fractional representations of rationals

Johan G. F. Belinfante
2012 August 29

\[ \text{In[1]} := \text{SetDirectory["l:"; } \text{<< goedel.12aug28a} \]

:Package Title: goedel.12aug28a 2012 August 28 at 3:25 p.m.
Loading takes about sixteen minutes, half that time due to builtin pauses.
It is now: 2012 Aug 29 at 14:3
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 Aug 29 at 14:18

\[ \text{summary} \]

A rational number can be written in infinitely many ways as a fraction. In the GOEDEL program, rational numbers are straight lines through the origin in the integer plane \( \mathbb{Z} \times \mathbb{Z} \). The rational number represents the slope of the corresponding line. Each point \((u, v)\) on the graph of a rational number other than the origin determines a representation of the rational number as a fraction with \(u\) as denominator and \(v\) as numerator.

\[ \text{fractions} \]

The following temporary definition for the fraction \(u \text{\frac{\text{\_}}{\text{\_}}} v\) with denominator \(u\) and numerator \(v\) is used in this notebook.

\[ \text{In[2]} := \text{frac[u\_, v\_] := composite[inverse[inttimes[u]], inttimes[v]]} \]

Each pair of integers determines a fraction. The point \((u, v)\) is a member of the fraction \(u \text{\frac{\text{\_}}{\text{\_}}} v\).

\[ \text{In[3]} := \text{member[pair[u, v], frac[u, v]]} \]

\[ \text{Out[3]} = \text{and[member[u, Z], member[v, Z]]} \]

This fraction is not necessarily a rational number, however, unless its numerator is an integer and the denominator is a non-zero integer. The integer zero is \(\text{id[0]}\).
In[4]:= member[frac[u, v], RATS]
Out[4]= and[member[u, Z], member[v, Z], not[equal[u, id[omega]]]]

introduction

It is often convenient to use the rat wrapper for rational numbers to avoid literals of the form \( x \in \text{RATS} \). Most of the important results in this notebook will be stated twice, with and without the use of the wrapper.

Every rational number is a function. If \((u, v) \in \text{rat}[x]\), then \( v \) is the value of \( \text{rat}[x] \) at \( u \).

In[5]:= implies[member[pair[u, v], rat[x]], equal[v, APPLY[rat[x], u]]]
Out[5]= True

Theorem. (Restatement without the rat wrapper.)

In[7]:= SubstTest[implies, equal[x, rat[t]],
  or[equal[v, APPLY[x, u]], not[member[pair[u, v], x]], t \to x] // Reverse
Out[7]= or[equal[v, APPLY[x, u]], not[member[x, RATS]], not[member[pair[u, v], x]]] := True

In[8]:= or[equal[v_, APPLY[x_, u_]],
  not[member[x_, RATS]], not[member[pair[u_, v_, x_]]] := True

It is sometimes convenient to denote an ordered pair \((u, v)\) by a single variable \( w \), with \( u = \text{first}[w] \) and \( v = \text{second}[w] \). The above statement can then be rewritten as follows.

Theorem. For any point on the graph of a rational number regarded as a straight line function in the integer place, the second coordinate of the point is the value of the rational number at the first coordinate.

In[9]:= SubstTest[implies, member[w, funpart[t]],
  equal[APPLY[funpart[t], first[w]], second[w]], t \to rat[x]] // Reverse
Out[9]= or[equal[APPLY[rat[x], first[w]], second[w]], not[member[w, rat[x]]]] := True

In[10]:= or[equal[APPLY[rat[x_], first[w_]], second[w_]], not[member[w_, rat[x_]]]] := True

Corollary. (Restatement without the rat wrapper.)

In[11]:= SubstTest[implies, equal[x, rat[t]],
  or[equal[second[w], APPLY[x, first[w]]], not[member[w, x]], t \to x] // Reverse
Out[11]= or[equal[APPLY[x, first[w]], second[w]], not[member[w, x]], not[member[x, RATS]]] := True

In[12]:= or[equal[APPLY[x_, first[w_]], second[w_]],
  not[member[w_, x_]], not[member[x_, RATS]]] := True
fractional representations of rationals

The function \texttt{RATIO} connects rational numbers with integer fractions. The domain of the function \texttt{RATIO} is \((\mathbb{Z} - \{\text{id}(\omega)\}) \times \mathbb{Z}\).

\begin{verbatim}
In[14]:= domain[RATIO]
\end{verbatim}

The function \texttt{RATIO} takes each point of the integer plane other than the vertical axis to the corresponding fraction.

\begin{verbatim}
In[15]:= implies[member[w, domain[RATIO]], equal[APPLY[RATIO, w], frac[first[w], second[w]]]]
\end{verbatim}

The basic connection between fractions and rational numbers is currently available in the following form.

\begin{verbatim}
In[16]:= implies[and[member[w, rat[x]], not[equal[first[w], id[omega]]]],
    equal[APPLY[RATIO, w], rat[x]]]
Out[16]= True
\end{verbatim}

In this section, several equivalent statements are derived that do not explicitly involve the function \texttt{RATIO}. The \texttt{APPLY} rule for \texttt{RATIO} involves \texttt{case}.

\begin{verbatim}
In[17]:= APPLY[RATIO, pair[u, v]]
Out[17]= union[case[or[equal[u, id[omega]], not[member[u, 2]]], not[member[v, Z]]],
    composite[inverse[inttimes[u]], inttimes[v]]]
\end{verbatim}

A theorem will be needed to deal with the \texttt{case} construct.

