ \text{In } [37] := \text{REFLEXIVE}[\text{composite}[\text{CAP}, \text{id}[x], \text{inverse}[\text{FIRST}]]] // \text{AssertTest} \]

\[ \text{Out}[37] = \text{REFLEXIVE}[\text{composite}[\text{CAP}, \text{id}[x], \text{inverse}[\text{FIRST}]]] =\]
\[ \text{and}[[\text{subclass}[\text{domain}[x], \text{fix}[\text{composite}[\text{inverse}[S], x]]], \]
\[ \text{subclass}[\text{image}[\text{CAP}, x], \text{fix}[\text{composite}[\text{inverse}[S], x]]]]] \]

\[ \text{In}[38] := \text{REFLEXIVE}[\text{composite}[\text{CAP}, \text{id}[x_{-}], \text{inverse}[\text{FIRST}]]] :=\]
\[ \text{and}[[\text{subclass}[\text{domain}[x], \text{fix}[\text{composite}[\text{inverse}[S], x]]], \]
\[ \text{subclass}[\text{image}[\text{CAP}, x], \text{fix}[\text{composite}[\text{inverse}[S], x]]]]] \]

Note that the reflexive property fails for the two main types of erasure under consideration if one does not restrict the erasure process to relations:

\[ \text{In}[39] := \text{REFLEXIVE}[\text{composite}[\text{CAP}, \text{id}[\text{cart}[V, \text{range}[\text{CART}]]], \text{inverse}[\text{FIRST}]]] \]
\[ \text{Out}[39] = \text{False} \]

\[ \text{In}[40] := \text{REFLEXIVE}[\text{composite}[\text{CAP}, \text{id}[\text{cart}[V, \text{image}[\text{CART, Id}]]], \text{inverse}[\text{FIRST}]]] \]
\[ \text{Out}[40] = \text{False} \]

One can fix this up by considering the restriction of these erasure relations to the class \( P[\text{cart}[V, V]] \) of relations.

\[ \text{In}[41] := \text{REFLEXIVE}[\text{composite}[\text{CAP}, \text{id}[\text{cart}[P[\text{cart}[V, V]], \text{range}[\text{CART}]]], \text{inverse}[\text{FIRST}]]] \]
\[ \text{Out}[41] = \text{True} \]

\[ \text{In}[42] := \text{REFLEXIVE}[\text{composite}[\text{CAP}, \text{id}[\text{cart}[P[\text{cart}[V, V]], \text{image}[\text{CART, Id}]]], \text{inverse}[\text{FIRST}]]] \]
\[ \text{Out}[42] = \text{True} \]

two examples of interest

Theorem. Horizontal and vertical erasure is a partial ordering on the class of relations.

\[ \text{In}[43] := \text{SubstTest[and, \text{REFLEXIVE}[t], \text{TRANSITIVE}[t], \text{subclass}[\text{intersection}[t, \text{inverse}[t]], \text{Id}], t \to \text{composite}[\text{CAP}, \text{id}[\text{cart}[P[\text{cart}[V, V]], \text{range}[\text{CART}]]], \text{inverse}[\text{FIRST}]]]} \]
\[ \text{Out}[43] = \text{PARTIALORDER[}\]
\[ \text{composite}[\text{CAP}, \text{id}[\text{cart}[P[\text{cart}[V, V]], \text{range}[\text{CART}]]], \text{inverse}[\text{FIRST}]]] = \text{True} \]

\[ \text{In}[44] := \text{PARTIALORDER[}\]
\[ \text{composite}[\text{CAP}, \text{id}[\text{cart}[P[\text{cart}[V, V]], \text{range}[\text{CART}]]], \text{inverse}[\text{FIRST}]]] := \text{True} \]

Theorem. Symmetric erasure defines a partial ordering on the class \( \text{PO} \) of partial orders.
In[45]:= SubstTest[and, REFLEXIVE[t], TRANSITIVE[t], subclass[intersection[t, inverse[t]], Id], 
t -> composite[CAP, id[cart[PO, image[CART, Id]]], inverse[FIRST]]]

Out[45]= PARTIALORDER[composite[CAP, id[cart[PO, image[CART, Id]]], inverse[FIRST]]] = True

In[46]:= PARTIALORDER[composite[CAP, id[cart[PO, image[CART, Id]]], inverse[FIRST]]] := True

Note that the fixed point set of this ordering is indeed the class PO.

In[47]:= fix[composite[CAP, id[cart[PO, image[CART, Id]]], inverse[FIRST]]]

Out[47]= PO