adding even and odd numbers

Johan G. F. Belinfante
2006 August 16

summary

If the sum of two natural numbers is even, then either both are even or both are odd. If the sum is odd, one of the numbers is even and the other is odd.

lemmas

The following formula for the set of odd numbers amounts to the statement that any odd number can be written as \( n + (n + 1) \).

| In[2]:= | ImageComp[composite[NATADD, cross[Id, NATADD]], ASSOC, cart[Id, set[set[0]]]] | Reverse
| Out[2]= | image[NATADD, composite[id[omega], SUCC]] := odd

In[3]:= image[NATADD, composite[id[omega], SUCC]] := odd

The following twist rule for NATADD is a variable-free statement of the fact that \( (a + c) + (b + d) = (a + b) + (c + d) \).

| In[4]:= | SubstTest[implies, and[associative[x], equal[flip[x], x]], equal[composite[x, cross[x, x]], TWIST], composite[x, cross[x, x]], x -> NATADD]
| Out[4]= | equal[composite[NATADD, cross[NATADD, NATADD]], composite[NATADD, cross[NATADD, NATADD], TWIST]] := True

| In[5]:= | composite[NATADD, cross[NATADD, NATADD], TWIST] := composite[NATADD, cross[NATADD, NATADD]]

A similar result holds for NATMUL, but it will not be used in this notebook.
The following three results follow from the lemmas in the preceding section.

In[9]:= `ImageComp[composite[NATADD, cross[NATADD, NATADD]], TWIST, cart[Id, Id]]`

Out[9]= `image[NATADD, cart[even, even]] = even`

In[10]:= `image[NATADD, cart[even, even]] := even`

In[11]:= `ImageComp[composite[NATADD, cross[NATADD, NATADD]], TWIST, cart[Id, plus[set[0]]]]`

Out[11]= `image[NATADD, cart[even, odd]] = odd`

In[12]:= `image[NATADD, cart[even, odd]] := odd`

In[13]:= `ImageComp[composite[NATADD, cross[NATADD, NATADD]], TWIST, cart[plus[set[0]], Id]]`

Out[13]= `image[NATADD, cart[odd, even]] = odd`

In[14]:= `image[NATADD, cart[odd, even]] := odd`

Comment. The image of cart[odd,odd] under NATADD is not equal to even. The even number 0 cannot be written as the sum of two odd natural numbers.

adding even and odd numbers

The sum of two even numbers is even.

In[15]:= `SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
                {u -> set[PAIR[x, y]], v -> cart[even, even], w -> NATADD}]`

Out[15]= `or[member[natadd[x, y], even], not[member[x, even]], not[member[y, even]]] = True`

In[16]:= `or[member[natadd[x_, y_], even], not[member[x_, even]], not[member[y_, even]]] := True`

The sum of an even and an odd number is odd.
The remaining case is derived by replacing \( x \) with \( \text{succ}[\text{x}] \).

\[
\text{In}[19]:= \text{SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],}
\{u \rightarrow \text{set}[\text{PAIR}[x, y]], v \rightarrow \text{cart}[\text{even, odd], w \rightarrow \text{NATADD}]\} = \text{True}
\]

\[
\text{In}[20]:= \text{or[member[natadd[x, y, even], not[member[x, even]], not[member[y, odd]]]} = \text{True}
\]

**inverse image lemmas**

Four lemmas are obtained by eliminating variables from the results of the preceding section.

\[
\text{In}[21]:= \text{Map[equal[0, composite[Id, complement[#]]], SubstTest[class,}
\text{pair[x, y], or[member[pair[x, y], v], not[member[x, u]], not[member[y, u]]],}
\{u \rightarrow \text{odd}, v \rightarrow \text{image}[\text{inverse[\text{NATADD}], even}]\}] = \text{Reverse}
\]

\[
\text{Out}[21]= \text{subclass[cart[odd, odd], image[\text{inverse[\text{NATADD}], even}] = True}
\]

\[
\text{In}[22]:= \% / \text{/. \text{Equal} \rightarrow \text{SetDelayed}}
\]

\[
\text{In}[23]:= \text{Map[equal[0, composite[Id, complement[#]]], SubstTest[class,}
\text{pair[x, y], or[member[pair[x, y], v], not[member[x, u]], not[member[y, u]]],}
\{u \rightarrow \text{even}, v \rightarrow \text{image}[\text{inverse[\text{NATADD}], even}]\}] = \text{Reverse}
\]

\[
\text{Out}[23]= \text{subclass[cart[even, even], image[\text{inverse[\text{NATADD}], even}] = True}
\]

\[
\text{In}[24]:= \% / \text{/. \text{Equal} \rightarrow \text{SetDelayed}}
\]

\[
\text{In}[25]:= \text{Map[equal[0, composite[Id, complement[#]]], SubstTest[class,}
\text{pair[x, y], or[member[pair[x, y], w], not[member[x, u]], not[member[y, v]]],}
\{u \rightarrow \text{odd}, v \rightarrow \text{even, w \rightarrow image[\text{inverse[\text{NATADD}], odd}]\}] = \text{Reverse}
\]

\[
\text{Out}[25]= \text{subclass[cart[odd, even], image[\text{inverse[\text{NATADD}], odd}] = True}
\]

\[
\text{In}[26]:= \% / \text{/. \text{Equal} \rightarrow \text{SetDelayed}}
\]

\[
\text{In}[27]:= \text{Map[equal[0, composite[Id, complement[#]]], SubstTest[class,}
\text{pair[x, y], or[member[pair[x, y], w], not[member[x, u]], not[member[y, v]]],}
\{u \rightarrow \text{even}, v \rightarrow \text{odd, w \rightarrow image[\text{inverse[\text{NATADD}], odd}]\}] = \text{Reverse}
\]

\[
\text{Out}[27]= \text{subclass[cart[even, odd], image[\text{inverse[\text{NATADD}], odd}] = True}
\]

\[
\text{In}[28]:= \% / \text{/. \text{Equal} \rightarrow \text{SetDelayed}}
\]
Corollaries made into temporary rewrite rules.

