PO and EQV

Johan G. F. Belinfante and Lee Martie
2006 June 18

```
In[1]:= SetDirectory["l:"]; << goedel82.16a; << tools.m

        :Package Title: goedel82.16a         2006 June 16 at 10:20 p.m.

        It is now:  2006 Jun 18 at 11:27

        Loading Simplification Rules

        TOOLS.M                      Revised 2006 June 6

        weightlimit = 40
```

summary

Some variable-free rewrite rules for equivalences and partial orderings are derived. Most of the rules for **PO** were initially discovered by using **reify** to remove wrapped variables. Quite a few simplification rules were needed to accomplish this, presumably due to the complexity of the definition of the **po** wrapper. Once the formulas were known, however, it was found to be relatively easy to obtain direct derivations of these results without resorting to reification. The **EQV** rules are derived in a completely analogous fashion.

RFX rules

Lemma.

```
In[2]:= symdif[composite[IMAGE[FIRST], id[RFX]],
          composite[IMAGE[inverse[DUP]], id[RFX]]] // RelnRenormality

Out[2]= union[composite[intersection[complement[IMAGE[FIRST]], IMAGE[inverse[DUP]]], id[RFX]],
          composite[intersection[complement[IMAGE[inverse[DUP]]], IMAGE[FIRST]], id[RFX]]] == 0

In[3]:= % /. Equal → SetDelayed
```

Theorem.

```
In[4]:= SubstTest[equal, 0, symdif[u, v],
         {u -> composite[IMAGE[FIRST], id[RFX]], v -> composite[IMAGE[inverse[DUP]], id[RFX]]}]

Out[4]= True == equal[composite[IMAGE[FIRST], id[RFX]], composite[IMAGE[inverse[DUP]], id[RFX]]]

In[5]:= composite[IMAGE[FIRST], id[RFX]] := composite[IMAGE[inverse[DUP]], id[RFX]]
```

Corollary.

```
In[6]:= Map[composite[#, id[RFX]] &, Assoc[IMAGE[FIRST], id[RFX], IMAGE[SWAP]]]

Out[6]= composite[IMAGE[SECOND], id[RFX]] == composite[IMAGE[inverse[DUP]], id[RFX]]

In[7]:= composite[IMAGE[SECOND], id[RFX]] := composite[IMAGE[inverse[DUP]], id[RFX]]
```

The rules in this section state that for a reflexive relation, the domain, range and fixed point sets are all equal.

ANTISYM rule

Lemma.

```
In[8]:= symdif[composite[CORE[SYM], id[ANTISYM]],
           composite[IMAGE[id[Id]], id[ANTISYM]]] // domain // Normality

Out[8]= union[intersection[ANTISYM,
           complement[fix[composite[inverse[CORE[SYM]], IMAGE[id[Id]]]]]], intersection[
           ANTISYM, complement[fix[composite[inverse[IMAGE[id[Id]]], CORE[SYM]]]]]] == 0

In[9]:= % /. Equal → SetDelayed
```

Theorem.

```
In[10]:= SubstTest[equal, 0, domain[symdif[u, v]],
           {u -> composite[CORE[SYM], id[ANTISYM]], v -> composite[IMAGE[id[Id]], id[ANTISYM]]}]

Out[10]= True == equal[composite[CORE[SYM], id[ANTISYM]], composite[IMAGE[id[Id]], id[ANTISYM]]]

In[11]:= composite[CORE[SYM], id[ANTISYM]] := composite[IMAGE[id[Id]], id[ANTISYM]]
```

The function **CORE[SYM]** yields the largest symmetric relation contained in a given relation. This is obtained by simply intersecting a relation with its inverse.

