AULA 1: INTRODUÇÃO

Profª Tânia Camila Kochmanscky Goulart

Introdução

Segundo a autora Neide Franco, o conjunto de números representáveis em qualquer máquina é finito, portanto discreto, ou seja, não e possível representar em uma máquina todos os números de um dado intervalo [a; b].

A implicação imediata desse fato é que o resultado de uma simples operação aritmética ou o cálculo de uma função, realizadas neste conjunto de números, podem conter erros. A menos que medidas apropriadas sejam tomadas, essas imprecisões causadas, por exemplo, por:

1. Simplificações no modelo matemático, necessárias para se obter um modelo matemático solúvel;
2. Erros de truncamento (troca de uma serie infinita por uma finita);
3. Erros de arredondamento, devido a própria estrutura da máquina;
4. Erros nos dados, dados imprecisos obtidos experimentalmente, ou arredondados na entrada podem diminuir, e algumas vezes destruir, a precisão dos resultados.

Assim, o objetivo inicial é o estudo dos erros inerentes a estrutura computacional, dar subsídios para evitá-los e interpretar os resultados obtidos.

Fontes de Erros

- Suponha que você está diante do seguinte problema: você está em cima de um edifício que não sabe a altura, mas precisa determiná-la. Tudo que tem em mãos é uma bola de metal e um cronômetro. O que fazer?
- Conhecemos também a equação onde:

$$d = d_0 + v_o t + (1/2)at^2$$

  - d é a posição final;
  - do é a posição inicial;
  - Vo é a velocidade inicial;
  - t é o tempo percorrido;
  - a é a aceleração.

Erros na modelagem

- A bolinha foi solta do topo do edifício e marcou-se no cronômetro que ela levou 2 segundos para atingir o solo. Com isso podemos conclui a partir da equação acima que a altura do edifício é de 19,6 metros.
- Essa resposta é confiável? Onde estão os erros?
- Erros de modelagem:
  - Resistência do ar,
  - Velocidade do vento,
  - Forma do objeto, etc.
- *Estes erros estão associados, em geral, à simplificação do modelo matemático.*

- Erros de resolução:

  – Precisão dos dados de entrada

  (Ex. Precisão na leitura do cronômetro para t = 2,3 segundos, h = 25,92 metros, gravidade);
- – Forma como os dados são armazenados;
- – Operações numéricas efetuadas;
- – Erro de truncamento (troca de uma série infinita por uma série finita).

Representação numérica

- Exemplo 1: Calcule a área de uma circunferência de raio igual a 100m.
- Resultados obtidos:

  1. A = 31400m^2;
  2. A = 31416m^2;
  3. A = 31415,92654m^2.

- Como justificar as diferenças entre os resultados apresentados no exemplo? É possível obter exatamente esta área?

- Os erros ocorridos dependem da representação do número (neste caso, do numero pi) na máquina utilizada e do número máximo de dígitos usados na sua representação.

- O número pi, por exemplo, não pode ser representado através de um numero finito de dígitos decimais. No exemplo anterior, o número foi escrito como 3,14, 3,1416 e 3,141592654 respectivamente.

- Para cada representação foi obtido um resultado diferente, e o erro neste caso depende exclusivamente da aproximação escolhida para pi.

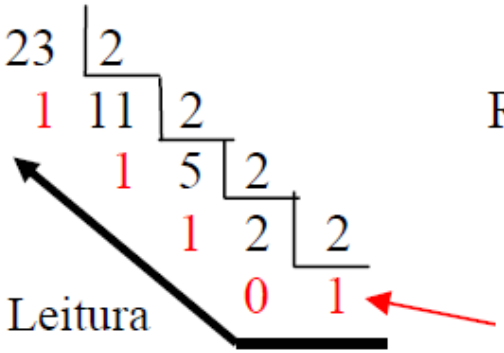
- Qualquer que seja a circunferência, a sua área nunca será obtida exatamente de forma numérica!
- Logo, qualquer cálculo que envolva números que não podem ser representados através de um número finito de dígitos não fornecerá como resultado um valor exato.

- Assim, voltamos novamente a ideia de que o conjunto dos números representáveis em qualquer máquina é finito, e portanto, *discreto, ou seja não é possível representar em uma máquina todos os números de um* dado intervalo [a,b].
- A representação de um número depende da BASE escolhida e do número máximo de dígitos usados na sua representação.