Lemma. Temporary simplification rule.

\begin{verbatim}
In[18]:= equiv[or[and[p, equal[V, x]], and[equal[V, x], equal[V, y]],
    and[equal[x, y], not[p]]],
    or[and[p, equal[V, x]], and[equal[x, y], not[p]]]] // not // not
Out[18]= True
\end{verbatim}

\begin{verbatim}
In[19]:= or[and[p_, equal[V, x_]], and[equal[V, x_], equal[V, y_]],
    and[equal[x_, y_], not[p_]]] := or[and[p, equal[V, x]], and[equal[x, y], not[p]]]
\end{verbatim}

Theorem. An equation involving \texttt{case}.

\begin{verbatim}
In[20]:= equal[x, union[y, case[p]]] // AssertTest
Out[20]= equal[x, union[y, case[p]]] = or[and[p, equal[V, x]], and[equal[x, y], not[p]]]
In[21]:= equal[x_, union[y_, case[p_]]] := or[and[p, equal[V, x]], and[equal[x, y], not[p]]]
\end{verbatim}
The following theorem connects rational numbers and fractions directly, eliminating the function \texttt{RATIO} altogether.

**Theorem.** If \((u, v) \in \texttt{rat}[x]\), and \(u\) is not the integer zero, then \(\texttt{rat}[x] = u \setminus v\).

\[
\begin{align*}
\text{In}[22]:= & \quad \text{SubstTest[imply, and[member[w, \texttt{rat}[x]], not[equal[first[w], id[\texttt{omega}]]]],}
\quad \text{equal[\text{APPLY}[\text{RATIO}, w], \texttt{rat}[x]], w \rightarrow \text{pair}[u, v]} \big/ \text{Reverse} \big/ \text{MapNotNot} \\
\text{Out}[22]= & \quad \text{or[equal[u, id[\texttt{omega}]], equal[composite[inverse[inttimes[u]], inttimes[v]], \texttt{rat}[x]],}
\quad \text{not[member[pair[u, v], \texttt{rat}[x]]]] = True}
\end{align*}
\]

**Corollary.** (Eliminate the \texttt{rat} wrapper.)

\[
\begin{align*}
\text{In}[23]:= & \quad \text{SubstTest[imply, equal[x, \texttt{rat}[t]],}
\quad \text{or[equal[u, id[\texttt{omega}]], equal[composite[inverse[inttimes[u]], inttimes[v]], x],}
\quad \text{not[member[pair[u, v], x]]], t \rightarrow x} \big/ \text{Reverse} \\
\text{Out}[24]= & \quad \text{or[equal[u, id[\texttt{omega}]], equal[x, composite[inverse[inttimes[u]], inttimes[v]]],}
\quad \text{not[member[x, \texttt{RATS}]], not[member[pair[u, v], x]]]] = True}
\end{align*}
\]

\[
\begin{align*}
\text{In}[25]:= & \quad \text{SubstTest[imply, id[\texttt{omega}], u, not[member[pair[u, v], x]], not[member[x, \texttt{RATS}]]] := True}
\end{align*}
\]

The result can also be restated using a single variable for the ordered pair.

**Lemma.** Simplification rule.

\[
\begin{align*}
\text{In}[26]:= & \quad \text{SubstTest[member, w, composite[Id, t], t \rightarrow \texttt{rat}[x]]} \\
\text{Out}[26]= & \quad \text{and[member[w, \texttt{rat}[x]], member[first[w], V]] = member[w, \texttt{rat}[x]]}
\end{align*}
\]

\[
\begin{align*}
\text{In}[27]:= & \quad \text{and[member[w, \texttt{rat}[x]], member[first[w], V]] := member[w, \texttt{rat}[x]]}
\end{align*}
\]

**Theorem.** If \(w \in \texttt{rat}[x]\) and if \(\text{first}[w]\) is not zero, then \(\texttt{rat}[x] = \text{first}[w] \setminus \text{second}[w]\).

