I See Your 127.32+, A Tale of Rationals

Xavier Noria
@fxn

EuRuKo 2013
5.00 * 4.33 = 21.65

stake

odds

profit
\[ 2.25 \times 4.33 = 9.74 \text{ (partial match 1)} \]
\[ 2.75 \times 4.33 = 11.90 \text{ (partial match 2)} \]

\[ 5.00 \quad 21.64 \text{ (should be 21.65)} \]
Natural Numbers (\(\mathbb{N}\))
0, 1, 2, 3, ...
\[0 = \{\}\]
\[1 = 0 \cup \{0\} = \{0\}\]
\[2 = 1 \cup \{1\} = \{0, 1\}\]
\[3 = 2 \cup \{2\} = \{0, 1, 2\}\]
\[\vdots\]
\[n + 1 = n \cup \{n\} = \{0, 1, 2, \ldots, n\}\]
Generalization: ordinals and cardinals
Theorem (Cantor): Card(A) < Card(P(A))
n < 2^n
Card(\mathbb{N}) < Card(P(\mathbb{N}))
Natural numbers map to the unsigned integer types of some languages like C.
Ruby does not have unsigned integers
Integer Numbers (Z)
..., -3, -2, -1, 0, 1, 2, 3, ...
Integers are constructed from the naturals:

\[(a, b) \sim (c, d) \iff a + d = b + c\]
$\text{Card}(\mathbb{Z}) = \text{Card}(\mathbb{N})$
1 3 5 7 \text{ } 2n + 1

0 1 2 3 \ldots \text{ } n
Ruby has arbitrary-precision integers
> (1..100).reduce(:*)
   => 9332621544394415268169923885626670049
   071596826438162146859296389521759999932299
   15608941463976156518286253697920827223758
   2511852109168640000000000000000000000000
Fixnums are immediate values in MRI
/* embeds integer in VALUE */

((VALUE)(((SIGNED_VALUE)(i))<<1 | FIXNUM_FLAG))
/* reads integer from VALUE */

(long)RSHIFT(((SIGNED_VALUE)(x)),1)
static VALUE
fix_succ(VALUE num)
{
    long i = FIX2LONG(num) + 1;
    return LONG2NUM(i);
}
/* include/ruby/ruby.h */

static inline VALUE
rb_long2num_inline(long v)
{
    if (FIXABLE(v))
        return LONG2FIX(v);
    else
        return rb_int2big(v);
}

#define LONG2NUM(x) rb_long2num_inline(x)
Bignums have a mixed representation in MRI
/* ruby/ruby.h */

struct RBignum {
    struct RBasic basic;
    union {
        struct {
            long len;
            BDIGIT *digits;
        } heap;
        BDIGIT ary[RBIGNUM_EMBED_LEN_MAX];
    } as;
};
LibTomMath
Rationals are constructed from the integers:

\[(a, b) \sim (p, q) \iff aq = pb\]
Card(\mathbb{Q}) = Card(\mathbb{N})
Ruby has rationals
struct RRational {
    struct RBasic basic;
    VALUE num;
    VALUE den;
};
* Exact arithmetic, including division
* Predictable across technologies
Real Numbers (\( \mathbb{R} \))
Reals are constructed from the rationals:

* Equivalence classes of Cauchy sequences of rationals
* Dedekind cuts
* Categorical characterization
* ...
\[ \text{Card}(\mathbb{R}) = \text{Card}(P(\mathbb{N})) > \text{Card}(\mathbb{N}) \]
Reals that are not rationals are called *irrationals*
\[ \sqrt{2} = \frac{a}{b} \]
\[
\sqrt{2} = \frac{a}{b} \quad \Rightarrow \\
\Rightarrow 2 = \frac{a^2}{b^2}
\]
$$\sqrt{2} = \frac{a}{b} \implies$$

$$\implies 2 = \frac{a^2}{b^2} \implies$$

$$\implies 2b^2 = a^2$$
Math generally speaking studies numbers as abstract entities, irrespective of their representation.
Theorem (Euclid): If $p$ is a prime such that $p|ab$, then $p$ divides $a$, or $p$ divides $b$. 
When is the representation of a rational in a given base finite?
$1.256 = \frac{1256}{10^3}$
$$1.256 = \frac{1256}{10^3} = \frac{8 \times 157}{8 \times 125}$$
\[ 1.256 = \frac{1256}{10^3} = \frac{8 \times 157}{8 \times 125} = \frac{157}{125} \]
A fraction in lowest terms $p/q$ has a finite representation in base $b$ iff $q$ divides $b^n$ for some $n$. 
Cents < 1 with finite representation in base 2:

0.00, 0.25, 0.50, 0.75

The other 96 are periodic.
Irrational numbers have an infinite non-periodic representation in any base.
IEEE 754 double precision (Wrapped by RFloat)

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>52</td>
</tr>
</tbody>
</table>

- sign
- exponent
- mantissa (+1 implicit)
Immediate Flonum (Ruby 2.0, 64-bit platforms)

- **MSB**: 9
- **52**
- **1**
- **2**

**exponent**

**mantissa (+1 implicit)**

**sign**

**mask**
1.2 - 1.1 == 0.1 # => false
require 'bigdecimal'

a = BigDecimal.new('1.2')
b = BigDecimal.new('1.1')
c = BigDecimal.new('0.1')

a - b == c #=> true
Internal representation of a BigDecimal:
* Arbitrary-precision integer as mantissa
* Mantissa uses base $10^N$
* Exponent
* Sign
* Flags
typedef struct {
    VALUE    obj;
    size_t   MaxPrec;
    size_t   Prec;
    SIGNED_VALUE exponent;
    short    sign;
    short    flag;
    BDIGIT   frac[FLEXIBLE_ARRAY_SIZE];
} Real;
BigDecimal caveats
require 'bigdecimal'

one  = BigDecimal.new('1')
three = BigDecimal.new('3')

three*(one/three) == one
# => false
require 'bigdecimal'
require 'bigdecimal/util'
65.1.to_d  # DON'T DO THIS
require 'bigdecimal'
require 'bigdecimal/util'

'65.1'.to_d # GOOD
Undecidable Statements
Is there any cardinal between $\text{Card}(\mathbb{N})$ and $\text{Card}(\mathcal{P}(\mathbb{N})) = \text{Card}(\mathbb{R})$?
Continuum Hypothesis: No
Paul Cohen proved CH to be undecidable in the 60s (provided ZF is consistent)
Thanks

Yukihiro Matsumoto
Pat Shaughnessy
Dirkjan Bussink
Charles Nutter
Koichi Sasada
Vicent Martí
Thanks for listening!