

APPLY[MIXTIMES, x] formula

Johan G. F. Belinfante
2006 December 9

```
In[1]:= SetDirectory["1:"]; << goedel88.07b; << tools.m

:Package Title: goedel88.07b          2006 December 7 at 12:50 noon

It is now: 2006 Dec 9 at 7:33

Loading Simplification Rules

TOOLS.M                               Revised 2006 December 5

weightlimit = 40
```

summary

In general, the rewrite rules for **MIXTIMES** can be patterned to a large extent on corresponding rules for **TIMES**. In this notebook a formula for **APPLY[MIXTIMES, x]** is derived, which is analogous to the following formula:

```
In[2]:= APPLY[TIMES, x]

Out[2]= union[complement[image[V, intersection[omega, set[x]]]], times[x]]
```

derivation

Lemma.

```
In[3]:= SubstTest[implies, equal[w, V], empty[funpart[w]], w → APPLY[MIXTIMES, x]] // Reverse
```

```
Out[3]= or[equal[0, funpart[APPLY[MIXTIMES, x]]], member[x, Z]] == True
```

```
In[4]:= (% /. x → x_) /. Equal → SetDelayed
```

The following corollary of a **FUNPART** theorem is needed:

```
In[5]:= Map[equal[#, APPLY[MIXTIMES, x]] &, ApComp[FUNPART, MIXTIMES, x]]
```

```
Out[5]= or[FUNCTION[APPLY[MIXTIMES, x]], not[member[x, Z]]] == True
```

```
In[6]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[7]:= SubstTest[implies, FUNCTION[t], not[equal[t, V]], t → APPLY[MIXTIMES, z]] // Reverse
```

```
Out[7]= or[member[z, Z], not[FUNCTION[APPLY[MIXTIMES, z]]]] == True
```

```
In[8]:= (% /. z → z_) /. Equal → SetDelayed
```

Combining the preceding two results yields this temporary rewrite rule:

```
In[9]:= equiv[FUNCTION[APPLY[MIXTIMES, x]], member[x, Z]]
```

```
Out[9]= True
```

```
In[10]:= FUNCTION[APPLY[MIXTIMES, x_]] := member[x, Z]
```

The following reformulation of this result is implicitly used in the derivation of the next lemma.

```
In[11]:= equal[funpart[APPLY[MIXTIMES, x]],
               intersection[image[V, intersection[Z, set[x]]], APPLY[MIXTIMES, x]]]
```

```
Out[11]= True
```

A sethood lemma:

```
In[12]:= SubstTest[implies, and[subclass[u, v], member[v, V]], member[u, V],
                  {u -> funpart[APPLY[MIXTIMES, x]],
                   v -> intersection[image[V, intersection[Z, set[x]]], APPLY[MIXTIMES, x]]}] // Reverse
```

```
Out[12]= member[funpart[APPLY[MIXTIMES, x]], V] == True
```

```
In[13]:= member[funpart[APPLY[MIXTIMES, x_]], V] := True
```

The key step of the derivation is this reification result:

```
In[14]:= Map[reify[w, #] &, ImageComp[eval[w], MIXTIMES, set[z]]] // Reverse
```

```
Out[14]= funpart[APPLY[MIXTIMES, z]] == composite[MIXMUL, LEFT[z]]
```

```
In[15]:= funpart[APPLY[MIXTIMES, z_]] := composite[MIXMUL, LEFT[z]]
```

Corollary.

```
In[16]:= SubstTest[equal, w, funpart[w], w -> APPLY[MIXTIMES, x]] // Reverse
```

```
Out[16]= equal[APPLY[MIXTIMES, x], composite[MIXMUL, LEFT[x]]] == member[x, Z]
```

```
In[17]:= equal[APPLY[MIXTIMES, x_], composite[MIXMUL, LEFT[x_]]] := member[x, Z]
```

Main theorem.

```
In[18]:= equal[APPLY[MIXTIMES, x],
               union[complement[image[V, intersection[Z, set[x]]]], funpart[APPLY[MIXTIMES, x]]]]
```

```
Out[18]= True
```

```
In[19]:= APPLY[MIXTIMES, x_] :=
  union[complement[image[V, intersection[Z, set[x]]]], composite[MIXMUL, LEFT[x]]]
```

replacement rule

An old rewrite rule is removed:

```
In[20]:= image[MIXTIMES, set[x_]] = .
```

Replacement rule:

```
In[21]:= SubstTest[image, funpart[w], set[x], w → MIXTIMES] // Reverse
```

```
Out[21]= image[MIXTIMES, set[x]] ==
  intersection[image[V, intersection[Z, set[x]]], set[composite[MIXMUL, LEFT[x]]]]
```

```
In[22]:= image[MIXTIMES, set[x_]] :=
  intersection[image[V, intersection[Z, set[x]]], set[composite[MIXMUL, LEFT[x]]]]
```

reify result

Lemma.

```
In[23]:= Assoc[IMAGE[SWAP], FUNPART, MIXTIMES]
```

```
Out[23]= composite[IMAGE[SWAP], MIXTIMES] == composite[INVERSE, MIXTIMES]
```

```
In[24]:= composite[IMAGE[SWAP], MIXTIMES] := composite[INVERSE, MIXTIMES]
```

Lemma.

```
In[25]:= composite[IMAGE[id[cart[V, V]]], VERTSECT[inverse[rotate[x]]]] // ReifNormality
```

```
Out[25]= composite[IMAGE[id[cart[V, V]]], VERTSECT[inverse[rotate[x]]]] ==
  VERTSECT[inverse[rotate[x]]]
```

```
In[26]:= composite[IMAGE[id[cart[V, V]]], VERTSECT[inverse[rotate[x_]]]] :=
  VERTSECT[inverse[rotate[x]]]
```

Lemma.

```
In[27]:= Map[composite[IMAGE[SWAP], VERTSECT[#]] &,
  SubstTest[reify, x, APPLY[t, x], t → MIXTIMES] // Reverse
```

```
Out[27]= composite[VERTSECT[inverse[rotate[composite[MIXMUL, SWAP]]]], id[Z]] ==
  composite[INVERSE, MIXTIMES]
```

```
In[28]:= % /. Equal → SetDelayed
```

Theorem.

```
In[29]:= SubstTest[composite, union[u, v], VERTSECT[inverse[rotate[composite[MIXMUL, SWAP]]]],  
             {u → id[set[0]], v → id[complement[set[0]]]} // Reverse
```

```
Out[29]= VERTSECT[inverse[rotate[composite[MIXMUL, SWAP]]]] ==  
         union[cart[complement[Z], set[0]], composite[INVERSE, MIXTIMES]]
```

```
In[30]:= VERTSECT[inverse[rotate[composite[MIXMUL, SWAP]]]] :=  
         union[cart[complement[Z], set[0]], composite[INVERSE, MIXTIMES]]
```