Conversão de número decimal para binário

- Para convertermos um número decimal em um número binário, devemos aplicar um método para a parte intera e um método para a parte fracionária.

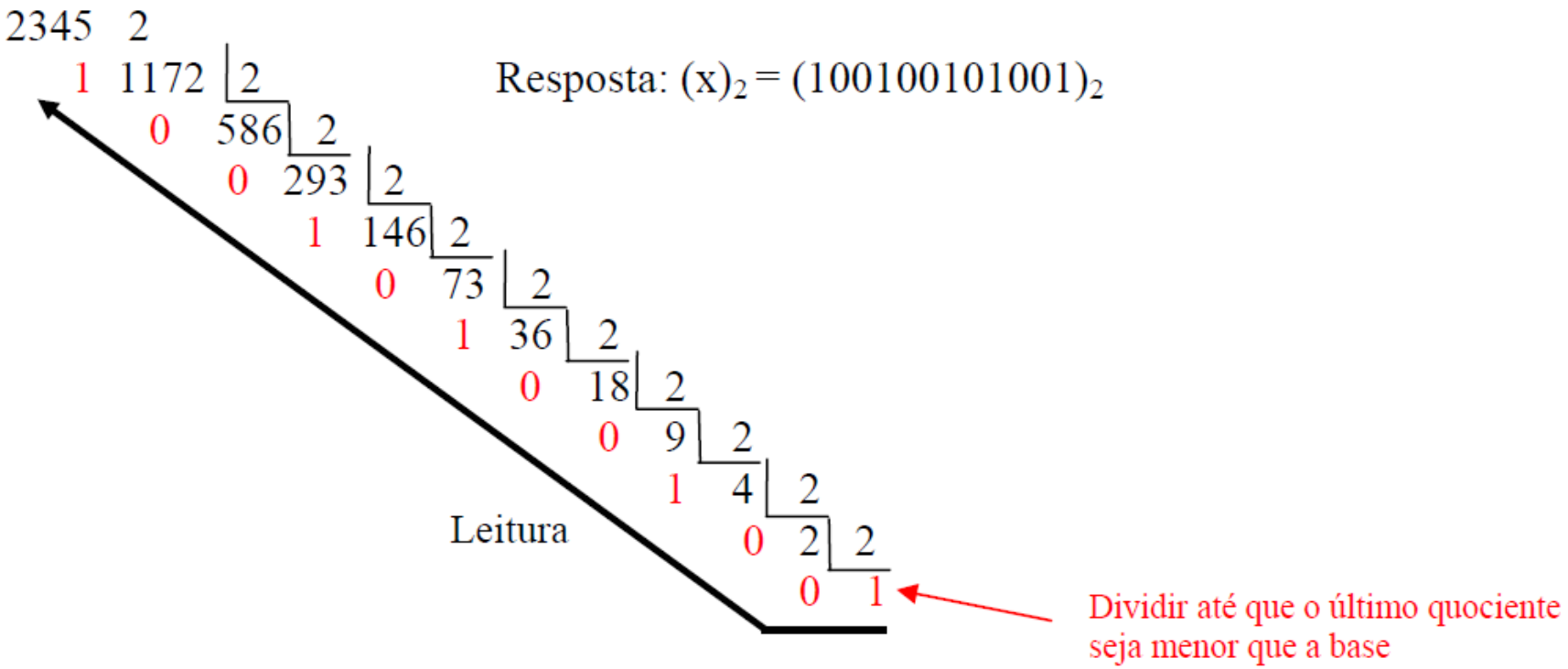
$(23)_{10} \rightarrow (x)_2$ Usando o método das divisões sucessivas.



Resposta: $(x)_2 = (10111)_2$

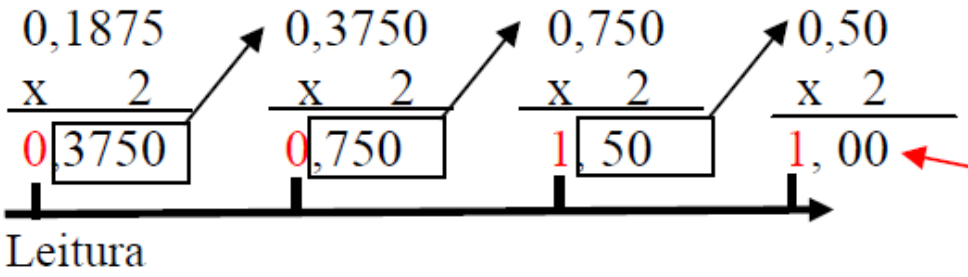
Dividir até que o último quociente seja menor que a base

$(2345)_{10} \rightarrow (x)_2$ Usando o método das divisões sucessivas.



