unions of chains of squares

Johan G. F. Belinfante
2011 November 30

In[1]:= **SetDirectory["l:"]; << goedel.11nov29a**

```
:Package Title: goedel.11nov29a              2011 November 29 at 10:20 a.m.

Loading takes about thirteen minutes, half that time due to builtin pauses.

It is now:  2011 Nov 30 at 12:22

Loading Simplification Rules

TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

weightlimit = 40

Loading completed.

It is now:  2011 Nov 30 at 12:34
```

summary

The union of a chain of squares is a square. The following concise variable-free statement of this is already available.

In[2]:= **Uchains[image[CART, Id]]**

Out[2]= image[CART, Id]

In this notebook a more general result is derived, dropping the requirement that the chain be a set. Some related results are also derived. Several derivations can be speeded up by a few seconds if one omits certain proof steps. The omitted steps are indicated by **(* ⋯ *)**.

derivation

It is convenient to introduce the following temporary wrapper for collections of squares.

In[3]:= **sq[x_] := image[CART, id[image[IMAGE[inverse[DUP]], x]]]**

The introduction rule for this wrapper is already available.

In[4]:= **subclass[sq[x], image[CART, Id]]**

Out[4]= True

So is the elimination rule.

```
In[5]:=  equal[x, sq[x]]

Out[5]=  subclass[x, image[CART, Id]]
```

Two immediate applications of this wrapper will now be given for arbitrary collections of squares.

Theorem. The union of any collection of squares is a reflexive relation.

```
In[6]:=  SubstTest[implies, equal[x, sq[t]], REFLEXIVE[U[x]], t → x] // Reverse

Out[6]=  or[not[subclass[x, image[CART, Id]]], REFLEXIVE[U[x]]] == True

In[7]:=  or[not[subclass[x_, image[CART, Id]]], REFLEXIVE[U[x_]]] := True
```

Theorem. The union of any collection of squares is a symmetric relation.

```
In[8]:=  SubstTest[implies, equal[x, sq[t]], SYMMETRIC[U[x]], t → x] // Reverse

Out[8]=  or[equal[inverse[U[x]], U[x]], not[subclass[x, image[CART, Id]]]] == True

In[9]:=  or[equal[inverse[U[x_]], U[x_]], not[subclass[x_, image[CART, Id]]]] := True
```

Theorem. Symmetric rectangles are square.

```
In[10]:=  Map[not, SubstTest[and, implies[and[p1, p2], p3], not[implies[and[p1, p2], p4]],
             {p1 -> equal[x, cart[domain[x], range[x]]], p2 -> equal[x, inverse[x]],
              p3 → equal[domain[x], range[x]], p4 → equal[x, cart[fix[x], fix[x]]]}]] // Reverse

Out[10]=  or[equal[x, cart[fix[x], fix[x]]],
            not[equal[x, cart[domain[x], range[x]]]], not[equal[x, inverse[x]]]] == True

In[11]:=  or[equal[cart[fix[x_], fix[x_]], x_],
            not[equal[cart[domain[x_], range[x_]], x_]], not[equal[inverse[x_], x_]]] := True
```

Theorem. The union of a chain of squares is a square.

```
In[12]:=  Map[not, SubstTest[and, (*implies[p1,p3],*) implies[and[p2, p3], p4],
             implies[and[p1, p2], p5], (*implies[and[p4,p5],p6],*) not[implies[and[p1, p2], p6]],
             {p1 → subclass[x, image[CART, Id]], p2 -> subclass[P[x], chains[S]],
              p3 -> subclass[x, range[CART]], p4 → RECTANGLE[U[x]],
              p5 → SYMMETRIC[U[x]], p6 → equal[U[x], cartsq[fix[U[x]]]]}]] // Reverse

Out[12]=  or[equal[cart[fix[U[x]], fix[U[x]]], U[x]], not[subclass[x, image[CART, Id]]],
            not[subclass[cart[x, x], union[S, inverse[S]]]]] == True

In[13]:=  or[equal[cart[fix[U[x_]], fix[U[x_]]], U[x_]],
            not[subclass[cart[x_, x_], union[S, inverse[S]]]],
            not[subclass[x_, image[CART, Id]]]] := True
```