binary closed under subtraction

Johan G. F. Belinfante
2011 November 12

```
In[1]:= SetDirectory["l:"]; << goedel.11nov11a

        :Package Title: goedel.11nov11a              2011 November 11 at 8:40 p.m.

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

        It is now:  2011 Nov 12 at 20:34

        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 Nov 12 at 20:47
```

summary

Subtraction is rotated addition. In this notebook, the condition of being binary closed under subtraction for sets of integers is related to the corresponding condition for natural numbers. For integers, subtraction is always possible. A subset $\mathbf{x \subset Z}$ is said to be **binary closed under subtraction** if $\mathbf{u \in x}$ and $\mathbf{v \in x}$ implies $\mathbf{u - v \in x}$. For natural numbers, subtraction is possible only when $\mathbf{v \leq u}$. Because of this restriction, a subset $\mathbf{x \subset \omega}$ is said to be **binary closed under subtraction** if $\mathbf{u \in x}$ and $\mathbf{v \in x}$ implies $\mathbf{u - v \in x}$ whenever $\mathbf{v \leq u}$.

introduction

Subtraction for integers is the binary operation **rotate[INTADD]**.

```
In[2]:= member[rotate[INTADD], BINOPS]

Out[2]= True
```

Theorem. For natural numbers, subtraction is the partial binary operation **rotate[NATADD]**.

```
In[3]:= member[rotate[NATADD],
          map[composite[id[omega], inverse[S], id[omega]], omega]] // AssertTest

Out[3]= member[rotate[NATADD], map[composite[id[omega], inverse[S], id[omega]], omega]] == True

In[4]:= member[rotate[NATADD], map[composite[id[omega], inverse[S], id[omega]], omega]] := True
```

These two functions are related as follows:

```
In[5]:= composite[inverse[PLUS], rotate[INTADD], cross[PLUS, PLUS]]

Out[5]= rotate[NATADD]
```

derivation

Lemma.

```
In[6]:= SubstTest[member, image[inverse[PLUS], x],
         binclosed[composite[inverse[PLUS], t, cross[PLUS, PLUS]]], t → rotate[INTADD]]

Out[6]= subclass[intersection[image[image[inverse[INTADD], intersection[x, range[PLUS]]],
            intersection[x, range[PLUS]]], range[PLUS]], x] == subclass[image[PLUS,
           image[image[inverse[NATADD], image[inverse[PLUS], x]], image[inverse[PLUS], x]]], x]

In[7]:= subclass[intersection[image[image[inverse[INTADD], intersection[x_, range[PLUS]]],
            intersection[x_, range[PLUS]]], range[PLUS]], x_] := subclass[image[PLUS,
           image[image[inverse[NATADD], image[inverse[PLUS], x]], image[inverse[PLUS], x]]], x]
```

Lemma.

```
In[8]:= SubstTest[implies, member[x, binclosed[t]], member[image[inverse[PLUS], x], binclosed[
            composite[inverse[PLUS], t, cross[PLUS, PLUS]]]], t → rotate[INTADD]] // Reverse

Out[8]= or[not[member[x, V]], not[subclass[image[image[inverse[INTADD], x], x], x]],
          subclass[image[PLUS, image[image[inverse[NATADD], image[inverse[PLUS], x]],
            image[inverse[PLUS], x]]], x]] == True

In[9]:= (% /. x → x_) /. Equal → SetDelayed
```

Main Theorem. A connection between the class **binclosed[rotate[INTADD]]** of sets of integers that are binary closed under integer subtraction and the class **binclosed[rotate[NATADD]]** of sets of natural numbers that are binary closed under subtraction of natural numbers.

```
In[10]:= Map[equal[V, domain[#]] &,
          SubstTest[reify, x, case[implies[member[x, u], member[image[inverse[PLUS], x], v]]],
           {u → binclosed[rotate[INTADD]], v -> binclosed[rotate[NATADD]]}]]

Out[10]= subclass[image[IMAGE[inverse[PLUS]],
            binclosed[composite[INTADD, cross[Id, INVERSE]]]], binclosed[rotate[NATADD]]] == True

In[11]:= subclass[
           image[IMAGE[inverse[PLUS]], binclosed[composite[INTADD, cross[Id, INVERSE]]]],
           binclosed[rotate[NATADD]]] := True
```

Comment. The following rewrite rule for integer subtraction amounts to the observation **u – v = u + (- v)**.

```
In[12]:= rotate[INTADD]

Out[12]= composite[INTADD, cross[Id, INVERSE]]
```

comments

Subtraction for natural numbers is not a binary operation because it fails to be defined for all pairs of natural numbers.

```
In[13]:= member[rotate[NATADD], BINOPS]

Out[13]= False
```

The one-to-one function **PLUS** is a binary homomorphism from **NATADD** to **INTADD**.

```
In[14]:= member[PLUS, binhom[NATADD, INTADD]]

Out[14]= True
```

Lemma.

```
In[15]:= Map[not, SubstTest[implies, equal[u, v], equal[domain[u], domain[v]],
            {u -> composite[INTADD, cross[PLUS, composite[INVERSE, PLUS]]],
             v -> composite[PLUS, rotate[NATADD]]}] // Reverse]

Out[15]= equal[composite[INTADD, cross[PLUS, composite[INVERSE, PLUS]]],
           composite[PLUS, rotate[NATADD]]] == False

In[16]:= % /. Equal→ SetDelayed
```

Theorem. The function **PLUS** is not a binary homomorphism from **rotate[NATADD]** to **rotate[INTADD]**.

```
In[17]:= Map[not, SubstTest[implies, member[t, binhom[x, y]],
            equal[composite[t, x], composite[y, cross[t, t]]],
            {t -> PLUS, x -> rotate[NATADD], y -> rotate[INTADD]}] // Reverse]

Out[17]= member[PLUS, binhom[rotate[NATADD], composite[INTADD, cross[Id, INVERSE]]]] == False

In[18]:= member[PLUS, binhom[rotate[NATADD], composite[INTADD, cross[Id, INVERSE]]]] := False
```