# pairset rules

Johan G. F. Belinfante and Tiffany D. Goble
2004 March 4

```
In[1]:= << goedel54.28c; << tools.m

        :Package Title: goedel54.28c        2004 February 28 at 10:30 p.m.

        It is now:  2004 Mar 5 at 13:22

        Loading Simplification Rules

        TOOLS.M                        Revised 2004 February 21

        weightlimit = 40
```

## summary

This notebook contains several theorems involving **pairset**.

## Quaife’s Theorem (UP4)

Lemma.

```
In[2]:= (subclass[pairset[x, y], w] // AssertTest) /. w -> pairset[x, z]

Out[2]= subclass[pairset[x, y], pairset[x, z]] == or[equal[x, y], equal[y, z], not[member[y, V]]]

In[3]:= subclass[pairset[x_, y_], pairset[x_, z_]] :=
         or[equal[x, y], equal[y, z], not[member[y, V]]]
```

Theorem **(UP4)**. The derivation is faster with **simplify** turned off.

```
In[4]:= simplify = False;

In[5]:= implies[and[member[pair[x, y], cart[u, v]], equal[pairset[x, z], pairset[y, z]]],
          equal[x, y]] // AssertTest

Out[5]= or[equal[x, y], not[equal[pairset[x, z], pairset[y, z]]],
          not[member[x, u]], not[member[y, v]]] == True

In[6]:= or[equal[x_, y_], not[equal[pairset[x_, z_], pairset[y_, z_]]],
          not[member[x_, u_]], not[member[y_, v_]]] := True
```

## Quaife’s Theorem (SC5)

The following result implies Quaife’s theorem **(SC5)**.

```
In[7]:= implies[member[pair[w, y], cart[x, z]],
          equal[union[u, v], union[intersection[u, image[V, singleton[w]]],
            intersection[v, image[V, singleton[y]]]]]] // AssertTest

Out[7]= or[equal[union[u, v], union[
            intersection[u, image[V, singleton[w]]], intersection[v, image[V, singleton[y]]]]],
          not[member[w, x]], not[member[y, z]]] == True

In[8]:= or[equal[union[u_, v_], union[intersection[u_, image[V, singleton[w_]]],
            intersection[v_, image[V, singleton[y_]]]]],
          not[member[w_, x_]], not[member[y_, z_]]] := True
```

## Theorems A−UP−1 and A−UP−2

Theorem **A−UP−1** resembles Quaife’s Theorem **(SC5)**, but with intersection replacing union.

```
In[9]:= implies[member[pair[x, y], cart[u, v]],
          equal[A[pairset[x, y]], intersection[x, y]]] // AssertTest

Out[9]= or[equal[A[pairset[x, y]], intersection[x, y]],
          not[member[x, u]], not[member[y, v]]] == True

In[10]:= or[equal[A[pairset[x_, y_]], intersection[x_, y_]],
           not[member[x_, u_]], not[member[y_, v_]]] := True
```

Theorem **A−UP−2**:

```
In[11]:= implies[and[member[x, z], subclass[x, y]], equal[A[pairset[x, y]], x]] // AssertTest

Out[11]= or[equal[x, A[pairset[x, y]]], not[member[x, z]], not[subclass[x, y]]] == True

In[12]:= or[equal[x_, A[pairset[x_, y_]]], not[member[x_, z_]], not[subclass[x_, y_]]] := True
```

## Theorems from the second FU group

Lemma.

```
In[13]:= FUNCTION[singleton[x]] // AssertTest

Out[13]= FUNCTION[singleton[x]] == or[member[first[x], V], not[member[x, V]]]

In[14]:= FUNCTION[singleton[x_]] := or[member[first[x], V], not[member[x, V]]]
```

Theorem **FU−UP−1**:

```
In[15]:= Map[implies[member[x, z], #] &, SubstTest[implies,
           FUNCTION[union[u, v]], FUNCTION[u], {u -> singleton[x], v -> singleton[y]}]]

Out[15]= or[member[first[x], V], not[FUNCTION[pairset[x, y]]], not[member[x, z]]] == True

In[16]:= or[member[first[x_], V], not[FUNCTION[pairset[x_, y_]]], not[member[x_, z_]]] := True
```

Theorem **FU−UP−3**:

```
In[17]:= implies[FUNCTION[pairset[pair[x, y], pair[x, z]]], equal[y, z]] // AssertTest

Out[17]= or[equal[y, z], not[FUNCTION[pairset[pair[x, y], pair[x, z]]]]] == True

In[18]:= or[equal[y_, z_], not[FUNCTION[pairset[pair[x_, y_], pair[x_, z_]]]]] := True
```

Theorem **FU−TP−1** follows from this:

```
In[19]:= subclass[pairset[pair[w, x], pair[y, z]], cart[u, v]] // AssertTest

Out[19]= subclass[pairset[pair[w, x], pair[y, z]], cart[u, v]] ==
          and[member[w, u], member[x, v], member[y, u], member[z, v]]

In[20]:= subclass[pairset[pair[w_, x_], pair[y_, z_]], cart[u_, v_]] :=
          and[member[w, u], member[x, v], member[y, u], member[z, v]]
```

Theorem **FU−TP−3** is a consequence of this:

```
In[21]:= FUNCTION[pairset[pair[x, y], pair[y, x]]] // AssertTest

Out[21]= FUNCTION[pairset[pair[x, y], pair[y, x]]] == and[member[x, V], member[y, V]]

In[22]:= FUNCTION[pairset[pair[x_, y_], pair[y_, x_]]] := and[member[x, V], member[y, V]]
```

## new results

The following results were discovered while studying well orderings.

```
In[23]:= Map[implies[and[member[pair[u, v], x], REFLEXIVE[x]], #] &,
           subclass[pairset[u, v], fix[x]] // AssertTest] // MapNotNot

Out[23]= or[not[member[pair[u, v], x]],
           not[subclass[x, cart[fix[x], fix[x]]]], subclass[pairset[u, v], fix[x]]] == True

In[24]:= or[not[member[pair[u_, v_], x_]], not[subclass[x_, cart[fix[x_], fix[x_]]]],
           subclass[pairset[u_, v_], fix[x_]]] := True
```

A disjointness result.

```
In[25]:= disjoint[x, pairset[y, z]] // AssertTest

Out[25]= equal[0, intersection[x, pairset[y, z]]] == and[not[member[y, x]], not[member[z, x]]]

In[26]:= equal[0, intersection[x_, pairset[y_, z_]]] := and[not[member[y, x]], not[member[z, x]]]
```

Lemma related to transitivity.

```
In[27]:= implies[and[subclass[pairset[u, v], x], subclass[pairset[v, w], x]],
           subclass[pairset[u, w], x]] // AssertTest

Out[27]= or[not[subclass[pairset[u, v], x]],
           not[subclass[pairset[v, w], x]], subclass[pairset[u, w], x]] == True

In[28]:= or[not[subclass[pairset[u_, v_], x_]],
           not[subclass[pairset[v_, w_], x_]], subclass[pairset[u_, w_], x_]] := True
```

The following somewhat special result is useful.

```
In[29]:= subclass[pairset[x, y], image[z, singleton[y]]] // AssertTest

Out[29]= subclass[pairset[x, y], image[z, singleton[y]]] ==
          or[and[member[y, fix[z]], member[pair[y, x], z]],
           and[member[y, fix[z]], not[member[x, V]]], and[not[member[x, V]], not[member[y, V]]]]
```