Para números fracionários utilizamos a regra da multiplicação.

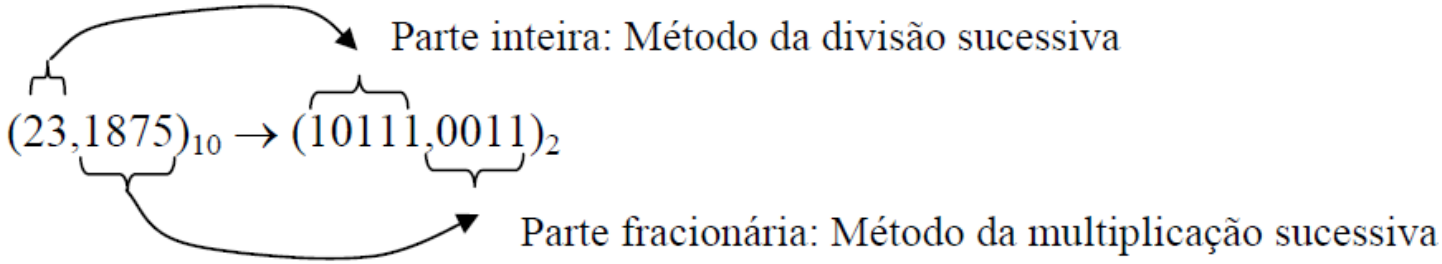
$(0,1875)_{10} \rightarrow (x)_2$



Resposta: $(x)_2 = (0,0011)_2$

Parar quando não existir mais a parte fracionária

Obs. O fato de um número não ter representação finita no sistema binário pode acarretar a ocorrência de erros aparentemente inexplicáveis nos cálculos dos dispositivos eletrônicos.



Conversão de número binário para decimal

$(10111)_2 \rightarrow (x)_{10}$

$(10111)_2 = \underbrace{1x2^4} + \underbrace{0x2^3} + \underbrace{1x2^2} + \underbrace{1x2^1} + \underbrace{1x2^0} = 23 = (23)_{10}$

$16 + 0 + 4 + 2 + 1$

d_4 d_0 d_1

$(110,11)_2 \rightarrow (x)_{10}$

$(110,11)_2 = \underbrace{1x2^2} + \underbrace{1x2^1} + \underbrace{0x2^0} + \underbrace{1x2^{-1}} + \underbrace{1x2^{-2}} = 6,75 = (6,75)_{10}$

$4 + 2 + 0 + 1/2 + 1/4$

d_2 d_{-1} d_0

Exercício

- Nem todo número real na base decimal possui uma representação finita na base binária. Faça a conversão de (0,1) na base 10 para a base 2 e explique a resposta encontrada. Em seguida, coloque o número encontrado na base 10 novamente. Observe o resultado.

0,1	0,2	0,4	0,8	0,6	0,2	
x 2	x 2	x 2	x 2	x 2	x 2	....
0,2	**0**,4	**0**,8	**1**,6	**1**,2	**0**,4	

Observe-se que as multiplicações começam a se repetir, impossibilitando que se chegue a um resultado com a parte fracionária nula. Portanto o 0,1, da base 10 , tem a seguinte representação na base 2:

$$(0,1)_{10} = (0,0001100110011...)_2$$

Ou seja, 0,10 da base 10 não tem representação finita na base 2 podendo ser representado , nesta base, somente de forma aproximada.

Em um computador com tamanho de palavra de 16 bits, $(0,1)_{10}$ é armazenado como

$$(0,0001100110011001)_2$$

$$(0,0999908447265625)_{10}$$

Produzindo, portanto, um erro da ordem de 9×10^{-6}.

Número em outras bases

Dado o número 12.20 que está na base 4, representá-lo na base 3.

