
Johan G. F. Belinfante
2006 December 6

In[1] := SetDirectory["l:" ]; << goedel88.05a; << tools.m

:Package Title: goedel88.05a 2006 December 5 at 3:45 p.m.

It is now: 2006 Dec 7 at 12:11

Loading Simplification Rules

TOOLS.M Revised 2006 December 5

weightlimit = 40

summary

In this notebook, the binary function MIXMUL for mixed multiplication of integers by natural numbers yielding integers is introduced, and some of its basic properties are derived. Traditionally multiplication is often defined in terms of iterated addition, but if one does so, then one must deal with the fairly complex details of the iteration process every time that the definition of multiplication is accessed, dramatically slowing down all applications of the Gödel algorithm for expressions involving multiplication. For this reason, mixed multiplication is here defined by its properties, rather than by a specific construction. Specifically, an integer $z$ is defined to be the product of an integer $x$ and a natural number $y$ if there exists an addition-preserving mapping $w$ from the set $\text{omega}$ of natural numbers to the set $Z$ of integers which takes the natural number $1 = \text{set}[0]$ to the integer $x$ and takes the natural number $y$ to the integer $z$. One might paraphrase this definition of multiplication loosely as follows: $x \cdot y = z$ $\leftrightarrow$ $x \cdot 1 = z \cdot y$. A class-wrapped membership rule for this function has been introduced:

In[2] := Begin["Goedel`Private`"];

In[3] := FirstMatch[class[t_, member[w_, HoldPattern[MIXMUL]]]]

Out[3] := class[u_, member[v_, MIXMUL]] := ReleaseHold[Module[{w = Unique[], x = Unique[], y = Unique[], z = Unique[]} , class[u, exists[w, x, y, z, and[ member[w, binhom[NATADD, INTADD]]], equal[v, pair[pair[x, y], z]], equal[ set[x], image[w, set[set[0]]]], equal[set[z], image[w, set[y]]]]]]]

normalization for MIXMUL

The function MIXMUL can be related to MIXTIMES as follows:


The definition of mixed multiplication avoids explicit mention of iteration, but iteration does enter directly or indirectly into certain proofs. The idea is to speed up applications of Gödel's algorithm by defining multiplication by its properties, relegating the role of iteration to the proofs of some existence and uniqueness theorems. In particular, the uniqueness theorem for iteration was used to show that MIXTIMES is a function, and from that fact it now follows immediately that MIXMUL is also a function.

domains of binary homomorphisms

Lemma. (The function IMAGE[FIRST] assigns to each set its domain. Since all members of map[x,y] have the same domain x, the restriction of IMAGE[FIRST] to map[x,y] is a constant function.)

In[8]:= equal[composite[IMAGE[FIRST], id[map[x, y]]], cart[map[x, y], set[x]]] // AssertTest
Out[8]= equal[cart[map[x, y], set[x]], composite[IMAGE[FIRST], id[map[x, y]]]] = True

Out[9]= composite[IMAGE[FIRST], id[map[x, y]]] := cart[map[x, y], set[x]]

Lemma. (An equation recognized to be true is made into a new rewrite rule.)

In[10]:= equal[intersection[binhom[x, y], map[fix[domain[x]], fix[domain[y]]]], binhom[x, y]]
Out[10]= True

In[11]:= intersection[binhom[x_, y_], map[fix[domain[x_]], fix[domain[y_]]]] := binhom[x, y]

Theorem. (All binary homomorphisms from x to y have the same domain fix[domain[x]].)

In[12]:= Assoc[IMAGE[FIRST], id[map[fix[domain[x]], fix[domain[y]]]], id[binhom[x, y]]]
Out[12]= composite[IMAGE[FIRST], id[binhom[x, y]]] = cart[binhom[x, y], set[fix[domain[x]]]]

In[13]:= composite[IMAGE[FIRST], id[binhom[x_, y_]]] := cart[binhom[x, y], set[fix[domain[x]]]]

FUNPART theorem

Theorem.
In[14]:=  composite[FUNPART, id[binom[x, y]]] // ReifNormality
Out[14]=  composite[FUNPART, id[binom[x, y]]] = id[binom[x, y]]

In[15]:=  composite[FUNPART, id[binom[x_, y_]]] := id[binom[x, y]]

Corollary.

In[16]:=  Assoc[FUNPART, id[range[MIXTIMES]], MIXTIMES]
Out[16]=  composite[FUNPART, MIXTIMES] = MIXTIMES

In[17]:=  composite[FUNPART, MIXTIMES] := MIXTIMES

domain of MIXTIMES

Theorem.

In[18]:=  Assoc[composite[IMAGE[FIRST]], id[binom[NATADD, INTADD]], inverse[eval[set[0]]]]

In[19]:=  composite[IMAGE[FIRST], MIXTIMES] := cart[Z, set[omega]]

Corollary.

In[20]:=  IminComp[composite[inverse[SINGLETON], IMG], cross[MIXTIMES, SINGLETON], V]

In[21]:=  domain[MIXMUL] := cart[Z, omega]

Corollary.

In[22]:=  Assoc[MIXMUL, id[domain[MIXMUL]], id[cart[V, V]]] // Reverse
Out[22]=  composite[MIXMUL, id[cart[V, V]]] = MIXMUL

In[23]:=  composite[MIXMUL, id[cart[V, V]]] := MIXMUL

Corollary. (A function is a set if and only if its domain is a set.)

