commuting equivalences

Johan G. F. Belinfante
2008 November 7

```
In[1]:= SetDirectory["l:"]; << goedel.08nov06b;<< tools.m

        :Package Title: goedel.08nov06b          2008 November 6 at 4:10 p.m.

        It is now:  2008 Nov 7 at 21:19

        Loading Simplification Rules

        TOOLS.M                                    Revised 2008 October 21

        weightlimit = 40
```

summary

The composite of commuting equivalence relations is an equivalence relation.

derivation

Lemma. A statement using **eqv** wrappers.

```
In[2]:= Map[implies[commute[eqv[x], eqv[y]], #] &,
         SubstTest[equal, composite[t, inverse[t]], t, t → composite[eqv[x], eqv[y]]]]

Out[2]= or[EQUIVALENCE[composite[eqv[x], eqv[y]]],
          not[equal[composite[eqv[x], eqv[y]], composite[eqv[y], eqv[x]]]]] == True

In[3]:= or[EQUIVALENCE[composite[eqv[x_], eqv[y_]]],
          not[equal[composite[eqv[x_], eqv[y_]], composite[eqv[y_], eqv[x_]]]]] := True
```

Theorem. A wrapper-free restatement.

```
In[4]:= SubstTest[implies, and[equal[x, eqv[u]], equal[y, eqv[v]], commute[x, y]],
          EQUIVALENCE[composite[x, y]], {u → x, v → y}] // Reverse

Out[4]= or[EQUIVALENCE[composite[x, y]], not[equal[composite[x, y], composite[y, x]]],
          not[EQUIVALENCE[x]], not[EQUIVALENCE[y]]] == True

In[5]:= or[EQUIVALENCE[composite[x_, y_]], not[equal[composite[x_, y_], composite[y_, x_]]],
          not[EQUIVALENCE[x_]], not[EQUIVALENCE[y_]]] := True
```

Corollary. A variable-free restatement.

```
In[11]:= Map[empty[domain[complement[#]]] &,
          SubstTest[class, pair[x, y], implies[member[pair[x, y], u], member[pair[x, y], v]],
           {u -> restrict[COMMUTE, EQV, EQV], v -> image[inverse[COMPOSE], EQV]}]]

Out[11]= subclass[image[COMPOSE, composite[id[EQV], COMMUTE, id[EQV]]], EQV] == True

In[12]:= % /. Equal→ SetDelayed
```

Lemma. Reverse inclusion.

```
In[13]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
           {t → COMPOSE, u→ id[EQV], v -> composite[id[EQV], COMMUTE, id[EQV]]}] // Reverse

Out[13]= subclass[EQV, image[COMPOSE, composite[id[EQV], COMMUTE, id[EQV]]]] == True

In[14]:= % /. Equal→ SetDelayed
```

Theorem. Restatement as an equation.

```
In[15]:= SubstTest[and, subclass[u, v], subclass[v, u],
           {u -> image[COMPOSE, composite[id[EQV], COMMUTE, id[EQV]]], v → EQV}]

Out[15]= equal[EQV, image[COMPOSE, composite[id[EQV], COMMUTE, id[EQV]]]] == True

In[17]:= image[COMPOSE, composite[id[EQV], COMMUTE, id[EQV]]] := EQV
```