Theorem WO−TO−1

Johan G. F. Belinfante
2004 January 24

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

        :Package Title: goedel53.24c      2004 January 24 at 3:45 p.m.

        It is now:  2004 Jan 26 at 16:7

        Loading Simplification Rules

        TOOLS.M                      Revised 2004 January 3

        weightlimit = 40
```

summary

Theorem **WO−TO−1** says that if **x** is a wellordering then any two elements of **fix[x]** can be compared.

inverse PAIRSET images

The derivation will be based on these two formulas:

```
In[2]:= image[inverse[PAIRSET], domain[LEAST[x]]] // RelnNormality

Out[2]= image[inverse[PAIRSET], domain[LEAST[x]]] ==
         union[composite[x, id[fix[x]]], composite[id[fix[x]], inverse[x]]]

In[3]:= image[inverse[PAIRSET], domain[LEAST[x_]]] :=
         union[composite[x, id[fix[x]]], composite[id[fix[x]], inverse[x]]]

In[4]:= image[inverse[PAIRSET], dif[P[fix[x]], singleton[0]]] // RelnNormality

Out[4]= image[inverse[PAIRSET], intersection[complement[singleton[0]], P[fix[x]]]] ==
         cart[fix[x], fix[x]]

In[5]:= image[inverse[PAIRSET], intersection[complement[singleton[0]], P[fix[x_]]]] :=
         cart[fix[x], fix[x]]
```

derivation

Lemma

```
In[6]:= SubstTest[implies, equal[u, v], equal[image[w, u], image[w, v]],
          {u -> intersection[complement[singleton[0]], P[fix[x]]],
        v -> domain[LEAST[x]], w -> inverse[PAIRSET]}]

Out[6]= or[equal[cart[fix[x], fix[x]],
           union[composite[x, id[fix[x]]], composite[id[fix[x]], inverse[x]]]], not[
           equal[domain[LEAST[x]], intersection[complement[singleton[0]], P[fix[x]]]]]] == True

In[7]:= (% /. x -> x_) /. Equal -> SetDelayed
```

Lemma.

```
In[8]:= Map[not, SubstTest[and, implies[p1, p2],
          implies[p2, p3], implies[p3, p4], implies[p4, p5], implies[p1, p6],
                not[implies[p1, and[p5, p6]]], {p1 -> WELLORDER[x],
           p2 -> equal[domain[LEAST[x]], intersection[complement[singleton[0]], P[fix[x]]]],
            p3 -> equal[cart[fix[x], fix[x]],
             union[composite[x, id[fix[x]]], composite[id[fix[x]], inverse[x]]]],
            p4 -> subclass[cart[fix[x], fix[x]], union[composite[x, id[fix[x]]],
              composite[id[fix[x]], inverse[x]]]],
            p5 -> subclass[cart[fix[x], fix[x]], union[x, inverse[x]]], p6 -> REFLEXIVE[x],
           p7 -> subclass[domain[x], fix[x]], p8 -> subclass[range[x], fix[x]]}]]

Out[8]= or[not[WELLORDER[x]], subclass[cart[fix[x], fix[x]], union[x, inverse[x]]]] == True

In[9]:= (% /. x -> x_) /. Equal -> SetDelayed
```

Lemma.

```
In[10]:= equiv[and[subclass[x, cart[fix[x], fix[x]]], subclass[domain[x], fix[x]],
             subclass[range[x], fix[x]]], subclass[x, cart[fix[x], fix[x]]]] // not // not

Out[10]= True

In[11]:= and[subclass[x_, cart[fix[x_], fix[x_]]], subclass[domain[x_], fix[x_]],
           subclass[range[x_], fix[x_]]] := subclass[x, cart[fix[x], fix[x]]]
```

Lemma.

```
In[12]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> union[x, inverse[x]], v -> cart[fix[x], fix[x]]}]

Out[12]= and[subclass[x, cart[fix[x], fix[x]]],
           subclass[cart[fix[x], fix[x]], union[x, inverse[x]]]] ==
          equal[cart[fix[x], fix[x]], union[x, inverse[x]]]

In[13]:= and[subclass[x_, cart[fix[x_], fix[x_]]],
           subclass[cart[fix[x_], fix[x_]], union[x_, inverse[x_]]]] :=
          equal[cart[fix[x], fix[x]], union[x, inverse[x]]]
```

Theorem **WO−TO−1**. Any two fixed points of a well−ordering can be compared.

```
In[14]:= SubstTest[and, implies[p1, p2], implies[p1, p3],
             {p1 -> WELLORDER[x], p2 -> REFLEXIVE[x],
           p3 -> subclass[cart[fix[x], fix[x]], union[x, inverse[x]]]}] // Reverse

Out[14]= or[equal[cart[fix[x], fix[x]], union[x, inverse[x]]], not[WELLORDER[x]]] == True

In[15]:= or[equal[cart[fix[x_], fix[x_]], union[x_, inverse[x_]]], not[WELLORDER[x_]]] := True
```