equations for ENUMS

Johan G. F. Belinfante
2011 September 15

```
In[1]:= SetDirectory["l:"]; << goedel.11sep12a

:Package Title: goedel.11sep12a          2011 September 12 at 12:40 noon

Loading takes about twelve minutes, half that time due to builtin pauses.

It is now:  2011 Sep 15 at 11:15

Loading Simplification Rules

TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

weightlimit = 40

Loading completed.

It is now:  2011 Sep 15 at 11:27
```

summary

Two equations for the class **ENUMS** of enumerators are derived.

derivation

Theorem. A formula for the class **ENUMS**.

```
In[2]:= SubstTest[class, x, member[x, y], y -> intersection[FUNS,
          P[cart[OMEGA, OMEGA]], image[inverse[IMAGE[SWAP]], subcommutant[inverse[E]]]]]

Out[2]= intersection[FUNS,
          image[inverse[IMAGE[SWAP]], subcommutant[inverse[E]]], P[cart[OMEGA, OMEGA]]] == ENUMS

In[3]:= intersection[FUNS,
          image[inverse[IMAGE[SWAP]], subcommutant[inverse[E]]], P[cart[OMEGA, OMEGA]]] := ENUMS
```

Another formula for **ENUMS** will be derived in which **subcommutant[inverse[E]]** is replaced with **monotone[E, E]**.

Theorem. A connection between **monotone[x, x]** and **subcommutant[x]**.

```
In[16]:= (Map[equal[V, #] &, SubstTest[class, t, or[not[subclass[P[t], z]],
              not[member[t, V]], not[subclass[image[inverse[s], domain[t]], domain[t]]],
              subclass[composite[t, s], composite[s, t]]],
            z → intersection[FUNS, monotone[s, s]]]]) /. s → inverse[x]

Out[16]= subclass[
           image[IMAGE[SWAP], intersection[monotone[x, x], U[image[MAP, cart[invar[x], V]]]]],
           subcommutant[x]] == True

In[17]:= subclass[image[IMAGE[SWAP], intersection[monotone[x_, x_],
            U[image[MAP, cart[invar[x_], V]]]]], subcommutant[x_]] := True
```

Corollary. A special case of the above theorem needed here.

```
In[19]:= SubstTest[subclass,
           image[IMAGE[SWAP], intersection[monotone[x, x], U[image[MAP, cart[invar[x], V]]]]],
           subcommutant[x], x → inverse[E]] // Reverse

Out[19]= subclass[
           image[IMAGE[SWAP], intersection[monotone[E, E], U[image[MAP, cart[FULL, V]]]]],
           subcommutant[inverse[E]]] == True

In[20]:= subclass[
           image[IMAGE[SWAP], intersection[monotone[E, E], U[image[MAP, cart[FULL, V]]]]],
           subcommutant[inverse[E]]] := True
```

Lemma.

```
In[22]:= U[image[MAP, cart[x, P[y]]]] // Normality // Reverse

Out[22]= intersection[P[complement[cart[V, complement[y]]]], U[image[MAP, cart[x, V]]]] ==
          U[image[MAP, cart[x, P[y]]]]

In[23]:= intersection[P[complement[cart[V, complement[y_]]]], U[image[MAP, cart[x_, V]]]] :=
          U[image[MAP, cart[x, P[y]]]]
```

Theorem.

```
In[24]:= intersection[P[cart[V, y]], U[image[MAP, cart[x, V]]]] // Normality

Out[24]= intersection[P[cart[V, y]], U[image[MAP, cart[x, V]]]] == U[image[MAP, cart[x, P[y]]]]

In[25]:= intersection[P[cart[V, y_]], U[image[MAP, cart[x_, V]]]] := U[image[MAP, cart[x, P[y]]]]
```

Lemma. An upper bound for **ENUMS**.

```
In[26]:= SubstTest[subclass, ENUMS, intersection[u, v, w],
           {u -> FUNS, v → P[cart[V, OMEGA]], w -> image[inverse[IMAGE[FIRST]], OMEGA]}] // Reverse

Out[26]= subclass[ENUMS, U[image[MAP, cart[OMEGA, P[OMEGA]]]]] == True

In[27]:= subclass[ENUMS, U[image[MAP, cart[OMEGA, P[OMEGA]]]]] := True
```

An inclusion in the opposite direction will now be derived.

Lemma.

```
In[28]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u -> intersection[monotone[E, E], U[image[MAP, cart[OMEGA, P[OMEGA]]]]],
            v -> U[image[MAP, cart[OMEGA, P[OMEGA]]]], w → P[cartsq[OMEGA]]}] // Reverse

Out[28]= subclass[U[intersection[monotone[E, E], U[image[MAP, cart[OMEGA, P[OMEGA]]]]]],
           cart[OMEGA, OMEGA]] == True

In[29]:= % /. Equal → SetDelayed
```

Lemma.

```
In[30]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u -> intersection[monotone[E, E], U[image[MAP, cart[OMEGA, P[OMEGA]]]]],
            v -> intersection[monotone[E, E], U[image[MAP, cart[FULL, V]]]],
            w → image[inverse[IMAGE[SWAP]], subcommutant[inverse[E]]]}] // Reverse

Out[30]= subclass[image[IMAGE[SWAP], intersection[monotone[E, E],
             U[image[MAP, cart[OMEGA, P[OMEGA]]]]]], subcommutant[inverse[E]]] == True

In[31]:= % /. Equal → SetDelayed
```

Lemma. A lower bound for **ENUMS**.

```
In[32]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
             {u -> intersection[monotone[E, E], U[image[MAP, cart[OMEGA, P[OMEGA]]]]],
              v → intersection[y, image[inverse[IMAGE[SWAP]], subcommutant[inverse[E]]]],
              w -> ENUMS}] /. y → intersection[FUNS, P[cartsq[OMEGA]]] // Reverse

Out[32]= subclass[intersection[monotone[E, E], U[image[MAP, cart[OMEGA, P[OMEGA]]]]], ENUMS] ==
          True

In[33]:= % /. Equal → SetDelayed
```

The two inclusions in opposite directions can be combined into an equation for **ENUMS**.

Theorem. Another formula for the class **ENUMS**.

```
In[34]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> intersection[monotone[E, E], U[image[MAP, cart[OMEGA, P[OMEGA]]]]], v -> ENUMS}]

Out[34]= equal[ENUMS, intersection[monotone[E, E], U[image[MAP, cart[OMEGA, P[OMEGA]]]]]] == True

In[35]:= intersection[monotone[E, E], U[image[MAP, cart[OMEGA, P[OMEGA]]]]] := ENUMS
```