Representing Rationals

Prof. Daniele Gorla

Rationals in Fixed Point Notation

Still a positional system in base b (≥ 2).
The first m digits are the integer part, the remaining n are the fractional part.

$$c_{m-1}\dots c_1 c_0, c_{-1} c_{-2} \dots c_{-n} = \sum_{i=0}^{m-1} c_i b^i + \sum_{i=-1}^{-n} c_i b^i = \sum_{i=0}^{m-1} c_i b^i + \sum_{i=1}^{n} \frac{c_{-i}}{b^i}$$

with $c_i \in \{ 0 , \dots , b\text{-}1 \}$.

Hence, a rational number N is a pair

$$< Ni , Nf >$$

Made up from an integer part (Ni) and a fractional one (Nf)

Base change

Turn $< Ni , Nf >_a$ into $< Ni' , Nf' >_b$

- For the integer part, we follow the procedure for naturals
- For the fractional part, we work in a similar way:
  - if the arrival base is 10, use the *polynomial method*
  - if the starting base is 10, use the *iterated multiplications* method (see later)
  - otherwise:
    - convert from base a to base 10 (polynomial method)
    - convert the result from base 10 to base b (iterated multiplications)

Polynomial Method (from base b to base 10)

$$c_{m-1}\dots c_1 c_0, c_{-1} c_{-2} \dots c_{-n} = \sum_{i=0}^{m-1} c_i b^i + \sum_{i=1}^{n} \frac{c_{-i}}{b^i}$$

Example: convert $1011{,}011_2$ in base 10

$$1011{,}011_2 = \left(1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 + \frac{0}{2^1} + \frac{1}{2^2} + \frac{1}{2^3}\right)_{10}$$

$$= \left(8+2+1+\frac{1}{4}+\frac{1}{8}\right)_{10} = \left(11+\frac{2+1}{8}\right)_{10} = \left(11+\frac{3}{8}\right)_{10} = 11{,}375$$

Fractional Part Convertion (from base 10 to base *b*)

Let us have a pure fractional number

$F = 0{,}c_{-1}\, c_{-2} \ldots c_{-n}$

We know that F represents

$$\sum_{i=1}^{n} \frac{c_{-i}}{b^i} = \frac{c_{-1}}{b} + \frac{c_{-2}}{b^2} + \frac{c_{-3}}{b^3} + \ldots + \frac{c_{-(n-1)}}{b^{n-1}} + \frac{c_{-n}}{b^n}$$

If we multiply F times b we obtain

$$b \cdot F = c_{-1} + \frac{c_{-2}}{b} + \frac{c_{-3}}{b^2} + \ldots + \frac{c_{-(n-1)}}{b^{n-2}} + \frac{c_{-n}}{b^{n-1}}$$

That is, a number of the form c_{-1} , $c_{-2} \ldots c_{-n}$

Hence, $b{\cdot}F$ is a number whose integer part is the first fractional digit of F and the fractional part is formed by the remaining digits of F.

Fractional Part Conversion (from base 10 to base *b*)

Now, we can iterate on the pure fractional number

$F^{(2)} = 0{,}c_{-2}\, c_{-3} \ldots c_{-n}$

If we multiply $F^{(2)}$ times b we obtain

$$b \cdot F^{(2)} = c_{-2} + \frac{c_{-3}}{b} + \frac{c_{-4}}{b^2} + \ldots + \frac{c_{-(n-1)}}{b^{n-3}} + \frac{c_{-n}}{b^{n-2}}$$

that is a number of the form c_{-2} , $c_{-3} \ldots c_{-n}$

We iterate this procedure until:

- $F^{(k)} = 0$, for some k *(OBS.: differently from the iterated divisions, this is NOT guaranteed to happen)*
- We obtain a periodical part (that returns infinitely often)
- Or we have reached the maximum number of available digits for representing the fractional part in base b

Example:

Convert $17{,}416_{10}$ in base 2 with 8 bits both for the I.P. and the F.P.

1. Convert the integer part (*iterated divisions*):

   17:2 = 8 rem. 1   8:2 = 4 rem. 0   4:2 = 2 rem. 0

   2:2 = 1 rem. 0   1:2 = 0 rem. 1

   Hence, $17_{10} = 10001_2$

2. Convert the fractional part (*iterated multiplications*):

$0{,}416 \times 2 = 0{,}832$	and so	I.P. = 0	F.P. = 0,832
$0{,}832 \times 2 = 1{,}664$	and so	I.P. = 1	F.P. = 0,664
$0{,}664 \times 2 = 1{,}328$	and so	I.P. = 1	F.P. = 0,328
$0{,}328 \times 2 = 0{,}656$	and so	I.P. = 0	F.P. = 0,656
$0{,}656 \times 2 = 1{,}312$	and so	I.P. = 1	F.P. = 0,312
$0{,}312 \times 2 = 0{,}624$	and so	I.P. = 0	F.P. = 0,624
$0{,}624 \times 2 = 1{,}248$	and so	I.P. = 1	F.P. = 0,248
$0{,}248 \times 2 = 0{,}496$	and so	I.P. = 0	F.P. = 0,496

   Hence, $0{,}416_{10} = 0{,}01101010_2$

To conclude, $17{,}416_{10} = 00010001{,}01101010_2$

Remark:

The number obtained in this way is, in general, an approximation (smaller than) of the original number. This happens every time we stop when the F.P. is not 0.

