an inclusion for enum[x]

Johan G. F. Belinfante
2010 November 23

```
In[1]:= SetDirectory["l:"]; << goedel.10nov22a

:Package Title: goedel.10nov22a              2010 November 22 at 2:05 p.m.

It is now:  2010 Nov 23 at 16:23

Loading Simplification Rules

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

weightlimit = 40
```

summary

The ordinal enumeration function **enum[x]** satisfies the inclusion

$$\mathbf{U[image[enum[x], y] \subset APPLY[enum[x], U[y]]}$$

for every ordinal **y**.

derivation

Lemma.

```
In[2]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
          {u → image[enum[x], y], v → APPLY[enum[x], y], w → succ[APPLY[enum[x], y]]}] // Reverse

Out[2]= subclass[image[enum[x], y], succ[APPLY[enum[x], y]]] == True

In[3]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[4]:= Map[subclass[#, succ[APPLY[enum[x], y]]] &,
          SubstTest[image, enum[x], union[y, z], z → set[y]]] // Reverse

Out[4]= subclass[image[enum[x], succ[y]], succ[APPLY[enum[x], y]]] == True

In[5]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[6]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
          subclass[u, w], {u -> image[enum[x], succ[y]],
           v -> succ[APPLY[enum[x], y]], w → P[APPLY[enum[x], y]]}] // Reverse

Out[6]= subclass[U[image[enum[x], succ[y]]], APPLY[enum[x], y]] == True

In[7]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem.

```
In[8]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
          subclass[u, w], {u -> U[image[enum[x], ord[y]]],
           v → U[image[enum[x], succ[U[ord[y]]]]], w → APPLY[enum[x], U[ord[y]]]}] // Reverse

Out[8]= subclass[U[image[enum[x], ord[y]]], APPLY[enum[x], U[ord[y]]]] == True

In[9]:= subclass[U[image[enum[x_], ord[y_]]], APPLY[enum[x_], U[ord[y_]]]] := True
```

Corollary. (Eliminate the **ord** wrapper.)

```
In[10]:= SubstTest[implies, equal[y, ord[t]],
           subclass[U[image[enum[x], y]], APPLY[enum[x], U[y]]], t → y] // Reverse

Out[10]= or[not[member[y, OMEGA]], subclass[U[image[enum[x], y]], APPLY[enum[x], U[y]]]] == True

In[11]:= or[not[member[y_, OMEGA]],
           subclass[U[image[enum[x_], y_]], APPLY[enum[x_], U[y_]]]] := True
```

Counterexample. In general the inclusion cannot be sharpened to an equation.

```
In[12]:= equal[U[image[enum[x], ord[y]]], APPLY[enum[x], U[ord[y]]]] /. {x → PRIMES, y → 0}

Out[12]= False
```