\[ \text{In } [29]:= \text{SubstTest}[\text{implies, and}[\text{subclass}[u, v], \text{subclass}[v, w]], \text{subclass}[u, w], \\
\text{\{u \rightarrow \text{cart}[\text{odd, even}], v \rightarrow \text{image}[\text{inverse}[\text{NATADD}], \text{odd}], \\
\text{w \rightarrow \text{complement}[\text{image}[\text{inverse}[\text{NATADD}], \text{even}]]}\}] \\
\text{Out}[29]= \text{equal}[0, \text{intersection}[\text{even, image}[\text{image}[\text{inverse}[\text{NATADD}], \text{even}], \text{odd}]]] = \text{True} \]

\[ \text{In } [30]:= \text{intersection}[\text{even, image}[\text{image}[\text{inverse}[\text{NATADD}], \text{even}], \text{odd}]] := 0 \]

\[ \text{In } [31]:= \text{SubstTest}[\text{implies, and}[\text{subclass}[u, v], \text{subclass}[v, w]], \text{subclass}[u, w], \\
\text{\{u \rightarrow \text{cart}[\text{even, odd}], v \rightarrow \text{image}[\text{inverse}[\text{NATADD}], \text{odd}], \\
\text{w \rightarrow \text{complement}[\text{image}[\text{inverse}[\text{NATADD}], \text{even}]]}\}] \\
\text{Out}[31]= \text{equal}[0, \text{intersection}[\text{odd, image}[\text{image}[\text{inverse}[\text{NATADD}], \text{even}], \text{even}]]] = \text{True} \]

\[ \text{In } [32]:= \text{intersection}[\text{odd, image}[\text{image}[\text{inverse}[\text{NATADD}], \text{even}], \text{even}]] := 0 \]

---

**domain and range lemmas for image[\text{inverse}[\text{NATADD}], x]**

Domain lemma.

\[ \text{In } [33]:= \text{Map}[\text{or}[#1, \text{subclass}[\text{domain}[\text{image}[\text{inverse}[\text{NATADD}], x]], \text{omega}]], \text{omega}] \text{ \&}, \text{SubstTest}[\text{subclass}, \text{image}[\text{inverse}[z], x], \text{domain}[z], z \rightarrow \text{NATADD}] \]

\[ \text{Out}[33]= \text{subclass}[\text{domain}[\text{image}[\text{inverse}[\text{NATADD}], x]], \text{omega}] = \text{True} \]

\[ \text{In } [34]= \text{subclass}[\text{domain}[\text{image}[\text{inverse}[\text{NATADD}], x_\_]], \text{omega}] = \text{True} \]

Range lemma.

\[ \text{In } [35]:= \text{SubstTest}[\text{subclass, image}[\text{inverse}[z], x], \text{domain}[z], z \rightarrow \text{NATADD}] \]

\[ \text{Out}[35]= \text{subclass}[\text{range}[\text{image}[\text{inverse}[\text{NATADD}], x]], \text{omega}] = \text{True} \]

\[ \text{In } [36]:= \text{subclass}[\text{range}[\text{image}[\text{inverse}[\text{NATADD}], x_\_]], \text{omega}] = \text{True} \]

---

**inverse image rules**

Lemma.

\[ \text{In } [37]:= \text{SubstTest}[\text{subclass, image}[\text{inverse}[\text{NATADD}], \text{even}], \text{intersection}[u, v, w], \\
\text{\{u \rightarrow \text{cart}[\text{omega, omega}], \\
v \rightarrow \text{complement}[, \text{cart}[\text{even, odd}], w \rightarrow \text{complement}[\text{cart}[\text{odd, even}]]\}] \\
\text{Out}[37]= \text{subclass}[\text{image}[\text{inverse}[\text{NATADD}], \text{even}], \text{union}[, \text{cart}[\text{even, even}], \text{cart}[\text{odd, odd}]]] = \text{True} \]

\[ \text{In } [38]= \% / . \text{Equal} \rightarrow \text{SetDelayed} \]
Theorem.

\>

\% In [39]:= SubstTest[and, subclass[u, v], subclass[v, u],
\% {u -> image[inverse[NATADD], even], v -> union[cart[even, even], cart[odd, odd]]}]

\% Out[39]= True = equal[image[inverse[NATADD], even], union[cart[even, even], cart[odd, odd]]]

\% In[40]:= image[inverse[NATADD], even] := union[cart[even, even], cart[odd, odd]]

Corollary.

\% In[41]:= SubstTest[intersection, image[inverse[NATADD], u],
\% image[inverse[NATADD], v], {u -> omega, v -> complement[even]}] // Reverse

\% Out[41]= image[inverse[NATADD], odd] = union[cart[even, odd], cart[odd, even]]

\% In[42]:= image[inverse[NATADD], odd] := union[cart[even, odd], cart[odd, even]]