binary homs from integer to rational addition, part 1. basics

Johan G. F. Belinfante
2012 December 11

```
In[1]:= SetDirectory["l:"]; << goedel.12dec09a

    :Package Title: goedel.12dec09a              2012 December 9 at 11:50 a.m.

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

    It is now:  2012 Dec 11 at 4:16

    Loading Simplification Rules

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

    weightlimit = 40

    Loading completed.

    It is now:  2012 Dec 11 at 4:32
```

summary

Binary homomorphisms from integer to rational addition are mappings from integers to rational numbers. They preserve the additive neutral element and additive inverses.

0 · x = 0

The rational number zero **0** is **e[RATADD] = Z × {id[ω]}**. The equation **0 · x = 0** for rational arithmetic holds if and only if **x** is a rational number.

Lemma.

```
In[2]:= SubstTest[implies, and[equal[u, v], member[u, V]], member[v, V],
          {u -> cart[Z, set[id[omega]]], v -> ratmul[x, cart[Z, set[id[omega]]]]}] // Reverse

Out[2]= or[member[x, RATS],
          not[equal[cart[Z, set[id[omega]]], ratmul[x, cart[Z, set[id[omega]]]]]]] == True

In[3]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[4]:= SubstTest[implies, equal[x, rat[t]],
          equal[cart[Z, set[id[omega]]], ratmul[x, cart[Z, set[id[omega]]]]], t → x] // Reverse

Out[4]= or[equal[cart[Z, set[id[omega]]], ratmul[x, cart[Z, set[id[omega]]]]],
          not[member[x, RATS]]] == True

In[5]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. A simplification rule.

```
In[6]:= equiv[equal[cart[Z, set[id[omega]]],
          ratmul[x, cart[Z, set[id[omega]]]]], member[x, RATS]]

Out[6]= True

In[7]:= equal[cart[Z, set[id[omega]]], ratmul[x_, cart[Z, set[id[omega]]]]] := member[x, RATS]
```

basics

Theorem. Any binary hom from **INTADD** to **RATADD** is a mapping from integers to rationals.

```
In[8]:= SubstTest[implies, member[x, binhom[u, v]],
          member[x, map[fix[domain[u]], fix[domain[v]]]], {u → INTADD, v → RATADD}] // Reverse

Out[8]= or[member[x, map[Z, RATS]], not[member[x, binhom[INTADD, RATADD]]]] == True

In[9]:= or[member[x_, map[Z, RATS]], not[member[x_, binhom[INTADD, RATADD]]]] := True
```

Corollary.

```
In[10]:= Map[equal[V, #] &, dif[binhom[INTADD, RATADD], map[Z, RATS]] // complement // Normality]

Out[10]= subclass[binhom[INTADD, RATADD], map[Z, RATS]] == True

In[11]:= subclass[binhom[INTADD, RATADD], map[Z, RATS]] := True
```

Theorem. If **x** is a binary hom from **INTADD** to **RATADD** and **y** is an integer, then **x(y)** is a rational number.

```
In[12]:= Map[not, SubstTest[and, implies[p1, p3],
            implies[and[p2, p3], p4], not[implies[and[p1, p2], p4]],
            {p1 -> member[x, binhom[INTADD, RATADD]], p2 → member[y, Z],
             p3 → member[x, map[Z, RATS]], p4 → member[APPLY[x, y], RATS]}]] // Reverse

Out[12]= or[member[APPLY[x, y], RATS],
           not[member[x, binhom[INTADD, RATADD]]], not[member[y, Z]]] == True

In[13]:= or[member[APPLY[x_, y_], RATS],
           not[member[x_, binhom[INTADD, RATADD]]], not[member[y_, Z]]] := True
```

Corollary. If **x** is a binary hom from **INTADD** to **RATADD**, then **x(1)** is a rational number.

```
In[14]:= SubstTest[implies, and[member[x, binhom[INTADD, RATADD]], member[y, Z]],
           member[APPLY[x, y], RATS], y → plus[set[0]]] // Reverse

Out[14]= or[member[APPLY[x, composite[id[omega], SUCC]], RATS],
           not[member[x, binhom[INTADD, RATADD]]]] == True

In[15]:= or[member[APPLY[x_, composite[id[omega], SUCC]], RATS],
           not[member[x_, binhom[INTADD, RATADD]]]] := True
```

Theorem. Binary homomorphisms from **INTADD** to **RATADD** preserve the neutral element: the integer zero is mapped to the rational zero.

