a reify rule for wo[x]

Johan G. F. Belinfante
2007 May 2

```
In[1]:= SetDirectory["l:"]; << goedel93.02a; << tools.m

        :Package Title: goedel93.02a      2007 May 2 at 3:05 p.m.

        It is now:  2007 May 2 at 19:56

        Loading Simplification Rules

        TOOLS.M                        Revised 2007 May 2

        weightlimit = 40
```

summary

A **reify** rule for the well-ordering wrapper **wo[x]** is derived. The inspiration for this derivation is the following rewrite rule that expresses the fact that a relation is a well-order if every restriction of it is:

```
In[2]:= subclass[image[IMAGE[id[x]], image[CART, Id]], WO]

Out[2]= WELLORDER[composite[Id, x]]
```

The idea is to hide most of the details of the rather complicated definition of wellordering into the constant **WO**, which is independent of **x**, and therefore plays little role in the reification process. An example is provided to show that this all works just fine.

derivation

Lemma. (Temporary rewrite rule.)

```
In[3]:= wo[x] // Normality // Reverse

Out[3]= composite[Id,
          intersection[x, complement[image[V, intersection[complement[fix[x]], domain[x]]]],
           complement[image[V, intersection[complement[fix[x]], range[x]]]],
           complement[image[V, intersection[complement[domain[funpart[LEAST[x]]]],
              complement[set[0]], P[fix[x]]]]]]] == wo[x]
```

```
In[4]:= composite[Id,
          intersection[complement[image[V, intersection[complement[fix[x_]], domain[x_]]]],
           complement[image[V, intersection[complement[fix[x_]], range[x_]]]],
           complement[image[V, intersection[complement[domain[funpart[LEAST[x_]]]],
              complement[set[0]], P[fix[x_]]]]]], x_]] := wo[x]
```

Main Theorem. (A succinct formula for **wo[x]** that hides the complexities of the definition into the constant **WO**.)

```
In[5]:= composite[id[complement[image[V, intersection[
              complement[WO], image[IMAGE[id[x]], image[CART, Id]]]]]], x] // Normality

Out[5]= composite[id[complement[image[V,
             intersection[complement[WO], image[IMAGE[id[x]], image[CART, Id]]]]]], x] == wo[x]

In[6]:= composite[id[complement[image[V, intersection[
              complement[WO], image[IMAGE[id[x_]], image[CART, Id]]]]]], x_] := wo[x]
```

reify rule for wo[x]

Theorem.

```
In[7]:= SubstTest[reify, x, composite[id[complement[image[V, intersection[complement[w],
               image[IMAGE[id[f[x]]], image[CART, Id]]]]]], f[x]], w → WO] // Reverse

Out[7]= reify[x, wo[f[x]]] ==
         composite[id[cart[V, V]], reify[x, f[x]], id[image[inverse[SINGLETON],
            lb[image[inverse[CART], image[inverse[IMAGE[id[reify[x, f[x]]]]]],
               image[inverse[IMAGE[SECOND]], WO]]], image[CART, Id]]]]]

In[8]:= reify[x_, wo[y_]] := composite[id[cart[V, V]], reify[x, y], id[image[inverse[SINGLETON],
            lb[image[inverse[CART], image[inverse[IMAGE[id[reify[x, y]]]]],
               image[inverse[IMAGE[SECOND]], WO]]], image[CART, Id]]]]]
```

an elementary example

In this section the new **reify** rule is used to obtain a variable-free reformulation of the following basic fact:

```
In[9]:= range[wo[x]]

Out[9]= fix[wo[x]]
```

Lemma.

```
In[10]:= equal[intersection[WO, complement[image[inverse[IMAGE[id[cart[V, V]]]], WO]]], 0]

Out[10]= True

In[11]:= intersection[WO, complement[image[inverse[IMAGE[id[cart[V, V]]]], WO]]] := 0
```

```
In[12]:= Map[composite[VERTSECT[#], id[WO]] &, SubstTest[reify, x, range[f[x]], f → wo]]

Out[12]= composite[IMAGE[SECOND],
           id[intersection[WO, image[inverse[IMAGE[id[cart[V, V]]]], WO]]]] ==
          composite[IMAGE[inverse[DUP]],
           id[intersection[WO, image[inverse[IMAGE[id[cart[V, V]]]], WO]]]]

In[13]:= composite[IMAGE[SECOND], id[WO]] := composite[IMAGE[inverse[DUP]], id[WO]]
```