Cartesian Squares

Johan G. F. Belinfante
2001 October 20

\[
\text{\texttt{\textless\textless goedel52.k87; \textless\textless tests.m}}
\]

:Package Title: GOEDEL52.K87 2001 October 20 at 7:55 a.m.

It is now: 2001 Oct 20 at 11:14

Loading Simplification Rules

TESTS.M Revised 2001 October 18

\[\text{weightlimit} = 30\]

Context switch to 'Goedel'Private is needed for ReplaceTest

Just ignore the error message about Unterminated use of BeginPackage

Get::bebal : Unterminated uses of BeginPackage or Begin in \texttt{\textless\textless tests.m}.

\section*{A temporary rule}

\begin{verbatim}
subclass[cart[singleton[0], singleton[0]], Id] // AssertTest
subclass[cart[singleton[0], singleton[0]], Id] == True
subclass[cart[singleton[0], singleton[0]], Id] := True
SubstTest[implies, subclass[u, v], subclass[image[CART, u], image[CART, v]],
\{u -> id[singleton[0]], v -> Id\}]
member[0, image[CART, Id]] == True
\end{verbatim}

We add this as a temporary rule...

\begin{verbatim}
member[0, image[CART, Id]] := True
\end{verbatim}

\begin{verbatim}
equal[union[image[CART, Id], singleton[0]], image[CART, Id]]
\end{verbatim}

\begin{verbatim}
True
\end{verbatim}

This too:

\begin{verbatim}
union[image[CART, Id], singleton[0]] := image[CART, Id]
\end{verbatim}
■ A Normality Test

\[\text{fix[composite[CART, DUP, IMAGE[inverse[DUP]]]]} /\text{ Normality // Reverse}\]

\[\text{intersection[RFX, fix[composite[S, CART, DUP, IMAGE[inverse[DUP]]]]]} ==\]

\[\text{fix[composite[CART, DUP, IMAGE[inverse[DUP]]]]}\]

\[\text{intersection[RFX, fix[composite[S, CART, DUP, IMAGE[inverse[DUP]]]]]} ==\]

\[\text{fix[composite[CART, DUP, IMAGE[inverse[DUP]]]]}\]

■ Corollary of the axiom of replacement

We derive a corollary of the axiom of replacement. First we consider the case that \(x\) is a set:

\[\text{forall} [x, \text{implies} [\text{member}[x, V], \text{member}\left[\text{image}\left[\text{funpart}[y], x\right], V]\right]] /\text{ assert}\]

\[\text{True}\]

The case that \(x\) is a proper class is done separately:

\[\text{member}[p, V] := \text{False}\]

\[\text{implies} [\text{member}[x, V], \text{member}\left[\text{image}\left[\text{funpart}[y], x\right], V\right]] / . x -> p\]

\[\text{True}\]

This proves:

\[\text{implies} [\text{member}[x, V], \text{member}\left[\text{image}\left[\text{funpart}[y], x\right], V\right]] == \text{True}\]

\[\text{or} [\text{member}\left[\text{image}\left[\text{funpart}[y], x\right], V\right], \text{not}\left[\text{member}[x, V]\right]] == \text{True}\]

This is a permanent new rule:

\[\text{or} [\text{member}\left[\text{image}\left[\text{funpart}[y], x\right], V\right], \text{not}\left[\text{member}[x, V]\right]] := \text{True}\]

From this new rule we deduce a corollary:

\[\text{SubstTest} [\text{implies}, \text{member}\left[x, V\right], \text{member}\left[\text{image}\left[\text{funpart}[y], x\right], V\right], y -> \text{inverse[DUP]}]\]

\[\text{or} [\text{member}\left[\text{fix}[x], V\right], \text{not}\left[\text{member}[x, V]\right]] == \text{True}\]

This is another candidate for a permanent rule:

\[\text{or} [\text{member}\left[\text{fix}[x], V\right], \text{not}\left[\text{member}[x, V]\right]] := \text{True}\]

\[\text{equiv} [\text{and} [\text{member}\left[x, V\right], \text{member}\left[\text{fix}[x], V\right]], \text{member}[x, V]] /\text{ assert}\]

\[\text{True}\]

\[\text{and} [\text{member}\left[x, V\right], \text{member}\left[\text{fix}[x], V\right]] := \text{member}[x, V]\]
Two descriptions of the class of cartesian squares

The following inverse image formula is the key to finding a connection between two descriptions of the class of cartesian squares.

\[ \text{image}[^{\text{inverse[CART]}}, \text{fix}[\text{composite[CART, DUP, IMAGE[inverse[DUP]]]]}] \text{ // VSNormality} \]

\[ \text{image}[^{\text{inverse[CART]}}, \text{fix}[\text{composite[CART, DUP, IMAGE[inverse[DUP]]]]}] = \text{union}\{\text{Id, cart}[V, \text{singleton[0]]}, \text{cart}[\text{singleton[0]}, V]\} \]

We add this as a temporary rule:

\[ \text{image}[^{\text{inverse[CART]}}, \text{fix}[\text{composite[CART, DUP, IMAGE[inverse[DUP]]]]}] := \text{union}\{\text{Id, cart}[V, \text{singleton[0]]}, \text{cart}[\text{singleton[0]}, V]\} \]

The following formula is needed to go further:

\[ \text{keyformula} := \text{ImageComp[CART, inverse[CART]}], \text{fix}[\text{composite[CART, DUP, IMAGE[inverse[DUP]]]]}] \]

First we derive a key membership rule:

\[ \text{Map}\{\text{member}[x, \#] \&, \text{keyformula}\} \text{ // Reverse} \]

\[ \text{member}[x, \text{image}[\text{CART}, \text{Id}]] = \text{and}[\text{equal}[x, \text{cart}[\text{fix}[x], \text{fix}[x]]], \text{member}[x, V]] \]

This membership rule can be made permanent:

\[ \text{member}[x, \text{image}[\text{CART}, \text{Id}]] := \text{and}[\text{equal}[x, \text{cart}[\text{fix}[x], \text{fix}[x]]], \text{member}[x, V]] \]

The key formula could be derived from this rule!

\[ \text{image}[\text{CART, Id}] \text{ // Normality \text{ // Reverse} \}

\[ \text{fix}[\text{composite[CART, DUP, IMAGE[inverse[DUP]]]]} = \text{image}[\text{CART, Id}] \]

\[ \text{fix}[\text{composite[CART, DUP, IMAGE[inverse[DUP]]]]} := \text{image}[\text{CART, Id}] \]

What about our temporary rules?

Let's remove the temporary rule about the empty set being a cartesian square:

\[ \text{member}[0, \text{image}[\text{CART, Id}]] = \text{.} \]

It is not needed anymore because of the more general rule that we added.

\[ \text{member}[0, \text{image}[\text{CART, Id}]] \]

\[ \text{True} \]

Another temporary rule can be replaced with a more general result:
subclass[cart[singleton[x], singleton[y]], Id] // AssertTest

subclass[cart[singleton[x], singleton[y]], Id] ==
or[equal[x, y], not[member[x, V]], not[member[y, V]]]

subclass[cart[singleton[x], singleton[y]], Id] :=
or[equal[x, y], not[member[x, V]], not[member[y, V]]]