Indeed:

$$00010001{,}01101010_2 = 2^4 + 1 + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^5} + \frac{1}{2^7}$$

$$= 17 + \frac{2^5 + 2^4 + 2^2 + 1}{2^7} = 17 + \frac{32+16+4+1}{128} = 17 + \frac{53}{128}$$

$= 17{,}4140625 < 17{,}416$

Example (with a period):

Convert $120{,}03_{10}$ in base 5

1. Convert the integer part:
   120:5 = 24 rem. 0 24:5 = 4 rem. 4 4:5 = 0 rem. 4
   Hence, $120_{10} = 440_5$
2. Convert the fractional part:

$0{,}03 \times 5 = 0{,}15$	and so	I.P. = 0	F.P. = 0,15
$0{,}15 \times 5 = 0{,}75$	and so	I.P. = 0	F.P. = 0,75
$0{,}75 \times 5 = 3{,}75$	and so	I.P. = 3	F.P. = 0,75
$0{,}75 \times 5 = 3{,}75$	and so	I.P. = 3	F.P. = 0,75
…			

So, $0{,}03_{10} = 0{,}00333\ldots_5$

Hence, $120{,}03_{10} = 440{,}00\overline{3}_5$

Opposite Conversion (with a period):

Convert $0{,}0\overline{3}$ from base 5 to base 10.

Still by using the polynomial method:

$$0{,}0\overline{3}_5 = \left(\frac{0}{5^1} + \frac{3}{5^2} + \frac{3}{5^3} + \ldots\right)_{10} = \sum_{i>1} \frac{3}{5^i} = 3\sum_{i>1} \frac{1}{5^i}$$

$$= 3\left(\sum_{i>0} \frac{1}{5^i} - \frac{1}{5}\right) = 3\left(\frac{1}{4} - \frac{1}{5}\right) = \frac{3}{20} = 0{,}15_{10}$$

where we used the geometrical series: $\sum_{i>0} \frac{1}{c^i} = \frac{1}{c-1}$
(with $c > 1$)

Problems in Fixed Point Notation

The representable interval is small and with very coarse approximations

Example: by having 32 bits (20 for the I.P. and 12 for the F.P.) we have that

- I.P. $\in \{-2^{19}+1, \ldots, 2^{19}-1\} = \{-524.287, \ldots, 524.287\}$
  (if we use the first bit to represent the sign)
- for the F.P. we have at most 4 digits in base 10
  (indeed, $2^{-12} = \frac{1}{4096} \approx 0{,}00025$)

Clearly, we can reduce the I.P. in favour of the F.P., to have a (slightly) higer precision; however, this shrinks the interval amplitude

However, ***this representation is NOT well-suited for real life scientific calulations!!***

Floating Point Representation

A rational r is given by the triple

$$< s, e, m >$$

where the elements are:

- *sign bit* (s=1 if the number is negative, s=0 otherwise)
- *exponent*, an integer e in Base Complemento
- *mantissa*, a rational number m in fixed point repr. in base b

The triple $< s, e, m >$ represents the number

$$(-1)^s \cdot m \cdot b^e$$

This comes from the well-known scientific representation, through which we write

-5×10^3 instead of -5000 or 4×10^{-2} instead of 0,04

Normalized Form

The same number can be represented in many ways:
$- 5 \times 10^3 = - 50 \times 10^2 = -0.5 \times 10^4 = \ldots$

To ensure unicity of the representation of a number, we use a *normalized form*, where the mantissa has the integer part made up of just a single non-zero digit

From now on, we shall always use normalized forms; so, in base 2, the triple $< s , e , m >$ is such that m is a sequence of bits and the represented number is

$$(-1)^s \cdot 1, m \cdot 2^e$$

OBS.: the only non-normalized number is zero

Representation Interval in Floating Point

If we have M bits for the mantissa and E for the exponent

Negative numbers: The mantissa lies in $[-1,\underbrace{11\ldots1}_{M} ; -1,\underbrace{00\ldots0}_{M}]$

Positive Numbers: The mantissa lies in $[+1,\underbrace{00\ldots0}_{M} ; +1,\underbrace{11\ldots1}_{M}]$