```
In[16]:= SubstTest[implies, and[member[t, binhom[u, v]], member[u, GROUPS], member[v, GROUPS]],
           equal[APPLY[t, e[u]], e[v]], {u → INTADD, v → RATADD}] // Reverse

Out[16]= or[equal[APPLY[t, id[omega]], cart[Z, set[id[omega]]]],
           not[member[t, binhom[INTADD, RATADD]]]] == True

In[17]:= or[equal[APPLY[t_, id[omega]], cart[Z, set[id[omega]]]],
           not[member[t_, binhom[INTADD, RATADD]]]] := True
```

preservation of additive inverses

If **x** is a binary homomorphism from integer to rational addition, then **inv[RATADD] ∘ x = x ∘ inv[INTADD]**.

Lemma.

```
In[18]:= SubstTest[implies, and[member[x, binhom[y, z]], member[y, GROUPS], member[z, GROUPS]],
           equal[composite[x, inv[y]], composite[inv[z], x]], {y → INTADD, z → RATADD}] // Reverse

Out[18]= or[equal[composite[inv[RATADD], x], composite[x, id[Z], INVERSE]],
           not[member[x, binhom[INTADD, RATADD]]]] == True

In[19]:= (% /. x → x_) /. Equal → SetDelayed
```

The factor of **id[Z]** can be eliminated.

Theorem.

```
In[20]:= Map[not, SubstTest[and, implies[p1, p2],
            implies[p1, p3], implies[and[p2, p3], p4], not[implies[p1, p4]],
            {p1 -> member[x, binhom[INTADD, RATADD]], p2 -> equal[domain[x], Z],
             p3 → equal[composite[inv[RATADD], x], composite[x, id[Z], INVERSE]],
             p4 -> equal[composite[inv[RATADD], x], composite[x, INVERSE]]}]] // Reverse

Out[20]= or[equal[composite[x, INVERSE], composite[inv[RATADD], x]],
           not[member[x, binhom[INTADD, RATADD]]]] == True

In[21]:= or[equal[composite[x_, INVERSE], composite[inv[RATADD], x_]],
           not[member[x_, binhom[INTADD, RATADD]]]] := True
```

In particular, this holds for the binary homomorphism **inv[RATADD] ∘ rattimes[rat[x]]**.

Theorem. A simplification rule.

```
In[22]:= Assoc[rattimes[rat[x]], inv[RATADD], INTTIMES] // Reverse

Out[22]= composite[inv[RATADD], rattimes[rat[x]], INTTIMES] ==
          composite[rattimes[rat[x]], INTTIMES, INVERSE]

In[23]:= composite[inv[RATADD], rattimes[rat[x_]], INTTIMES] :=
          composite[rattimes[rat[x]], INTTIMES, INVERSE]
```

Theorem. (Eliminating the **rat** wrapper.)

```
In[24]:= SubstTest[implies, member[t, binhom[INTADD, RATADD]],
           equal[composite[t, INVERSE], composite[inv[RATADD], t]],
           t -> composite[rattimes[x], INTTIMES]] // Reverse

Out[24]= or[equal[composite[inv[RATADD], rattimes[x], INTTIMES],
            composite[rattimes[x], INTTIMES, INVERSE]], not[member[x, RATS]]] == True

In[25]:= or[equal[composite[inv[RATADD], rattimes[x_], INTTIMES],
            composite[rattimes[x_], INTTIMES, INVERSE]], not[member[x_, RATS]]] := True
```

rattimes[x] ∘ INTTIMES ∘ INVERSE

A particular binary hom from integer to rational addition is considered in this section.

Lemma.

```
In[26]:= SubstTest[or, member[composite[t, INVERSE], binhom[INTADD, RATADD]],
           not[member[t, binhom[INTADD, RATADD]]],
           t -> composite[rattimes[x], INTTIMES]] // Reverse

Out[26]= or[member[composite[rattimes[x], INTTIMES, INVERSE], binhom[INTADD, RATADD]],
           not[member[x, RATS]]] == True

In[27]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[28]:= SubstTest[implies, member[t, binhom[INTADD, RATADD]],
           equal[APPLY[t, id[omega]], cart[Z, set[id[omega]]]],
           t -> composite[rattimes[x], INTTIMES, INVERSE]] // Reverse

Out[28]= or[member[x, RATS], not[
            member[composite[rattimes[x], INTTIMES, INVERSE], binhom[INTADD, RATADD]]]] == True

In[29]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. The function **rattimes[x] ∘ INTTIMES ∘ INVERSE** is a binary homomorphism from **INTADD** to **RATADD** if and only if **x** is a rational number.

```
In[30]:= equiv[member[composite[rattimes[x], INTTIMES, INVERSE], binhom[INTADD, RATADD]],
          member[x, RATS]]

Out[30]= True

In[31]:= member[composite[rattimes[x_], INTTIMES, INVERSE], binhom[INTADD, RATADD]] :=
          member[x, RATS]
```