1. *Representar o* $(12.20)_4$ *na base 10:*

$$\begin{aligned} 12 &= 2 \times 4^0 + 1 \times 4^1 = 6, \\ 0.20 &= 2 \times 4^{-1} + 0 \times 4^{-2} = \frac{2}{4} = 0.5 \end{aligned}$$

Portanto: $(12.20)_4 = (6.5)_{10}$.

2. *Representar* $(6.5)_{10}$ *na base 3:*

$$\begin{array}{r|l} 6 & 3 \\ \hline (0) & (2) \end{array} \qquad (6)_{10} = (20)_3$$

$$
\begin{aligned}
0.5 \times 3 &= \mathbf{1}.5 \\
0.5 \times 3 &= \mathbf{1}.5 \\
&\vdots
\end{aligned}
$$

Assim: $(12.20)_4 = (20.111\ldots)_3$. *Observe que o número dado na base 4 tem representação exata na base 4, mas não na base 3.*

Exercícios

Resolver a lista 01 - exercícios 3 ao 5

Representação de um número inteiro

A representação de um número inteiro não apresenta dificuldade. Qualquer computador trabalha internamente com uma base fixa β, onde β é um inteiro ≥ 2 e é escolhido como uma potência de 2.

Assim, dado um número inteiro $n \neq 0$, ele possui uma única representação:

$$n = \pm (n_{-k}\, n_{-k+1} \ldots n_{-1}\, n_0) = \pm \left(n_0\beta^0 + n_{-1}\beta^1 + \ldots + n_{-k}\beta^k\right)$$

onde n_i, $i = 0, -1, \ldots, -k$ são inteiros satisfazendo $0 \leq n_i \leq \beta$ e $n_{-k} \neq 0$.

Exemplo 2. *Na base $\beta = 10$, o número 1997 é representado por:*

$$1997 = 7 \times 10^0 + 9 \times 10^1 + 9 \times 10^2 + 1 \times 10^3$$

e é armazenado como $n_{-3}\, n_{-2}\, n_{-1}\, n_0$.

Representação de um número real por ponto fixo

A representação de um número real no computador pode ser feita por representação em *Ponto Fixo* ou *Ponto Flutuante*,

(i) Representação em Ponto Fixo: Dado um número real $x \neq 0$, ele será representado em ponto fixo por:

$$x = \pm \sum_{i=k}^{n} x_i \beta^{-1},$$

onde k e n são inteiros satisfazendo $k < n$ e, usualmente, $k \leq 0$ e $n > 0$ e os x_i são inteiros satisfazendo $0 \leq x_i < \beta$.

Exemplo 3. *Na base 10, o número 1997.16 é representado por:*

$$1997.16 = \sum_{i=-3}^{2} x_i \beta^{-i} = 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 7 \times 10^0 + 1 \times 10^{-1} + 6 \times 10^{-2}$$

e é armazenado como $x_{-3}\, x_{-2}\, x_{-1}\, x_0 \,.\, x_1\, x_2$.

Representação no número real por ponto flutuante

- A representação de números reais mais utilizada em máquinas é a de ponto flutuante. Este número possui três partes:

sinal, parte fracionária (mantissa) e o expoente

$$m = \pm\ ,d_1 d_2 d_3 \ldots\ d_t \times \beta^e$$

$d_{i's}$: dígitos da parte fracionária, $d_1 \neq 0$, $0 \leq d_i \leq \beta\text{-}1$
β: base (em geral 2, 10 ou 16),
t: no de dígitos na mantissa.
e: expoente inteiro.

Em outras palavras, $0.d_1d_2d_3...d_t$ é uma fração na base *b*, também chamada de **mantissa**, com $0 \leq d_i \leq b\text{-}1$, para todo $i = 1,2,3,...,t$, sendo *t* o número máximo de dígitos da mantissa que é determinado pelo comprimento de palavra do computador; **e** é um expoente que varia em um intervalo dado pelos limites da máquina utilizada.

