

characteizing RATADD endomorphisms

Johan G. F. Belinfante
2013 March 10

```
In[1]:= SetDirectory["1:"]; << goedel.13mar09a
      :Package Title: goedel.13mar09a          2013 March 10 at 10:15 p.m.
      Loading takes about sixteen minutes, half that time due to builtin pauses.
      It is now: 2013 Mar 10 at 22:16
      Loading Simplification Rules
      TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3
      weightlimit = 40
      Loading completed.
      It is now: 2013 Mar 10 at 22:33
```

summary

Any endomorphism f of rational addition is the operation of multiplying by the rational number $\text{APPLY}[f, \text{id}[Z]]$.

```
In[2]:= implies[member[f, binhom[RATADD, RATADD]], equal[f, rattimes[APPLY[f, id[Z]]]]]
Out[2]= True
```

In this notebook a succinct variable-free reformulation of this fact is derived.

derivation

Lemma. (Inclusion in easy direction.)

```
In[5]:= SubstTest[subclass, fix[composite[u, v]],
      range[u], {u → RATTIMES, v → eval[id[Z]]}] // Reverse
Out[5]= subclass[fix[composite[RATTIMES, eval[id[Z]]]], binhom[RATADD, RATADD]] = True
In[6]:= % /. Equal → SetDelayed
```

Lemma.

```
In[7]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
  {u -> binhom[RATADD, RATADD], v -> map[RATS, RATS], w -> domain[eval[id[Z]]]} // Reverse
```

```
Out[7]= subclass[binhom[RATADD, RATADD], domain[eval[id[Z]]]] == True
```

```
In[8]:= % /. Equal -> SetDelayed
```

Lemma.

```
In[17]:= Assoc[eval[id[Z]], RATTIMES, inverse[RATTIMES]]
```

```
Out[17]= composite[eval[id[Z]], id[binhom[RATADD, RATADD]]] == inverse[RATTIMES]
```

```
In[18]:= composite[eval[id[Z]], id[binhom[RATADD, RATADD]]] := inverse[RATTIMES]
```

Corollary.

```
In[19]:= composite[id[binhom[RATADD, RATADD]], inverse[eval[id[Z]]]] // DoubleInverse
```

```
Out[19]= composite[id[binhom[RATADD, RATADD]], inverse[eval[id[Z]]]] == RATTIMES
```

```
In[20]:= composite[id[binhom[RATADD, RATADD]], inverse[eval[id[Z]]]] := RATTIMES
```

Lemma.

```
In[22]:= SubstTest[implies, subclass[u, v], subclass[composite[t, u], composite[t, v]],
  {t -> id[binhom[RATADD, RATADD]], u -> id[binhom[RATADD, RATADD]],
  v -> composite[inverse[eval[id[Z]]], eval[id[Z]]]} // Reverse
```

```
Out[22]= subclass[binhom[RATADD, RATADD], fix[composite[RATTIMES, eval[id[Z]]]]] == True
```

```
In[23]:= % /. Equal -> SetDelayed
```

Theorem.

```
In[24]:= SubstTest[and, subclass[u, v], subclass[v, u],
  {u -> fix[composite[RATTIMES, eval[id[Z]]], v -> binhom[RATADD, RATADD]}]
```

```
Out[24]= equal[binhom[RATADD, RATADD], fix[composite[RATTIMES, eval[id[Z]]]]] == True
```

```
In[26]:= fix[composite[RATTIMES, eval[id[Z]]]] := binhom[RATADD, RATADD]
```