\[
\begin{align*}
\text{In}[28]:= & \quad \text{SubstTest[imply, and[member[w, \texttt{rat}[x]], not[equal[first[w], id[\texttt{omega}]]]],}
\quad \text{equal[\text{APPLY}[\text{RATIO}, w], \texttt{rat}[x]], w \rightarrow \text{PAIR}[\text{first}[w], \text{second}[w]]} \big/ \text{Reverse} \big/ \text{MapNotNot} \\
\text{Out}[28]= & \quad \text{or[composite[inverse[inttimes[first[w]], inttimes[second[w]]], \texttt{rat}[x]],}
\quad \text{equal[first[w], id[\texttt{omega}]], not[member[w, \texttt{rat}[x]]]] = True}
\end{align*}
\]

\[
\begin{align*}
\text{In}[29]:= & \quad \text{SubstTest[imply, and[member[w, \texttt{rat}[x]], not[equal[first[w], id[\texttt{omega}]]]],}
\quad \text{equal[\text{APPLY}[\text{RATIO}, w], \texttt{rat}[x]], w \rightarrow \text{PAIR}[\text{first}[w], \text{second}[w]]} \big/ \text{Reverse} \big/ \text{MapNotNot} \\
\text{Out}[29]= & \quad \text{or[composite[inverse[inttimes[first[w]], inttimes[second[w]]], \texttt{rat}[x]],}
\quad \text{equal[first[w], id[\texttt{omega}]], not[member[w, \texttt{rat}[x]]]] = True}
\end{align*}
\]

**Corollary.** (Eliminating the \texttt{rat} wrapper.)
Given a member of the domain of a function, one can obtain a point on the function itself using \texttt{APPLY}.

**Theorem.** Points on a rational number \texttt{rat[x]} are determined by their first coordinates.

```math
\texttt{SubstTest}[\texttt{implies}, \texttt{equal[x, rat[t]]},
  \texttt{or[equal[composite[\texttt{inverse[\texttt{inttimes[first[w]]}, \texttt{inttimes[second[w]]}], x}],
    \texttt{equal[first[w], id[omega]}, \texttt{not[member[w, x]]}], t \rightarrow x] // Reverse}
\texttt{Out[30]} = \texttt{or[equal[x, composite[\texttt{inverse[\texttt{inttimes[first[w]]}, \texttt{inttimes[second[w]]}]},
    \texttt{equal[first[w], id[omega]}, \texttt{not[member[w, x]]}], \not[member[x, \texttt{RATS}]]}] = \texttt{True}}
```

```math
\texttt{SubstTest}[\texttt{member}, \texttt{pair[u, APPLY[funpart[t], u]]}, \texttt{funpart[t], t \rightarrow rat[x]] // Reverse}
\texttt{Out[32]} = \texttt{member[pair[u, APPLY[\texttt{rat[x]}, u]], \texttt{rat[x]}] = member[u, domain[\texttt{rat[x]}]]}
```

```math
\texttt{SubstTest}[\texttt{implies}, \texttt{and[member[pair[u, v], \texttt{rat[x]}], \not[equal[u, id[omega]]]]},
  \texttt{equal[\texttt{rat[x]}, \texttt{frac[u, v]}, v \rightarrow APPLY[\texttt{rat[x]}, u]] // Reverse}
\texttt{Out[34]} = \texttt{or[equal[u, id[omega]]],
  \texttt{equal[composite[\texttt{inverse[\texttt{inttimes[u]}], \texttt{inttimes[APPLY[\texttt{rat[x]}, u]]}], \texttt{rat[x]}],
    \not[member[u, domain[\texttt{rat[x]}]]}] = \texttt{True}}
```

**Theorem.** If \texttt{u} is a non-zero member of \texttt{domain[rat[x]]}, then \texttt{rat[x] = u \setminus APPLY[\texttt{rat[x]}, u]}.

```math
\texttt{SubstTest}[\texttt{implies}, \texttt{equal[x, rat[t]]}, \texttt{or[equal[u, id[omega]]],
  \texttt{equal[composite[\texttt{inverse[\texttt{inttimes[u]}], \texttt{inttimes[APPLY[x, u]]}], x],
    \not[member[u, domain[x]]]}, t \rightarrow x] // Reverse}
\texttt{Out[36]} = \texttt{or[equal[u, id[omega]]],
  \texttt{equal[x, composite[\texttt{inverse[\texttt{inttimes[u]}], \texttt{inttimes[APPLY[x, u]]}]},
    \not[member[u, domain[x]]], \not[member[x, \texttt{RATS}]]}] = \texttt{True}}
```

**Corollary.** (Restatement without the \texttt{rat} wrapper.)

```math
\texttt{SubstTest}[\texttt{implies}, \texttt{equal[x, rat[t]]}, \texttt{or[equal[u, id[omega]]],
  \texttt{equal[composite[\texttt{inverse[\texttt{inttimes[u]}], \texttt{inttimes[APPLY[x, u]]}], x],
    \not[member[u, domain[x]]]}, t \rightarrow x] // Reverse}
\texttt{Out[38]} = \texttt{or[equal[u, id[omega]]],
  \texttt{equal[x, composite[\texttt{inverse[\texttt{inttimes[u]}], \texttt{inttimes[APPLY[x, u]]}]},
    \not[member[u, domain[x]]], \not[member[x, \texttt{RATS}]]}] = \texttt{True}}
```