- Esse tipo de representação é chamada de **ponto flutuante** pois o ponto da fração "flutua" conforme o número a ser representado e sua posição é expressa pelo expoente **e.**

Equivalente à:
$(0{,}0001100110011001.....)_2$

Ex.

x=34,2 (decimal); β=10; t=4 $x=0{,}3420 \times 10^{2}$	x=0,1 (decimal) ; β=2; t=9 $x=0{,}110011001 \times 2^{-3}$

Exemplos

Número na base decimal	*Representação em ponto flutuante*	*mantissa*	*base*	*Expoente*
1532	0.1532×10^{4}	0.1532	10	4
15.32	0.1532×10^{2}	0.1532	10	2
0.00255	0.255×10^{-2}	0.255	10	-2
10	0.10×10^{2}	0.10	10	2
10	0.1010×2^{4}	0.1010	2	4

Outros exemplos

$$x_1 = 0.35 \qquad x_2 = -5.172 \qquad x_3 = 0.0123 \qquad x_4 = 5391.3 \qquad x_5 = 0.0003,$$

onde todos estão na base $\beta = 10$, *em ponto flutuante na forma normalizada.*

Temos:

$$\begin{aligned}
0.35 &= (3 \times 10^{-1} + 5 \times 10^{-2}) \times 10^{0} = 0.35 \times 10^{0} \\
-5.172 &= -(5 \times 10^{-1} + 1 \times 10^{-2} + 7 \times 10^{-3} + 2 \times 10^{-4}) \times 10^{1} = -0.5172 \times 10^{1} \\
0.0123 &= (1 \times 10^{-1} + 2 \times 10^{-2} + 3 \times 10^{-3}) \times 10^{-1} = 0.123 \times 10^{-1} \\
5391.3 &= (5 \times 10^{-1} + 3 \times 10^{-2} + 9 \times 10^{-3} + 1 \times 10^{-4} + 3 \times 10^{-5}) \times 10^{4} = 0.53913 \times 10^{4} \\
0.0003 &= (3 \times 10^{-1}) \times 10^{-3} = 0.3 \times 10^{-3}
\end{aligned}$$

Notação: para representar um sistema numérico em ponto flutuante normalizado, na base β, com t dígitos significativos e com limites dos expoentes m e M, usamos $F(\beta, t, m, M)$.

Assim, um número em $F(\beta, t, m, M)$ será representado por:

$$\pm 0.d_1\, d_2\, \ldots,\, d_t \times \beta^e$$

onde $d_1 \neq 0$ e $-m \leq e \leq M$.

Considere o sistema $F(10, 3, 2, 2)$. Represente neste sistema os números do Exemplo

Neste sistema, um número será representado por:

$$0.d_1\, d_2\, d_3 \times 10^e,$$

onde $-2 \leq e \leq 2$. Assim:

$$\begin{aligned} 0.35 &= 0.350 \times 10^0, \\ -5.172 &= -0.517 \times 10^1, \\ 0.0123 &= 0.123 \times 10^{-1} \end{aligned}$$

Note que os números 5391.3 e 0.0003 não podem ser representados no sistema, pois $5391.3 = 0.539 \times 10^4$ *e, portanto, o expoente é maior que 2, causando* **overflow**. *Por outro lado,* $0.0003 = 0.300 \times 10^{-3}$ *e, assim, o expoente é menor do que -2, causando* **underflow**.

Exemplo 1: Vejamos os sistemas de ponto flutuante de algumas máquinas antigas: HP 25, F(10,9,-98,100); Texas SR 50 e HP 41C, F(10,10,-98,100); Texas SR 52, F(10,12,-98,100); IBM 360/370, F(16,6,-64,63); Burroughs B 6700, F(8,13,-51,77). Comparando com sua calculadora ou seu microcomputador, estas máquinas podem ser ditas obsoletas, no ponto de vista do sistema de ponto flutuante?

Máquina e Aritmética	β	t	e_{min}	e_{max}
Cray-1 Precisão Simples	2	48	-8192	8191
Cray-1 Precisão Dupla	2	96	-8192	8191
DEC VAX formato G Dupla	2	53	-1023	1023
DEC VAX formato D Dupla	2	56	-127	127
Calculadoras HP 28 e 48G	10	12	-499	499
IBM 3090 Precisão Simples	16	6	-64	63
IBM 3090 Precisão Dupla	16	14	-64	63
IBM 3090 Precisão Extendida	16	28	-64	63
IEEE Precisão Simples	2	24	-126	127
IEEE Precisão Dupla	2	53	-1022	1023
PDP 11	2	24	-128	127
Control Data 6600	2	48	-976	1070

Para entender melhor

- Os números reais podem ser representados por uma reta contínua. Entretanto, em ponto flutuante, podemos representar apenas pontos discretos na reta real.