The exponent, in 2-compl, lies in $[-2^{E-1} +1 ; +2^{E-1}-1]$

Hence, positive numbers lie in $[\,1\times 2^{-2^{E-1}+1} ; 1,1\ldots1\times 2^{2^{E-1}-1}]$

Negative numbers lie in $[-1,1\ldots\ldots1\times 2^{2^{E-1}-1} ; -1\times 2^{-2^{E-1}+1}]$

Bias

Working with exponents in 2-compl complicates the practical handling of the floating point numbers

In practice, all exponents are translated, to become non-negative

This is done by summing to the exponent a ***bias***
→ with E exponent bits, the bias is $2^{E-1} - 1$

The bias turns the exponent interval from $[-2^{E-1} +1 ; +2^{E-1}-1]$ to $[0 ; 2^E-2]$

REMARK: this is just a trick for comparing positive exponents (that it is easier in practice); the meaning of the exponent remains the previous one, i.e. as an integer number

More on the Biased Representation

In practice, the exponent 0 (in the biased format) is used for specific purposes

Hence, the real interval for the exponent is $[1 ; 2^E-2]$

Special sequences:
- $e = 0 , m = 0$ → zeros (both positive and negative)
- $e = 0 , m \neq 0$ → denormalized numbers
- $e = 2^E-1 , m = 0$ → infinities (both positive and negative)
- $e = 2^E-1 , m \neq 0$ → NaN

Interplay between M and E (with the same $M+E$)

SAPIENZA Università di Roma Dipartimento di Informatica

E=3 bits ($bias$=3) M=4 bits

E \ M	0000	0001	...	1111
001 (=-2)	0,25	0,265625	...	0,484375
010 (=-1)	0,5	0,53125	...	0,96875
011 (= 0)	1	1,0625	...	1,9375
100 (= 1)	2	2,125	...	3,875
101 (= 2)	4	4,25	...	7,75
110 (= 3)	8	8,5	...	15,5

E=4 bits ($bias$=7) M=3 bits

E \ M	000	001	...	111
0001 (=-6)	0,015625	0,017578125	...	0,0234375
...	...	...	...	...
0110 (=-1)	0,5	0,5625	...	0,9375
0111 (= 0)	1	1,125	...	1,875
...	...	...	...	...
1110 (= 7)	128	144	...	240

Precision vs Amplitude

SAPIENZA Università di Roma Dipartimento di Informatica

- ***precision**:* distance between two adjacent numbers
- ***amplitude**:* the absolute value of the biggest/smallest representable number

- Higher **precision** → more bits to the mantissa
- Higher **amplitude** → more bits to the exponent

A compromex is needed!

IEEE Standard 754-1985 (different precisions):

	Half	Single	Double	Quadruple
No. of sign bit	1	1	1	1
No. of exponent bit	5	8	11	15
No. of fraction	10	23	52	111
Total bits used	16	32	64	128
Bias	15	127	1023	16383

Our reference format in all the exercises in this course

Base Changes in Floating Point

SAPIENZA Università di Roma Dipartimento di Informatica

From base 2 (with bias B) to base 10: Given the triple $< s , e , m >$ (that is not a special sequence):

- Write it in the fixed point format: $1{,}m \cdot 2^{e-B} = (h,k)_2$
- Convert $(h,k)_2$ in base 10 by using the polynomial method
- The final number is the positive version of the result, if s=0, its negative version, otherwise

From base 10 to base 2 (with bias B): Given $\pm(h,k)_{10}$:

- use the conversion method for the fixed point format (iterated divisiond for the I.P. and iterated multiplications for the F.P.) to obtain $(p,q)_2$
- Convert $(p,q)_2$ in the (normalized) floating point format, to obtain m and e
- The result is $< s , e+B , m >$, where s=1, if the original number was negative, s=0, otherwise (provided that it is not a special sequence)

Example (from base 10 to 2, and back)

SAPIENZA Università di Roma Dipartimento di Informatica

Convert in base 2 the decimal number $0{,}09375_{10}$, in the IEEE half-precision format.

1. Iterated Multiplications:
   $0{,}09375\times2 = 0{,}1875$ $\quad 0{,}1875\times2 = 0{,}375$ $\quad 0{,}375\times2 = 0{,}75$
   $0{,}75\times2 = 1{,}5$ $\quad 0{,}5\times2 = 1{,}0$
   Hence, $0{,}09375_{10} = 0{,}00011_2$
2. Normalized Floating Point: $1{,}1\times2^{-4}$
3. The triple representation in base 2 (with bias 15) is:
   $< 0 \ , 01011 \ , 1000000000 >$
4. Coming back to base 10, we have:
   $< 0 \ , 01011 \ , 1000000000 > = 1{,}1\times2^{-4} = 0{,}00011_2 = {}^1/_{16} + {}^1/_{32}$
   $= 0{,}09375_{10}$