IDEM rule

A relation **x** is idempotent if **composite[x, x]** is equal to **x**.

```
In[12]:= SubstTest[composite, funpart[x], id[fix[funpart[x]]], x → composite[COMPOSE, DUP]]

Out[12]= composite[COMPOSE, DUP, id[IDEM]] == id[IDEM]

In[13]:= composite[COMPOSE, DUP, id[IDEM]] := id[IDEM]
```

EQV rules

Equivalence relations are reflexive, symmetric and transitive relations. The rules for **EQV** are immediate corollaries of the rules derived above.

```
In[14]:= Assoc[IMAGE[id[cart[V, V]]], id[P[cart[V, V]]], id[EQV]]

Out[14]= composite[IMAGE[id[cart[V, V]]], id[EQV]] == id[EQV]

In[15]:= composite[IMAGE[id[cart[V, V]]], id[EQV]] := id[EQV]
```

From the reflexive property one finds:

```
In[16]:= Assoc[IMAGE[FIRST], id[RFX], id[EQV]]

Out[16]= composite[IMAGE[FIRST], id[EQV]] == composite[IMAGE[inverse[DUP]], id[EQV]]

In[17]:= composite[IMAGE[FIRST], id[EQV]] := composite[IMAGE[inverse[DUP]], id[EQV]]

In[18]:= Assoc[IMAGE[SECOND], id[RFX], id[EQV]]

Out[18]= composite[IMAGE[SECOND], id[EQV]] == composite[IMAGE[inverse[DUP]], id[EQV]]

In[19]:= composite[IMAGE[SECOND], id[EQV]] := composite[IMAGE[inverse[DUP]], id[EQV]]
```

From the symmetric property one finds:

```
In[20]:= Assoc[CORE[SYM], id[SYM], id[EQV]]

Out[20]= composite[CORE[SYM], id[EQV]] == id[EQV]

In[21]:= composite[CORE[SYM], id[EQV]] := id[EQV]
```

The reflexive and transitive properties imply idempotence, from which one finds:

```
In[22]:= Assoc[composite[COMPOSE, DUP], id[IDEM], id[EQV]]

Out[22]= composite[COMPOSE, DUP, id[EQV]] == id[EQV]

In[23]:= composite[COMPOSE, DUP, id[EQV]] := id[EQV]
```

EQUIVALENCE rule

```
In[24]:= equiv[and[EQUIVALENCE[x], REFLEXIVE[x]], EQUIVALENCE[x]]

Out[24]= True

In[25]:= and[EQUIVALENCE[x_], REFLEXIVE[x_]] := EQUIVALENCE[x]
```

PO rules

Partial orderings are reflexive, antisymmetric and transitive relations. The rules for **PO** are also immediate corollaries of rules derived above.

```
In[26]:= Assoc[IMAGE[id[cart[V, V]]], id[P[cart[V, V]]], id[PO]]

Out[26]= composite[IMAGE[id[cart[V, V]]], id[PO]] == id[PO]

In[27]:= composite[IMAGE[id[cart[V, V]]], id[PO]] := id[PO]
```

From the reflexive property:

```
In[28]:= Assoc[IMAGE[FIRST], id[RFX], id[PO]]

Out[28]= composite[IMAGE[FIRST], id[PO]] == composite[IMAGE[inverse[DUP]], id[PO]]

In[29]:= composite[IMAGE[FIRST], id[PO]] := composite[IMAGE[inverse[DUP]], id[PO]]

In[30]:= Assoc[IMAGE[SECOND], id[RFX], id[PO]]

Out[30]= composite[IMAGE[SECOND], id[PO]] == composite[IMAGE[inverse[DUP]], id[PO]]

In[31]:= composite[IMAGE[SECOND], id[PO]] := composite[IMAGE[inverse[DUP]], id[PO]]
```

The antisymmetric property:

```
In[32]:= Assoc[CORE[SYM], id[ANTISYM], id[PO]]

Out[32]= composite[CORE[SYM], id[PO]] == composite[IMAGE[id[Id]], id[PO]]

In[33]:= composite[CORE[SYM], id[PO]] := composite[IMAGE[id[Id]], id[PO]]
```

The idempotence property:

```
In[34]:= Assoc[composite[COMPOSE, DUP], id[IDEM], id[PO]]

Out[34]= composite[COMPOSE, DUP, id[PO]] == id[PO]

In[35]:= composite[COMPOSE, DUP, id[PO]] := id[PO]
```