FUNCTION[LEAST[x]]

Johan G. F. Belinfante
2004 February 12

In[1]:= `<< goedel54.11a; << tools.m`

```
:Package Title: goedel54.11a      2004 February 11 at 5:30 p.m.

It is now:  2004 Feb 13 at 15:2

Loading Simplification Rules

TOOLS.M                        Revised 2004 January 3

weightlimit = 40
```

summary

If **x** is an antisymmetric relation, then the relations **GREATEST[x]** and **LEAST[x]** are functions. These facts had been proved 2000 November 24 using **Otter**, and added to the **GOEDEL** program at that time as conditional rules:

In[2]:= `Begin["Goedel‘Private‘"];`

This is Theorem **GT−FU−2**, proved 2000 November 24 using **Otter**:

In[3]:= `InfoMatch[FUNCTION[GREATEST[x_]]]`

Out[3]//TableForm=
```
FUNCTION[GREATEST[x_]] := True /; subclass[intersection[x, inverse[x]], Id]
```

This is Theorem **LT−FU**, proved using **Otter**:

In[4]:= `InfoMatch[FUNCTION[LEAST[x_]]]`

Out[4]//TableForm=
```
FUNCTION[LEAST[x_]] := True /; subclass[intersection[x, inverse[x]], Id]
```

The aim of this notebook is to rederive these two formulas using the **GOEDEL** program, and to supplement these conditional rewrite rules with clauses that can be used for reasoning. Such reasoning was used in the notebook **WO−WRAP.NB** dated 2004 January 24 in connection with a rederivation of Theorem **WO−FU−LT** which asserts that if **x** is a wellordering, then **LEAST[x]** is a function.

GREATEST

The rules for **GREATEST[x]** are the focus of this section. The starting point is the following derivation of Theorem **GT−FU−1** that had also been proved 2000 November 24 using **Otter**:

```
In[5]:= SubstTest[implies, subclass[u, v], subclass[composite[u, w], composite[v, w]],
          {u -> GREATEST[x], v -> UB[x], w -> inverse[GREATEST[x]]}]

Out[5]= subclass[composite[GREATEST[x], inverse[GREATEST[x]]], x] == True

In[6]:= subclass[composite[GREATEST[x_], inverse[GREATEST[x_]]], x_] := True
```

From this, one immediately derives Theorem **GT−FU−2** as a corollary:

```
In[7]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
            {u -> composite[GREATEST[x], inverse[GREATEST[x]]],
          v -> intersection[x, inverse[x]], w -> Id}]

Out[7]= or[FUNCTION[GREATEST[x]], not[subclass[intersection[x, inverse[x]], Id]]] == True

In[8]:= or[FUNCTION[GREATEST[x_]], not[subclass[intersection[x_, inverse[x_]], Id]]] := True
```

LEAST

The results for **LEAST[x]** follow immediately by replacing **x** with **inverse[x]**. (This result did not appear in the **Otter** work.)

```
In[9]:= SubstTest[subclass, composite[GREATEST[y], inverse[GREATEST[y]]],
          intersection[y, inverse[y]], y -> inverse[x]]

Out[9]= subclass[composite[LEAST[x], inverse[LEAST[x]]], x] == True

In[10]:= subclass[composite[LEAST[x_], inverse[LEAST[x_]]], x_] := True
```

The function rule also generalizes, yielding Theorem **LT−FU**:

```
In[11]:= SubstTest[implies, subclass[intersection[y, inverse[y]], Id],
          FUNCTION[GREATEST[y]], y -> inverse[x]]

Out[11]= or[FUNCTION[LEAST[x]], not[subclass[intersection[x, inverse[x]], Id]]] == True

In[12]:= or[FUNCTION[LEAST[x_]], not[subclass[intersection[x_, inverse[x_]], Id]]] := True
```