the class of partitions

Johan G. F. Belinfante
2010 January 5

```
In[1]:= SetDirectory["l:"]; << goedel.10jan04a; << tools.m

        :Package Title: goedel.10jan04a              2010 January 4 at 7:30 p.m.

        It is now:  2010 Jan 5 at 17:39

        Loading Simplification Rules

        TOOLS.M                                      Revised 2009 December 17

        weightlimit = 40
```

summary

A partition **x** of a class **y** is a pairwise disjoint collection of nonempty subsets of **y** whose sum class is **y**. Since **y = U[x]**, one can dispense with the variable **y** altogether, and work with just the single variable **x**. The class **PARTNS** of all (small) partitions will accordingly be defined as the class of all pairwise disjoint sets that do not hold the empty set.

```
In[3]:= member[x_, PARTNS] :=
          and[member[x, V], not[member[0, x]], subclass[cart[x, x], union[DISJOINT, Id]]]
```

In this notebook, some basic properties of this class are derived.

normalization

The class **PARTNS** is normalized by the following rewrite rule.

Theorem. Normalization rule.

```
In[4]:= PARTNS // Normality // Reverse

Out[4]= intersection[cliques[union[DISJOINT, Id]], P[complement[set[0]]]] == PARTNS

In[5]:= intersection[cliques[union[DISJOINT, Id]], P[complement[set[0]]]] := PARTNS
```

Corollary. Partitions are pairwise disjoint sets.

```
In[6]:= SubstTest[subclass, intersection[u, v], u,
          {u -> cliques[union[DISJOINT, Id]], v -> P[complement[set[0]]]}] // Reverse

Out[6]= subclass[PARTNS, cliques[union[DISJOINT, Id]]] == True
```

```
In[7]:= subclass[PARTNS, cliques[union[DISJOINT, Id]]] := True
```

CORE[PARTNS]

Theorem. The sum class of **PARTNS** is the class of non-empty sets.

```
In[8]:= SubstTest[U, intersection[u, P[v]],
          {u -> cliques[union[DISJOINT, Id]], v -> complement[set[0]]}] // Reverse

Out[8]= U[PARTNS] == complement[set[0]]

In[9]:= U[PARTNS] := complement[set[0]]
```

A similar result holds for **core**.

Lemma.

```
In[10]:= AssInt[cliques[union[DISJOINT, Id]], P[complement[set[0]]], P[x]]

Out[10]= intersection[cliques[union[DISJOINT, Id]], P[intersection[x, complement[set[0]]]]] ==
          intersection[PARTNS, P[x]]

In[11]:= intersection[cliques[union[DISJOINT, Id]], P[intersection[x_, complement[set[0]]]]] :=
          intersection[PARTNS, P[x]]
```

Theorem.

```
In[12]:= SubstTest[U, intersection[u, P[v]], {u -> cliques[union[DISJOINT, Id]],
            v → intersection[x, complement[set[0]]]}] // Reverse

Out[12]= core[PARTNS, x] == intersection[x, complement[set[0]]]

In[13]:= core[PARTNS, x_] := intersection[x, complement[set[0]]]
```

Corollary.

```
In[15]:= Map[VERTSECT, SubstTest[reify, x, core[t, x], t → PARTNS]]

Out[15]= CORE[PARTNS] == IMAGE[id[complement[set[0]]]]

In[16]:= CORE[PARTNS] := IMAGE[id[complement[set[0]]]]
```

other properties of the class of partitions

Although the class **PARTNS** only holds partitions which are sets, it is also useful for partitions that are proper classes.

Theorem. A class **x** is a (possibly large) partition if and only if **P[x] ⊂ PARTNS**.

```
In[17]:= SubstTest[subclass, t, intersection[u, v],
           {t → P[x], u -> cliques[union[DISJOINT, Id]], v -> P[complement[set[0]]]}] // Reverse

Out[17]= subclass[P[x], PARTNS] ==
          and[not[member[0, x]], subclass[cart[x, x], union[DISJOINT, Id]]]

In[18]:= subclass[P[x_], PARTNS] :=
          and[not[member[0, x]], subclass[cart[x, x], union[DISJOINT, Id]]]
```

Theorem. The class **PARTNS** is a class of the form **cliques[x]** for the restriction of **DISJOINT** $\bigcup$ **Id** to the class of all nonempty sets.

```
In[19]:= SubstTest[cliques, intersection[u, v],
           {u → union[DISJOINT, Id], v -> cartsq[complement[set[0]]]}] // Reverse

Out[19]= cliques[union[composite[id[complement[set[0]]], DISJOINT, id[complement[set[0]]]],
            id[complement[set[0]]]]] == PARTNS

In[20]:= cliques[union[composite[id[complement[set[0]]], DISJOINT, id[complement[set[0]]]],
            id[complement[set[0]]]]] := PARTNS
```

Corollary. The class of partitions is hereditary.

```
In[21]:= SubstTest[image, inverse[S], cliques[x],
           x -> union[composite[id[complement[set[0]]], DISJOINT, id[complement[set[0]]]],
             id[complement[set[0]]]]] // Reverse

Out[21]= image[inverse[S], PARTNS] == PARTNS

In[22]:= image[inverse[S], PARTNS] := PARTNS
```

Corollary. The class of partitions is closed under arbitrary intersections. No new rewrite rule is needed for this.

```
In[24]:= Aclosure[PARTNS]

Out[24]= PARTNS
```

Corollary. The union of a chain of partitions is a partition.

```
In[25]:= SubstTest[Uchains, cliques[x],
           x -> union[composite[id[complement[set[0]]], DISJOINT, id[complement[set[0]]]],
             id[complement[set[0]]]]] // Reverse

Out[25]= Uchains[PARTNS] == PARTNS

In[26]:= Uchains[PARTNS] := PARTNS
```

The class of partitions is not closed under arbitrary unions. In fact, any set of nonempty sets is the sum class of a set of partitions. No new rewrite rule is needed for this.

```
In[28]:= Uclosure[PARTNS]

Out[28]= P[complement[set[0]]]
```

replacement rules

Some old rules can be replaced with the rules derived in this section.

Theorem.

```
In[32]:= Map[equal[V, #] &,
          dif[PARTNS, fix[composite[DISJOINT, IMAGE[inverse[PS]]]]]] // complement // Normality]

Out[32]= subclass[PARTNS, fix[composite[DISJOINT, IMAGE[inverse[PS]]]]] == True

In[33]:= subclass[PARTNS, fix[composite[DISJOINT, IMAGE[inverse[PS]]]]] := True
```

Lemma. Every set can be partitioned.

```
In[47]:= (SubstTest[implies, member[u, v],
            not[empty[v]], {u → image[SINGLETON, t], v → intersection[PARTNS,
               image[inverse[BIGCUP], set[t]]]}] // Reverse) /. t → setpart[x]

Out[47]= member[setpart[x], image[BIGCUP, PARTNS]] == True

In[48]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem.

```
In[49]:= SubstTest[class, x, member[setpart[x], t], t -> image[BIGCUP, PARTNS]]

Out[49]= image[BIGCUP, PARTNS] == V

In[50]:= image[BIGCUP, PARTNS] := V
```