ordinal addition, part 4. examples

Johan G. F. Belinfante
2010 December 16

```
In[1]:= SetDirectory["l:"]; << goedel.10dec16a

        :Package Title: goedel.10dec16a            2010 December 16 at 5:10 p.m.

        It is now:  2010 Dec 16 at 17:27

        Loading Simplification Rules

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

        weightlimit = 40
```

summary

This fourth notebook in a series about ordinal addition is concerned with examples. It is shown that the restriction of ordinal addition to natural numbers is the usual addition of natural numbers. A standard counterexample is used to show that ordinal addition is not commutative.

a reify rule for ordplus

A **reify** rule for **ordplus** is derived in this section and used in the next.

Theorem. A **reify** rule for **ordplus**.

```
In[2]:= SubstTest[reify, x, composite[t, LEFT[f[x]]], t → ORDADD] // Reverse

Out[2]= reify[x, ordplus[f[x]]] ==
         composite[SWAP, inverse[rotate[composite[ORDADD, SWAP]]], VERTSECT[reify[x, f[x]]]]

In[3]:= reify[x_, ordplus[y_]] :=
         composite[SWAP, inverse[rotate[composite[ORDADD, SWAP]]], VERTSECT[reify[x, y]]]
```

restriction of ordinal addition to natural numbers

In this section it is shown that ordinal addition is an extension of the usual addition of natural numbers. The starting point is a lemma that makes use of the fact that the restriction of **enum[x]** to ω is **ordlist[x]**. The following available result is used here.

```
In[5]:= ordlist[complement[nat[x]]]

Out[5]= plus[nat[x]]
```

Lemma. Connection between **ordplus[x]** and **plus[x]** for $\mathbf{x} \in \omega$.

```
In[6]:= ((SubstTest[composite, enum[complement[t]], id[omega], t → ord[w]]) /. w → nat[x]) //
         Reverse

Out[6]= composite[ordplus[nat[x]], id[omega]] == plus[nat[x]]

In[7]:= composite[ordplus[nat[x_]], id[omega]] := plus[nat[x]]
```

All that is needed now is to apply **reify** to this formula.

Theorem. The restriction of ordinal addition to natural numbers.

```
In[8]:= Map[rotate[inverse[composite[#, id[omega]]]] &,
         SubstTest[reify, x, composite[ordplus[nat[x]], id[w]], w → omega]]

Out[8]= composite[ORDADD, id[cart[omega, omega]]] == NATADD

In[9]:= composite[ORDADD, id[cart[omega, omega]]] := NATADD
```

Corollary. Ordinal addition for natural numbers.

```
In[10]:= ApComp[ORDADD, id[cart[omega, omega]], PAIR[nat[x], nat[y]]]

Out[10]= ordadd[nat[x], nat[y]] == natadd[nat[x], nat[y]]

In[11]:= ordadd[nat[x_], nat[y_]] := natadd[nat[x], nat[y]]
```

---

another formula for ordplus

If **x** and **y** are ordinals, the inclusion $\mathbf{x} \subset \mathbf{y}$ is equivalent to the negation of $\mathbf{y} \in \mathbf{x}$. This observation is used in this section to derive a new formula for **ordplus[x]**.

Lemma. The class of ordinals not less than **ord[x]** is the class of ordinals greater than or equal to **ord[x]**.

```
In[12]:= equal[intersection[OMEGA, complement[ord[x]]],
           intersection[OMEGA, image[S, set[ord[x]]]]] // AssertTest

Out[12]= equal[intersection[OMEGA, complement[ord[x]]],
           intersection[OMEGA, image[S, set[ord[x]]]]] == True

In[13]:= intersection[OMEGA, image[S, set[ord[x_]]]] := intersection[OMEGA, complement[ord[x]]]
```

Theorem. An alternate formula for **ordplus**.

```
In[14]:= SubstTest[enum, intersection[OMEGA, t], t -> image[S, set[ord[x]]]]

Out[14]= enum[image[S, set[ord[x]]]] == ordplus[ord[x]]

In[15]:= enum[image[S, set[ord[x_]]]] := ordplus[ord[x]]
```

a recursion relation for ordplus

The following rewrite rule is oriented to eliminate **HULL** because that helps with evaluating ordinal sums.

