power set initial segment example

Johan G. F. Belinfante
2010 July 20

```
In[1]:= SetDirectory["l:"]; << goedel.10jul19a;<< tools.m

:Package Title: goedel.10jul19a               2010 July 19 at 3:55 p.m.

It is now:  2010 Jul 20 at 10:37

Loading Simplification Rules

TOOLS.M                                        Revised 2010 February 26

weightlimit = 40
```

summary

Harzheim defines a set **s** to be a **initial segment** of a relation **r** if **image[inverse[r], s] = s**.

```
In[2]:=  "Egbert Harzheim, Ordered Sets, Advances in Mathematics, volume 7, Springer
          Science+Business Media, Inc., 2005.  ISBN 0387-24219-8.  QA171.48 .H37";
```

Example 8.11 on page 32 of this book states that the set of finite subsets of a set **x** is an initial segment of the subset relation restricted to the power set of **x**. A variable-free expression of this fact is derived in this notebook.

derivation

Theorem. (Harzheim's theorem 8.11.)

```
In[3]:= SubstTest[member, u, fix[IMAGE[v]], {u -> intersection[FINITE, P[setpart[x]]],
          v -> restrict[inverse[S], P[setpart[x]], P[setpart[x]]]}] // Reverse

Out[3]= member[intersection[FINITE, P[setpart[x]]],
          fix[composite[IMAGE[inverse[S]], IMAGE[id[P[setpart[x]]]]]]] == True

In[4]:= member[intersection[FINITE, P[setpart[x_]]],
          fix[composite[IMAGE[inverse[S]], IMAGE[id[P[setpart[x_]]]]]]] := True
```

a variable-free statement

Lemma.

```
In[5]:= subclass[composite[x, POWER], composite[y, POWER]] // AssertTest

Out[5]= subclass[composite[x, POWER], composite[y, POWER]] ==
         subclass[composite[x, id[range[POWER]]], y]

In[6]:= subclass[composite[x_, POWER], composite[y_, POWER]] :=
         subclass[composite[x, id[range[POWER]]], y]
```

Lemma.

```
In[7]:= dif[composite[IMAGE[id[FINITE]], POWER], composite[
           fix[composite[inverse[SECOND], IMAGE[inverse[S]], CAP]], POWER]] // FastReifNormality

Out[7]= composite[
          intersection[complement[fix[composite[inverse[SECOND], IMAGE[inverse[S]], CAP]]],
           IMAGE[id[FINITE]]], POWER] == 0

In[8]:= % /. Equal→SetDelayed
```

Theorem. A variable-free statement.

```
In[9]:= SubstTest[empty, dif[u, v], {u -> composite[IMAGE[id[FINITE]], POWER],
          v -> composite[fix[composite[inverse[SECOND], IMAGE[inverse[S]], CAP]], POWER]}]

Out[9]= subclass[composite[IMAGE[id[FINITE]], id[range[POWER]]],
          fix[composite[inverse[SECOND], IMAGE[inverse[S]], CAP]]] == True

In[10]:= subclass[composite[IMAGE[id[FINITE]], id[range[POWER]]],
           fix[composite[inverse[SECOND], IMAGE[inverse[S]], CAP]]] := True
```

The connection with initial segments will now be established. The class of initial segments of a restriction of the subset relation **S** is the class of final segments for the same restriction of **inverse[S]**. In general, the class of final segments of a relation **x** is **fix[IMAGE[x]]**. The relation of being a final segment is:

```
In[17]:= reify[x, fix[IMAGE[x]]]

Out[17]= fix[composite[inverse[SECOND], IMG]]
```

The following membership rule makes this fact more explicit:

```
In[18]:= member[pair[x, u], fix[composite[inverse[SECOND], IMG]]]

Out[18]= and[equal[u, image[x, u]], member[x, V]]
```

Theorem. A simplification rule.

```
In[11]:= composite[fix[composite[inverse[SECOND], IMG]],
           IMAGE[composite[id[inverse[S]], inverse[FIRST]]]] // VSNormality

Out[11]= composite[fix[composite[inverse[SECOND], IMG]],
           IMAGE[composite[id[inverse[S]], inverse[FIRST]]]] ==
          fix[composite[inverse[SECOND], IMAGE[inverse[S]], CAP]]
```

```
In[12]:= composite[fix[composite[inverse[SECOND], IMG]],
           IMAGE[composite[id[inverse[S]], inverse[FIRST]]]] :=
          fix[composite[inverse[SECOND], IMAGE[inverse[S]], CAP]]
```

The two-sided restriction of **inverse[S]** to a power class is automatically rewritten as a one-sided restriction.

```
In[19]:= restrict[inverse[S], P[x], P[x]]

Out[19]= composite[inverse[S], id[P[x]]]
```

Theorem. A variable-free statement involving final segments of two-sided restrictions of **inverse[S]**.

```
In[15]:= Map[subclass[composite[IMAGE[id[FINITE]], POWER], #] &,
          Map[composite[fix[composite[inverse[SECOND], IMG]], #] &,
           composite[IMAGE[id[inverse[S]]], CART, DUP, POWER] // RelnNormality]]

Out[15]= subclass[composite[IMAGE[id[FINITE]], id[range[POWER]]], composite[
            fix[composite[inverse[SECOND], IMG]], IMAGE[id[inverse[S]]], CART, DUP]] == True

In[16]:= subclass[composite[IMAGE[id[FINITE]], id[range[POWER]]], composite[
            fix[composite[inverse[SECOND], IMG]], IMAGE[id[inverse[S]]], CART, DUP]] := True
```