

# APPLY rules and PAIR[x, y]

Johan G. F. Belinfante  
2008 September 8

```
In[1]:= SetDirectory["1:"]; << goedel.08sep07a;<< tools.m

:Package Title: goedel.08sep07a          2008 September 7 at 3:05 p.m.

It is now: 2008 Sep 8 at 12:48

Loading Simplification Rules

TOOLS.M                                Revised 2008 July 5

weightlimit = 40
```

---

## summary

A formula for **PAIR[x,y]** is derived which makes it easy to derive **APPLY** rules for binary functions.

---

## introduction

For some binary functions there may only be available an **APPLY** rule involving **PAIR**. For example:

```
In[2]:= APPLY[NATEXP, PAIR[x, y]]
```

```
Out[2]= natexp[x, y]
```

```
In[5]:= APPLY[NATEXP, x]
```

```
Out[5]= APPLY[NATEXP, x]
```

All three **Normality** tests fail to provide anything useful, and **ApComp** yields the following true but useless result.

```
In[6]:= ApComp[NATEXP, id[cart[V, V]], x]
```

```
Out[6]= union[APPLY[NATEXP, x], complement[image[V, set[first[x]]]]] == APPLY[NATEXP, x]
```

In this notebook a new rewrite rule for **PAIR[x, y]** is derived that allows one to obtain a more useful result. The following result obtained using **Normality** helps motivate the new result.

```
In[9]:= PAIR[first[x], second[x]] // Normality
```

```
Out[9]= PAIR[first[x], second[x]] ==
union[complement[image[V, set[first[x]]]], pair[first[x], second[x]]]
```

Further motivation comes from this formula:

```
In[20]:= APPLY[id[cart[V, V]], x]
Out[20]= union[x, complement[image[V, set[first[x]]]]]
```

---

## a new formula for PAIR

The derivation of the new formula is an example of a situation in which the **GOEDEL** program knows a fact to be true, but lacks a corresponding rewrite rule. One simply has to ask the right question! The motivating examples in the preceding section provided a clue that led to asking the right question as it were.

Theorem.

```
In[21]:= equal[union[x, complement[image[V, set[first[x]]]], PAIR[first[x], second[x]]]
Out[21]= True

In[22]:= union[x_, complement[image[V, set[first[x_]]]]] := PAIR[first[x], second[x]]
```

With this new rewrite rule in place, the use of **ApComp** allows one to derive general **APPLY** rules for binary functions when special **APPLY** rules involving **PAIR** are available. The next section provides some examples.

---

## applications

Theorem.

```
In[23]:= ApComp[INTMUL, id[cart[V, V]], x] // Reverse
Out[23]= APPLY[INTMUL, x] == intmul[first[x], second[x]]

In[24]:= APPLY[INTMUL, x_] := intmul[first[x], second[x]]
```

Theorem.

```
In[25]:= ApComp[MIXMUL, id[cart[V, V]], x] // Reverse
Out[25]= APPLY[MIXMUL, x] == intmul[first[x], plus[second[x]]]

In[26]:= APPLY[MIXMUL, x_] := intmul[first[x], plus[second[x]]]
```

Theorem.

```
In[27]:= ApComp[NATEXP, id[cart[V, V]], x] // Reverse
Out[27]= APPLY[NATEXP, x] == natexp[first[x], second[x]]

In[28]:= APPLY[NATEXP, x_] := natexp[first[x], second[x]]
```