Theorem.

```
In[16]:= SubstTest[HULL, intersection[w, image[S, set[ord[x]]]], w → OMEGA] // Reverse

Out[16]= HULL[intersection[OMEGA, complement[ord[x]]]] ==
          composite[TC, id[P[OMEGA]], ADJOIN[ord[x]]]

In[17]:= HULL[intersection[OMEGA, complement[ord[x_]]]] :=
          composite[TC, id[P[OMEGA]], ADJOIN[ord[x]]]
```

Theorem. A recursion equation for **ordplus**.

```
In[18]:= SubstTest[composite, HULL[intersection[OMEGA, t]],
           IMAGE[enum[t]], id[domain[enum[t]]], t → complement[ord[x]]] // Reverse

Out[18]= composite[TC, id[P[OMEGA]], ADJOIN[ord[x]], IMAGE[ordplus[ord[x]]], id[OMEGA]] ==
          ordplus[ord[x]]

In[19]:= composite[TC, id[P[OMEGA]], ADJOIN[ord[x_]],
           IMAGE[ordplus[ord[x_]]], id[OMEGA]] := ordplus[ord[x]]
```

right-addition for 1 = set[0]

The recursion relation derived in the preceding section is used to derive a formula for right-addition by the ordinal **1 = set[0]**.

Lemma. A temporary rule with an **ord** wrapper.

```
In[20]:= ApComp[composite[TC, id[P[OMEGA]], ADJOIN[ord[x]], IMAGE[enum[complement[ord[x]]]]],
           id[OMEGA], set[0]] // Reverse

Out[20]= ordadd[ord[x], set[0]] == succ[ord[x]]

In[21]:= ordadd[ord[x_], set[0]] := succ[ord[x]]
```

Lemma. Eliminating the **ord** wrapper.

```
In[22]:= SubstTest[implies, equal[x, ord[t]], equal[ordadd[x, set[0]], succ[x]], t → x] // Reverse
Out[22]= or[equal[ordadd[x, set[0]], succ[x]], not[member[x, OMEGA]]] == True
In[23]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. A general formula for right-addition by **1**.

```
In[24]:= equal[ordadd[x, set[0]],
           union[succ[x], complement[image[V, intersection[OMEGA, set[x]]]]]]
Out[24]= True
In[25]:= ordadd[x_, set[0]] := union[complement[image[V, intersection[OMEGA, set[x]]]], succ[x]]
```

Corollary. A variable-free statement.

```
In[26]:= composite[ORDADD, RIGHT[set[0]]] // RelnNormality
Out[26]= composite[ORDADD, RIGHT[set[0]]] == composite[id[OMEGA], SUCC]
In[27]:= composite[ORDADD, RIGHT[set[0]]] := composite[id[OMEGA], SUCC]
```

Comment. A similar rule can be derived for right-addition by **2 = succ[set[0]]**, and so on.

ordinal addition is not commutative

In this section it is shown that ordinal addition is not commutatve: $\mathbf{1} + \omega = \omega$ whereas $\omega + \mathbf{1} = \mathbf{succ}[\omega]$.

Lemma. Simplification rule.

```
In[28]:= SubstTest[implies, subclass[x, OMEGA],
           equal[tc[x], union[x, U[x]]], x → dif[omega, set[0]]] // Reverse
Out[28]= equal[omega, tc[intersection[omega, complement[set[0]]]]] == True
In[29]:= tc[intersection[omega, complement[set[0]]]] := omega
```

Lemma. Simplifcation rule.

```
In[30]:= Map[tc, ImageComp[enum[complement[x]], id[omega], V] /. x → set[0]] // Reverse
Out[30]= tc[image[ordplus[set[0]], omega]] == omega
In[31]:= tc[image[ordplus[set[0]], omega]] := omega
```

Theorem. $\mathbf{1} + \omega = \omega$.

```
In[32]:= Map[APPLY[#, omega] &, SubstTest[composite, TC, id[P[OMEGA]],
           ADJOIN[ord[w]], IMAGE[ordplus[ord[w]]], id[OMEGA], w → set[0]]]

Out[32]= ordadd[set[0], omega] == omega

In[33]:= ordadd[set[0], omega] := omega
```

On the other hand, the sum in the reverse order is different: $\omega + 1 = \mathbf{succ}[\omega]$.

```
In[34]:= ordadd[omega, set[0]]

Out[34]= succ[omega]
```

Corollary. Ordinal addition is not commutative.

```
In[35]:= Map[not, SubstTest[implies, equal[u, v], equal[APPLY[u, w], APPLY[v, w]],
            {u → ORDADD, v → flip[ORDADD], w → PAIR[set[0], omega]}]] // Reverse

Out[35]= equal[ORDADD, composite[ORDADD, SWAP]] == False

In[36]:= equal[ORDADD, composite[ORDADD, SWAP]] := False
```