COMMUTATIVE

Johan G. F. Belinfante and Ming Li
2006 September 19

```
In[1]:= SetDirectory["l:"]; << goedel85.19a; << tools.m

:Package Title: goedel85.19a          2006 September 19 at 5:15 p.m.

It is now:  2006 Sep 19 at 18:12

Loading Simplification Rules

TOOLS.M                       Revised 2006 August 22

weightlimit = 40
```

summary

The class **COMMUTATIVE** of commutative relations is defined and some properties of this class are derived.

definition

The class **COMMUTATIVE** is defined by the following membership rule.

```
In[2]:= member[x_, COMMUTATIVE] := and[equal[x, composite[x, SWAP]], member[x, V]]
```

An explicit formula is readily derived:

```
In[3]:= Map[equal[0, #] &, symdif[COMMUTATIVE, fix[IMAGE[cross[SWAP, Id]]]] // Normality]

Out[3]= equal[COMMUTATIVE, fix[IMAGE[cross[SWAP, Id]]]] == True

In[4]:= fix[IMAGE[cross[SWAP, Id]]] := COMMUTATIVE
```

another formula for COMMUTATIVE

Lemma.

```
In[5]:= SubstTest[implies, equal[u, v], equal[image[w, u], image[w, v]],
         {u -> fix[IMAGE[cross[Id, x]]], v -> subvar[cross[Id, x]], w → IMAGE[SWAP]}]

Out[5]= or[equal[fix[IMAGE[cross[x, Id]]], subvar[cross[x, Id]]],
          not[equal[fix[IMAGE[cross[Id, x]]], subvar[cross[Id, x]]]]] == True
```

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

Theorem.

```
In[7]:= Map[not, SubstTest[and, implies[p1, p2],
          not[implies[p1, p3]], {p1 -> subclass[composite[x, x], Id],
           p2 -> equal[fix[IMAGE[cross[Id, x]]], subvar[cross[Id, x]]],
           p3 -> equal[fix[IMAGE[cross[x, Id]]], subvar[cross[x, Id]]]}]]

Out[7]= or[equal[fix[IMAGE[cross[x, Id]]], subvar[cross[x, Id]]],
          not[subclass[composite[x, x], Id]]] == True

In[8]:= or[equal[fix[IMAGE[cross[x_, Id]]], subvar[cross[x_, Id]]],
          not[subclass[composite[x_, x_], Id]]] := True
```

Corollary. A relation is commutative if it is flip-invariant.

```
In[9]:= SubstTest[implies, subclass[composite[x, x], Id],
         equal[fix[IMAGE[cross[x, Id]]], subvar[cross[x, Id]]], x → SWAP]

Out[9]= equal[COMMUTATIVE, intersection[invar[cross[SWAP, Id]], P[cart[cart[V, V], V]]]] == True

In[10]:= intersection[invar[cross[SWAP, Id]], P[cart[cart[V, V], V]]] := COMMUTATIVE
```

Yet another useful formula for **COMMUTATIVE** automatically follows from this result, and does not require a separate rewrite rule:

```
In[11]:= subvar[cross[SWAP, Id]]

Out[11]= COMMUTATIVE
```

Uclosure and Aclosure

The following result is an application of the formula derived in the preceding section.

```
In[12]:= SubstTest[Uclosure, subvar[x], x -> cross[SWAP, Id]]

Out[12]= Uclosure[COMMUTATIVE] == COMMUTATIVE

In[13]:= Uclosure[COMMUTATIVE] := COMMUTATIVE
```

A similar result holds for **Aclosure**, but the derivation uses a different formula.

```
In[14]:= SubstTest[Aclosure, intersection[invar[x], P[y]],
          {x -> cross[SWAP, Id], y -> cart[cart[V, V], V]}]

Out[14]= Aclosure[COMMUTATIVE] == COMMUTATIVE

In[15]:= Aclosure[COMMUTATIVE] := COMMUTATIVE
```

This is similar:

```
In[16]:= SubstTest[fix[HULL[#]] &, intersection[invar[x], P[y]],
          {x -> cross[SWAP, Id], y -> cart[cart[V, V], V]}]

Out[16]= fix[HULL[COMMUTATIVE]] == COMMUTATIVE

In[17]:= fix[HULL[COMMUTATIVE]] := COMMUTATIVE
```

U[COMMUTATIVE]

Lemma.

```
In[18]:= SubstTest[subclass, fix[x], range[x], x -> IMAGE[cross[SWAP, Id]]]

Out[18]= subclass[U[COMMUTATIVE], cart[cart[V, V], V]] == True

In[19]:= % /. Equal → SetDelayed
```

In the oppososite direction, one has:

```
In[20]:= SubstTest[subclass, cart[U[subvar[x]], U[subvar[y]]],
          U[subvar[cross[x, y]]], {x → SWAP, y → Id}]

Out[20]= subclass[cart[cart[V, V], V], U[COMMUTATIVE]] == True

In[21]:= % /. Equal → SetDelayed
```

Combining these results yields an equation:

```
In[22]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> U[COMMUTATIVE], v -> cart[cart[V, V], V]}]

Out[22]= True == equal[cart[cart[V, V], V], U[COMMUTATIVE]]

In[23]:= U[COMMUTATIVE] := cart[cart[V, V], V]
```

Corollary.

```
In[24]:= equal[intersection[COMMUTATIVE, P[cart[cart[V, V], V]]], COMMUTATIVE]

Out[24]= True

In[25]:= intersection[COMMUTATIVE, P[cart[cart[V, V], V]]] := COMMUTATIVE
```

IMAGE[FIRST] and IMAGE[SECOND] formulas

Theorem. Any symmetric relation can be the domain of a commutative relation.

```
In[26]:= SubstTest[image, IMAGE[FIRST], subvar[cross[x, Id]], x → SWAP]

Out[26]= image[IMAGE[FIRST], COMMUTATIVE] == SYM
```

```
In[27]:= image[IMAGE[FIRST], COMMUTATIVE] := SYM
```

Lemma.

```
In[28]:= ImageComp[IMAGE[SECOND], CART, cart[SYM, V]] // Reverse

Out[28]= image[IMAGE[SECOND], image[CART, cart[SYM, V]]] == V

In[29]:= image[IMAGE[SECOND], image[CART, cart[SYM, V]]] := V
```

Lemma.

```
In[30]:= Map[equal[0, #] &,
          dif[cart[SYM, V], image[inverse[CART], COMMUTATIVE]] // RelnNormality]

Out[30]= subclass[image[CART, cart[SYM, V]], COMMUTATIVE] == True

In[31]:= subclass[image[CART, cart[SYM, V]], COMMUTATIVE] := True
```

Lemma.

```
In[32]:= SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
          {u -> image[CART, cart[SYM, V]], v -> COMMUTATIVE, w → IMAGE[SECOND]}]

Out[32]= subclass[V, image[IMAGE[SECOND], COMMUTATIVE]] == True

In[33]:= % /. Equal → SetDelayed
```

Theorem. Any set can be the range of a commutative relation.

```
In[34]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> image[IMAGE[SECOND], COMMUTATIVE], v → V}]

Out[34]= True == equal[V, image[IMAGE[SECOND], COMMUTATIVE]]

In[35]:= image[IMAGE[SECOND], COMMUTATIVE] := V
```