In[24]:=  member[MIXMUL, V] // AssertTest

In[25]:=  member[MIXMUL, V] := True
curry results relating MIXMUL and MIXTIMES

Theorem. (A variant of the formula connecting MIXMUL with MIXTIMES.)

\( \text{In [26]} := \text{Assoc[rotate[E], cross[FUNPART, Id], cross[MIXTIMES, Id]]} \)

\( \text{Out [26]} = \text{composite[rotate[E], cross[MIXTIMES, Id]]} = \text{MIXMUL} \)

\( \text{In [27]} := \text{composite[rotate[E], cross[MIXTIMES, Id]]} := \text{MIXMUL} \)

Theorem. (Yet another variant.)

\( \text{In [28]} := (\text{composite[rotate[E], cross[x, Id]]}) / \text{TripleRotate} \� \text{Reverse} / \text{x} \rightarrow \text{MIXTIMES} \)

\( \text{Out [28]} = \text{rotate[composite[inverse[MIXTIMES], E]]} = \text{MIXMUL} \)

\( \text{In [29]} := \text{rotate[composite[inverse[MIXTIMES], E]]} := \text{MIXMUL} \)

Theorem.

\( \text{In [30]} := \text{ApComp[composite[IMAGE[inverse[ASSOC]], IMAGE[cross[Id, inverse[E]]]], id[range[CURRY]], MIXTIMES] \� \text{Reverse}} \)

\( \text{Out [30]} = \text{APPLY[inverse[CURRY], MIXTIMES]} = \text{MIXMUL} \)

\( \text{In [31]} := \text{APPLY[inverse[CURRY], MIXTIMES]} := \text{MIXMUL} \)

Theorem.

\( \text{In [32]} := \text{ApComp[CURRY, inverse[CURRY], MIXTIMES]} \)

\( \text{Out [32]} = \text{APPLY[CURRY, MIXMUL]} = \text{MIXMUL} \)

\( \text{In [33]} := \text{APPLY[CURRY, MIXMUL]} := \text{MIXTIMES} \)

Corollary. (Mapping property of MIXMUL.)

\( \text{In [34]} := \text{SubstTest[implies, member[w, map[x, map[y, z]]],}
\text{or[empty[y], member[APPLY[inverse[CURRY], w], map[cart[x, y, z]]],}
\{w \rightarrow \text{MIXTIMES, } x \rightarrow Z, y \rightarrow \omega, z \rightarrow Z\} \� \text{Reverse}} \)

\( \text{Out [34]} = \text{member[MIXMUL, map[cart[Z, omega], Z]]} = \text{True} \)

\( \text{In [35]} := \text{member[MIXMUL, map[cart[Z, omega], Z]]} = \text{True} \)

Corollary. (Composite with eval[x].)

\( \text{In [36]} := \text{SubstTest[composite, eval[x],}
\text{APPLY[CURRY, composite[funpart[setpart[y]], id[cart[V, V]]]], y \rightarrow \text{MIXMUL} \� \text{Reverse}} \)

\( \text{Out [36]} = \text{composite[eval[x], MIXTIMES]} = \text{composite[MIXMUL, RIGHT[x]]} \)
In[37]:= composite[eval[x[_]], MIXTIMES] := composite[MIXMUL, RIGHT[x]]

range

Lemma. (Corollary of the mapping property derived in the preceding section.)

In[38]:= SubstTest[implies, member[w, map[x, y]], subclass[range[w], y], {w -> MIXMUL, x -> cart[Z, omega], y -> Z}] // Reverse

Out[38]= subclass[range[MIXMUL], Z] == True

In[39]:= % /. Equal -> SetDelayed

Lemma. (A consequence of the fact that eval[x] is a function.)

In[40]:= Map[inverse, SubstTest[composite, funpart[x], id[y], inverse[funpart[x]], {x -> eval[set[0]], y -> binhom[NATADD, INTADD]}]] // Reverse

Out[40]= composite[MIXMUL, RIGHT[set[0]]] == id[Z]

In[41]:= composite[MIXMUL, RIGHT[set[0]]] := id[Z]

Lemma. (Any image provides a lower bound for the range.)

In[42]:= Map[subclass[#1, range[MIXMUL]] &, ImageComp[MIXMUL, RIGHT[set[0]], V]]

Out[42]= subclass[Z, range[MIXMUL]] == True

In[43]:= % /. Equal -> SetDelayed

Theorem. (Equation for the range of MIXMUL.)

In[44]:= SubstTest[and, subclass[x, y], subclass[y, x], {x -> range[MIXMUL], y -> Z}]

Out[44]= equal[Z, range[MIXMUL]] == True

In[45]:= range[MIXMUL] := Z

connection with MIXDIV

The mixed divisibility relation MIXDIV was defined directly in terms of binary homomorphisms before mixed multiplication had been introduced:

In[46]:= U[binhom[NATADD, INTADD]]

Out[46]= MIXDIV

Traditionally, divisibility is defined in terms of multiplication. This connection between multiplication and divisibility is now a theorem, which can be derived quickly as follows:
\texttt{In[47]}:= \texttt{Assoc[rotate\[E\], cross[MIXTIMES, Id], inverse[SECOND]] // Reverse}

\texttt{Out[47]}= \texttt{composite[MIXMUL, inverse[SECOND]] == MIXDIV}

\texttt{In[48]}:= \texttt{composite[MIXMUL, inverse[SECOND]] := MIXDIV}