PLUS is the range of unital submonoid of dir[NATADD, INTADD]

Johan G. F. Belinfante
2011 December 14

In[1]:= **SetDirectory["l:"]; << goedel.11dec13a**

```
:Package Title: goedel.11dec13a            2011 December 13 at 11:30 p.m.

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

It is now:  2011 Dec 14 at 11:44

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 Dec 14 at 11:57
```

summary

The function **PLUS** is the range of a unital submonoid of the direct product of **NATADD** and **INTADD**. This fact is derived by using the fact that **PLUS** is a binary homomorphism from **NATADD** to **INTADD**.

In[2]:= **member[PLUS, binhom[NATADD, INTADD]]**

Out[2]= True

direct product of NATADD and INTADD

Theorem. The direct product of **NATADD** and **INTADD** is a monoid.

In[3]:= **SubstTest[implies, and[member[u, MONOIDS], member[v, MONOIDS]], member[direct[u, v], MONOIDS], {u → NATADD, v → INTADD}] // Reverse**

Out[3]= member[composite[cross[NATADD, INTADD], TWIST], MONOIDS] == True

In[4]:= **member[composite[cross[NATADD, INTADD], TWIST], MONOIDS] := True**

Some weaker results will also be made available because these may be needed in proofs that make use of very general results.

Corollary.

```
In[5]:=  SubstTest[implies, and[member[u, SEMIGPS], member[v, SEMIGPS]],
           member[direct[u, v], SEMIGPS], {u → NATADD, v → INTADD}] // Reverse

Out[5]=  member[composite[cross[NATADD, INTADD], TWIST], SEMIGPS] == True

In[6]:=  member[composite[cross[NATADD, INTADD], TWIST], SEMIGPS] := True
```

Corollary.

```
In[7]:=  SubstTest[implies, and[member[u, BINOPS], member[v, BINOPS]],
           member[direct[u, v], BINOPS], {u → NATADD, v → INTADD}] // Reverse

Out[7]=  member[composite[cross[NATADD, INTADD], TWIST], BINOPS] == True

In[8]:=  member[composite[cross[NATADD, INTADD], TWIST], BINOPS] := True
```

Theorem. The neutral element of **direct[NATADD, INTADD]**.

```
In[9]:=  SubstTest[e, direct[binop[x], binop[y]], {x → NATADD, y → INTADD}] // Reverse

Out[9]=  e[composite[cross[NATADD, INTADD], TWIST]] == PAIR[0, id[omega]]

In[10]:=  e[composite[cross[NATADD, INTADD], TWIST]] := PAIR[0, id[omega]]
```

PLUS as the range of a monoid

Theorem. Any binary homomorphism from **NATADD** to **INTADD** is binary closed under **direct[NATADD, INTADD]**.

```
In[15]:=  SubstTest[subclass, binhom[funpart[x], y],
            binclosed[direct[funpart[x], y]], {x → NATADD, y → INTADD}] // Reverse

Out[15]=  subclass[binhom[NATADD, INTADD],
            binclosed[composite[cross[NATADD, INTADD], TWIST]]] == True

In[16]:=  subclass[binhom[NATADD, INTADD],
            binclosed[composite[cross[NATADD, INTADD], TWIST]]] := True
```

Lemma.

```
In[17]:=  SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w], {u → PLUS,
             v → binhom[NATADD, INTADD], w → binclosed[direct[NATADD, INTADD]]}] // Reverse

Out[17]=  subclass[composite[COMPOSE, cross[PLUS, PLUS], inverse[NATADD]], PLUS] == True

In[18]:=  % /. Equal → SetDelayed
```

Theorem. The function **PLUS** is binary closed under **direct[NATADD, INTADD]**.

```
In[19]:= SubstTest[implies, and[subclass[u, v], FUNCTION[v]],
           equal[u, composite[v, id[domain[u]]]]],
           {u -> composite[COMPOSE, cross[PLUS, PLUS], inverse[NATADD]], v → PLUS}] // Reverse

Out[19]= equal[PLUS, composite[COMPOSE, cross[PLUS, PLUS], inverse[NATADD]]] == True

In[20]:= composite[COMPOSE, cross[PLUS, PLUS], inverse[NATADD]] := PLUS
```

The function **PLUS** is the range of a submonoid of **direct[NATADD, INTADD]**.

Corollary. The restriction of **direct[NATADD, INTADD]** to the cartesian square of **PLUS** is a monoid.

```
In[21]:= SubstTest[or, member[composite[x, id[cart[y, y]]], MONOIDS], not[member[x, MONOIDS]],
           not[member[e[x], y]], not[subclass[image[x, cart[y, y]], y]],
           {x → direct[NATADD, INTADD], y → PLUS}] // Reverse

Out[21]= member[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]], MONOIDS] == True

In[22]:= member[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]], MONOIDS] := True
```

Corollary. The function **PLUS** is the range of a submonoid of **direct[NATADD, INTADD]**.

```
In[26]:= SubstTest[implies, member[x, y], member[range[x], image[IMAGE[SECOND], y]],
           {x → composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]],
            y → intersection[MONOIDS, P[direct[NATADD, INTADD]]]}] // Reverse

Out[26]= member[PLUS, image[IMAGE[SECOND],
            intersection[MONOIDS, P[composite[cross[NATADD, INTADD], TWIST]]]]] == True

In[27]:= member[PLUS, image[IMAGE[SECOND],
            intersection[MONOIDS, P[composite[cross[NATADD, INTADD], TWIST]]]]] := True
```

Lemma. The ordered pair of **0** and **id[ω]** is a neutral element of restriction of **direct[NATADD, INTADD]** to the cartesian square of **PLUS**.

```
In[28]:= member[pair[0, id[omega]],
           ids[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]]]] // AssertTest

Out[28]= member[pair[0, id[omega]],
           ids[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]]]] == True

In[29]:= % /. Equal → SetDelayed
```

Theorem. A formula for the class of identities of this submonoid.

```
In[30]:= SubstTest[implies, and[member[t, ids[x]], member[x, MONOIDS]],
           equal[ids[x], set[t]], {t -> pair[0, id[omega]],
            x -> composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]]}] // Reverse

Out[30]= equal[cart[set[0], set[id[omega]]],
           ids[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]]]] == True
```

```
In[31]:= ids[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]]] :=
          cart[set[0], set[id[omega]]]
```

The submonoid of **direct[NATADD, INTADD]** with range **PLUS** is unital.

Theorem. The neutral element of the submonoid of **direct[NATADD, INTADD]** with range **PLUS** is **PAIR[0, id[ω]]**.

```
In[32]:= SubstTest[A, ids[t], t -> composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]]]

Out[32]= e[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]]] == PAIR[0, id[omega]]

In[33]:= e[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]]] := PAIR[0, id[omega]]
```

Theorem. A mapping statement about the submonoid.

```
In[38]:= member[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]],
           map[cart[PLUS, PLUS], PLUS]] // AssertTest

Out[38]= member[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]],
           map[cart[PLUS, PLUS], PLUS]] == True

In[39]:= member[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]],
           map[cart[PLUS, PLUS], PLUS]] := True
```

Comment. The monoid composition law is given by the following:

```
In[34]:= APPLY[composite[cross[NATADD, INTADD], TWIST, id[cart[PLUS, PLUS]]],
          PAIR[PAIR[nat[x], plus[nat[x]]], PAIR[nat[y], plus[nat[y]]]]]

Out[34]= PAIR[natadd[nat[x], nat[y]], plus[natadd[nat[x], nat[y]]]]
```