# wrapped WELLORDER predicate

Johan G. F. Belinfante
2004 January 24

```
In[1]:= << goedel53.24b; << tools.m

        :Package Title: goedel53.24b      2004 January 24 at 2:05 p.m.

        It is now:  2004 Jan 26 at 15:51

        Loading Simplification Rules

        TOOLS.M                       Revised 2004 January 3

        weightlimit = 40
```

## summary

The definition of the predicate **WELLORDER[x]** has been wrapped with **class** to facilitate reasoning. In this notebook, some properties of this predicate are established.

## defining clauses

From the wrapped definition of **WELLORDER[x]**, it is easy to rederive the three clauses characterizing this predicate that formed the basis of the work done with **Otter**.

```
In[2]:= implies[WELLORDER[x], REFLEXIVE[x]] // AssertTest

Out[2]= or[not[WELLORDER[x]], subclass[x, cart[fix[x], fix[x]]]] == True

In[3]:= or[not[WELLORDER[x_]], subclass[x_, cart[fix[x_], fix[x_]]]] := True

In[4]:= implies[WELLORDER[x],
          subclass[P[fix[x]], union[domain[funpart[LEAST[x]]], singleton[0]]]] // AssertTest

Out[4]= or[not[WELLORDER[x]],
          subclass[P[fix[x]], union[domain[funpart[LEAST[x]]], singleton[0]]]] == True

In[5]:= or[not[WELLORDER[x_]],
          subclass[P[fix[x_]], union[domain[funpart[LEAST[x_]]], singleton[0]]]] := True

In[6]:= implies[and[REFLEXIVE[x], subclass[P[fix[x]],
            union[domain[funpart[LEAST[x]]], singleton[0]]]], WELLORDER[x]] // AssertTest

Out[6]= or[not[subclass[x, cart[fix[x], fix[x]]]],
          not[subclass[P[fix[x]], union[domain[funpart[LEAST[x]]], singleton[0]]]],
          WELLORDER[x]] == True

In[7]:= or[not[subclass[x_, cart[fix[x_], fix[x_]]]],
          not[subclass[P[fix[x_]], union[domain[funpart[LEAST[x_]]], singleton[0]]]],
          WELLORDER[x_]] := True
```

## examples

Some examples of well−orderings are presented in this section.

```
In[8]:= WELLORDER[0] // AssertTest

Out[8]= WELLORDER[0] == True

In[9]:= WELLORDER[0] := True

In[10]:= WELLORDER[id[singleton[x]]] // AssertTest

Out[10]= WELLORDER[cart[singleton[x], singleton[x]]] == True

In[11]:= WELLORDER[cart[singleton[x_], singleton[x_]]] := True

In[12]:= implies[WELLORDER[x], subclass[x, cart[V, V]]] // AssertTest

Out[12]= or[not[WELLORDER[x]], subclass[x, cart[V, V]]] == True

In[13]:= or[not[WELLORDER[x_]], subclass[x_, cart[V, V]]] := True

In[14]:= WELLORDER[restrict[S, x, x]] // AssertTest

Out[14]= WELLORDER[composite[id[x], S, id[x]]] ==
          subclass[P[x], union[fix[composite[E, BIGCAP]], singleton[0]]]

In[15]:= WELLORDER[composite[id[x_], S, id[x_]]] :=
          subclass[P[x], union[fix[composite[E, BIGCAP]], singleton[0]]]
```

Corollary: The restriction of the subset relation to the class of ordinal numbers is a wellordering.

```
In[16]:= WELLORDER[composite[id[OMEGA], S, id[OMEGA]]]

Out[16]= True
```

## lemma

The following lemma is needed to deduce Theorem **WO−DO−LT**.

```
In[17]:= SubstTest[implies, subclass[x, u], subclass[image[x, y], image[u, y]],
              u -> cart[y, z]]

Out[17]= or[not[subclass[x, cart[y, z]]], subclass[image[x, y], z]] == True

In[18]:= or[not[subclass[x_, cart[y_, z_]]], subclass[image[x_, y_], z_]] := True
```

Corollary.

```
In[19]:= implies[REFLEXIVE[x], invariant[x, fix[x]]]

Out[19]= True
```

## Theorem WO−DO−LT

Theorem **WO−DO−LT** is derived using a combination of **AssertTest** and double negation.

```
In[20]:= or[not[WELLORDER[x]], equal[domain[LEAST[x]],
            intersection[complement[singleton[0]], P[fix[x]]]]] // AssertTest // MapNotNot

Out[20]= or[equal[domain[LEAST[x]], intersection[complement[singleton[0]], P[fix[x]]]],
          not[WELLORDER[x]]] == True
```

If **x** is a well ordering, then every nunempty subset of fixed points has a least member.

```
In[21]:= or[equal[domain[LEAST[x_]], intersection[complement[singleton[0]], P[fix[x_]]]],
          not[WELLORDER[x_]]] := True
```

## Theorem WO−FU−LT

If the domain of a relation is equal to the domain of its funpart, then the relation is a function. This is the argument used to derive Theorem **WO−FU−LT**.

```
In[22]:= SubstTest[implies, and[equal[u, v], subclass[v, w]], subclass[u, w],
            {u -> domain[LEAST[x]], v -> intersection[complement[singleton[0]], P[fix[x]]],
          w -> domain[funpart[LEAST[x]]]}]

Out[22]= or[FUNCTION[LEAST[x]],
          not[equal[domain[LEAST[x]], intersection[complement[singleton[0]], P[fix[x]]]]],
          not[subclass[P[fix[x]], union[domain[funpart[LEAST[x]]], singleton[0]]]] == True

In[23]:= (% /. x -> x_) /. Equal -> SetDelayed
```

If **x** is a wellordering then a set can have only one least member. Therefore **LEAST[x]** is a function

```
In[24]:= Map[not, SubstTest[and, implies[p1, p2], implies[p1, p3],
          implies[and[p2, p3], p4], not[implies[p1, p4]], {p1 -> WELLORDER[x],
                p2 -> subclass[P[fix[x]], union[domain[funpart[LEAST[x]]], singleton[0]]],
                p3 ->
            equal[domain[LEAST[x]], intersection[complement[singleton[0]], P[fix[x]]]],
                p4 -> FUNCTION[LEAST[x]]}]]

Out[24]= or[FUNCTION[LEAST[x]], not[WELLORDER[x]]] == True

In[25]:= or[FUNCTION[LEAST[x_]], not[WELLORDER[x_]]] := True
```