reflexive and transitive: counterexamples

Johan G. F. Belinfante
2009 January 18

```
In[1]:= SetDirectory["l:"]; << goedel.09jan16a;<< tools.m

        :Package Title: goedel.09jan16a        2009 January 16 at 1:10 p.m.

        It is now:  2009 Jan 18 at 14:50

        Loading Simplification Rules

        TOOLS.M                                     Revised 2009 January 15

        weightlimit = 40
```

summary

A reflexive relation need not be transitive and vice versa. Minimal-cardinality counterexamples needed to prove these facts are constructed in this notebook. Additional examples that occur naturally in certain applications, but which are proper classes, are also considered.

minimal examples

Theorem. The digraph **0 → 1** with two vertices and one edge provides a minimal example of a transitive relation that is not reflexive.

```
In[2]:= Map[not, SubstTest[implies, and[member[u, v], subclass[v, w]],
           member[u, w], {u → cart[set[0], set[set[0]]], v → TRV, w → RFX}]] // Reverse

Out[2]= subclass[TRV, RFX] == False

In[3]:= subclass[TRV, RFX] := False
```

A minimal example of a reflexive relation that is not transitive is provided by the reflexive closure of the digraph **0 → 1 → 2**. Adding three loops at each of the vertices produces a graph with 3 vertices and 5 edges. It is easy to establish the needed facts about the corresponding relation.

Lemma.

```
In[4]:= SubstTest[subclass, t, cartsq[fix[t]], t -> union[cart[set[0], set[set[0]]],
           cart[set[set[0]], set[succ[set[0]]]], id[succ[succ[set[0]]]]]]

Out[4]= REFLEXIVE[union[cart[set[0], set[set[0]]],
           cart[set[set[0]], set[succ[set[0]]]], id[succ[succ[set[0]]]]]] == True
```

```
In[5]:= REFLEXIVE[union[cart[set[0], set[set[0]]],
           cart[set[set[0]], set[succ[set[0]]]], id[succ[succ[set[0]]]]]] := True
```

Lemma.

```
In[6]:= SubstTest[subclass, composite[t, t], t, t -> union[cart[set[0], set[set[0]]],
           cart[set[set[0]], set[succ[set[0]]]], id[succ[succ[set[0]]]]]]

Out[6]= TRANSITIVE[union[cart[set[0], set[set[0]]],
           cart[set[set[0]], set[succ[set[0]]]], id[succ[succ[set[0]]]]]] == False

In[7]:= TRANSITIVE[union[cart[set[0], set[set[0]]],
           cart[set[set[0]], set[succ[set[0]]]], id[succ[succ[set[0]]]]]] := False
```

Corollary.

```
In[8]:= Map[not, SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u -> union[cart[set[0], set[set[0]]], cart[set[set[0]], set[succ[set[0]]]],
              id[succ[succ[set[0]]]]], v → RFX, w → TRV}]] // Reverse

Out[8]= subclass[RFX, TRV] == False

In[9]:= subclass[RFX, TRV] := False
```

comparability

Two sets **x** and **y** are **comparable** if one of them is a subset of the other.

Lemma. Sets need not be comparable.

```
In[10]:= subclass[cart[V, V], union[S, inverse[S]]] // AssertTest

Out[10]= subclass[cart[V, V], union[S, inverse[S]]] == False

In[11]:= subclass[cart[V, V], union[S, inverse[S]]] := False
```

Theorem. The comparability relation is not transitive.

```
In[12]:= SubstTest[subclass, composite[t, t], t, t -> union[S, inverse[S]]]

Out[12]= TRANSITIVE[union[S, inverse[S]]] == False

In[13]:= TRANSITIVE[union[S, inverse[S]]] := False
```

Theorem. The comparability relation is reflexive.

```
In[14]:= SubstTest[subclass, t, cartsq[fix[t]], t -> union[S, inverse[S]]]

Out[14]= REFLEXIVE[union[S, inverse[S]]] == True

In[15]:= REFLEXIVE[union[S, inverse[S]]] := True
```

pairwise disjointness

A collection of sets is **pairwise disjoint** if for any elements **x** and **y**, either **x** and **y** are equal or they are disjoint. The vertical sections of an equivalence relation have this property:

```
In[16]:= image[IMAGE[SECOND], image[VS, EQV]]

Out[16]= intersection[cliques[union[DISJOINT, Id]], P[complement[set[0]]]]
```

The pairwise-disjointness relation **union[DISJOINT,Id]** is obviously reflexive:

```
In[17]:= REFLEXIVE[union[DISJOINT, Id]]

Out[17]= True
```

Lemma. In general, sets need not be pairwise disjoint.

```
In[18]:= equal[cart[V, V], union[DISJOINT, Id]] // AssertTest

Out[18]= equal[cart[V, V], union[DISJOINT, Id]] == False

In[19]:= equal[cart[V, V], union[DISJOINT, Id]] := False
```

Theorem. The pairwise-disjointness relation is not transitive.

```
In[20]:= SubstTest[equal, t, trv[t], t → union[DISJOINT, Id]]

Out[20]= TRANSITIVE[union[DISJOINT, Id]] == False

In[21]:= TRANSITIVE[union[DISJOINT, Id]] := False
```

commutativity

Relations **x** and **y** are said to commute if **composite[x, y] = composite[y, x]**. Since any relation commutes with itself, commutativity is a reflexive relation:

```
In[22]:= REFLEXIVE[COMMUTE]

Out[22]= True
```

Lemma. Commutativity for cartesian squares is equivalent to pairwise disjointness.

```
In[23]:= composite[inverse[DUP], inverse[CART], COMMUTE, CART, DUP] // RelnNormality

Out[23]= composite[inverse[DUP], inverse[CART], COMMUTE, CART, DUP] == union[DISJOINT, Id]

In[24]:= composite[inverse[DUP], inverse[CART], COMMUTE, CART, DUP] := union[DISJOINT, Id]
```

Theorem. Commutativity is not a transitive relation.

```
In[25]:= Map[not, SubstTest[implies, TRANSITIVE[x],
            TRANSITIVE[composite[inverse[funpart[y]], x, funpart[y]]],
            {x → COMMUTE, y → composite[CART, DUP]}]] // Reverse

Out[25]= TRANSITIVE[COMMUTE] == False

In[26]:= TRANSITIVE[COMMUTE] := False
```