U[ENUMS] = id[Ω] ∘ S ∘ id[Ω]

Johan G. F. Belinfante
2010 December 6

```
In[1]:= SetDirectory["l:"]; << goedel.10dec05a

        :Package Title: goedel.10dec05a              2010 December 5 at 11:00 a.m.

        It is now:  2010 Dec 6 at 6:37

        Loading Simplification Rules

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

        weightlimit = 40
```

summary

A formula for the sum class of the class of enumerations is derived.

derivation

The uniqueness theorem for enumerations will be used to derive an explicit formula for the enumeration function for the union of an ordinal α and the singleton of a greater ordinal β. This enumeration function is the union of the identity on α with the single point **{pair[α, β]}**.

Lemma.

```
In[2]:= FUNCTION[union[cart[set[ord[x]], set[ord[y]]], id[ord[x]]]] // AssertTest

Out[2]= FUNCTION[union[cart[set[ord[x]], set[ord[y]]], id[ord[x]]]] == True

In[3]:= FUNCTION[union[cart[set[ord[x_]], set[ord[y_]]], id[ord[x_]]]] := True
```

Lemma.

```
In[4]:= SubstTest[and, member[t, FUNS], subclass[t, cartsq[OMEGA]],
          subcommute[t, E], t -> union[id[ord[x]], cart[set[ord[x]], set[ord[y]]]]]

Out[4]= member[union[cart[set[ord[x]], set[ord[y]]], id[ord[x]]], ENUMS] ==
          not[member[ord[y], ord[x]]]

In[5]:= member[union[cart[set[ord[x_]], set[ord[y_]]], id[ord[x_]]], ENUMS] :=
          not[member[ord[y], ord[x]]]
```

The following theorem is not actually needed later, but may help make matters more explicit.

Theorem. An explicit formula for the enumeration of the set $\alpha \bigcup \{\beta\}$ when $\alpha \subset \beta$.

```
In[6]:= SubstTest[implies, and[member[u, ENUMS], member[v, ENUMS], equal[range[u], range[v]]],
          equal[u, v], {u -> union[cart[set[ord[x]], set[ord[y]]], id[ord[x]]],
           v -> enum[union[ord[x], set[ord[y]]]]}] // Reverse

Out[6]= or[equal[enum[union[ord[x], set[ord[y]]]],
           union[cart[set[ord[x]], set[ord[y]]], id[ord[x]]]], member[ord[y], ord[x]]] == True

In[7]:= or[equal[enum[union[ord[x_], set[ord[y_]]]], union[
            cart[set[ord[x_]], set[ord[y_]]], id[ord[x_]]]], member[ord[y_], ord[x_]]] := True
```

a formula for U[ENUMS]

Lemma. If $\alpha \subset \beta$, then **pair[α, β] ∈ U[ENUMS]**.

```
In[8]:= SubstTest[implies, and[member[u, v], member[v, w]],
          member[u, U[w]], {u → pair[ord[x], ord[y]],
           v -> union[cart[set[ord[x]], set[ord[y]]], id[ord[x]]], w → ENUMS}] // Reverse

Out[8]= or[member[ord[y], ord[x]], member[pair[ord[x], ord[y]], U[ENUMS]]] == True

In[9]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. (Eliminate the **ord** wrappers.)

```
In[10]:= SubstTest[implies, and[equal[x, ord[u]], equal[y, ord[v]]],
           or[member[y, x], member[pair[x, y], U[ENUMS]]], {u → x, v → y}] // Reverse

Out[10]= or[member[y, x], member[pair[x, y], U[ENUMS]],
           not[member[x, OMEGA]], not[member[y, OMEGA]]] == True

In[11]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. (Eliminate the variables **x** and **y**.)

```
In[12]:= Map[empty[composite[Id, complement[#]]] &,
          SubstTest[class, pair[x, y], or[member[y, x], member[pair[x, y], t],
            not[member[x, OMEGA]], not[member[y, OMEGA]]], t → U[ENUMS]]]

Out[12]= subclass[composite[id[OMEGA], S, id[OMEGA]], U[ENUMS]] == True

In[13]:= % /. Equal → SetDelayed
```

The reverse inclusion was derived on a previous occasion, and can be combined with the above result to obtain a rewrite rule for the sum class of the class of enumerations.

Corollary.

```
In[15]:= SubstTest[and, subclass[u, v], subclass[v, u],
           {u -> U[ENUMS], v -> composite[id[OMEGA], S, id[OMEGA]]}]

Out[15]= equal[composite[id[OMEGA], S, id[OMEGA]], U[ENUMS]] == True

In[17]:= U[ENUMS] := composite[id[OMEGA], S, id[OMEGA]]
```