- Exercício 1:

  Considere o número F (2, 3, 1, 2). **Quantos** e **quais** números podem ser representados neste sistema?

Temos que $\beta = 2$, então os dígitos podem ser 0 ou 1; $m = 1$ e $M = 2$, então, $-1 \leq e \leq 2$ e $t = 3$. Assim, os números são da forma:

$$\pm 0.d_1\, d_2\, d_3 \times \beta^e$$

Logo, temos: duas possibilidades para o sinal, uma possibilidade para d_1, duas possibilidades para d_2 e d_3 e quatro possibilidades para β^e. Fazendo o produto $2 \times 1 \times 2 \times 2 \times 4 = 32$. Assim, neste sistema podemos representar 33 números, visto que o zero faz parte de qualquer sistema.

Para responder quais são os números, notemos que as formas da mantissa são: 0.100, 0.101, 0.110 e 0.111, e as formas de β^e são: 2^{-1}, 2^0, 2^1 e 2^2. Assim, obtemos os seguintes números:

$$0.100 \times \begin{cases} 2^{-1} &= (0.25)_{10} \\ 2^0 &= (0.5)_{10} \\ 2^1 &= (1.0)_{10} \\ 2^2 &= (2.0)_{10} \end{cases}$$

desde que $(0.100)_2 = (0.5)_{10}$;

$$0.101 \times \begin{cases} 2^{-1} &= (0.3125)_{10} \\ 2^{0} &= (0.625)_{10} \\ 2^{1} &= (1.25)_{10} \\ 2^{2} &= (2.5)_{10} \end{cases}$$

desde que $(0.101)_2 = (0.625)_{10}$*;*

$$0.110 \times \begin{cases} 2^{-1} &= (0.375)_{10} \\ 2^{0} &= (0.75)_{10} \\ 2^{1} &= (1.5)_{10} \\ 2^{2} &= (3.0)_{10} \end{cases}$$

desde que $(0.110)_2 = (0.75)_{10}$*;*

$$0.111 \times \begin{cases} 2^{-1} &= (0.4375)_{10} \\ 2^{0} &= (0.875)_{10} \\ 2^{1} &= (1.75)_{10} \\ 2^{2} &= (3.5)_{10} \end{cases}$$

desde que $(0.111)_2 = (0.875)_{10}$.

Podemos representar também por uma tabela

Seja o sistema de ponto flutuante F = F(2, 3, -1, 2). Como a base é dois, os dígitos possíveis são 0 ou 1. Assim, como os números deste sistema devem ter até três dígitos, as mantissas podem ser: 0.100, 0.101, 0.110 e 0.111. Estes números representam, respectivamente, as quantidades 1, 5/4, 3/2 7/4. E mais, os expoentes da base possíveis são –1, 0, 1 ou 2. Portanto, na tabela abaixo escrevemos (em negrito) todos os números positivos do sistema de ponto flutuante, já colocados na base dez:

Expoentes		Mantissas			
e	2^e	0.100	0.101	0.110	0.111
-1	1/2	**1/4**	**5/16**	**3/8**	**7/16**
0	1	**1/2**	**5/8**	**3/4**	**7/8**
1	2	**1**	**5/4**	**3/2**	**7/4**
2	4	**2**	**5/2**	**3**	**7/2**

Exercício2

Considerando o mesmo sistema F (2,3,1,2)

represente os números: $x_1 = 0.38$, $x_2 = 5.3$ *e* $x_3 = 0.15$ *dados na base 10.*

- Solução do exercício 2:

$$(0.38)_{10} = 0.110 \times 2^{-1},\ (5.3)_{10} = 0.101 \times 2^{3}\ e\ (0.15)_{10} = 0.100 \times 2^{-2}.$$

apenas o primeiro número pode ser representado no sistema,

pois para o segundo teremos **overflow** *e, para o terceiro,* **underflow**.

- Exercício 3: Numa máquina que opera no sistema

$$\beta = 10; t = 3; e \in [-5,5]$$

Portanto, F (10 , 3 , 5 , 5), qual o maior e o menor número desta máquina em módulo?

Resposta do exercício 3

- Nesta máquina, em módulo, o

menor número, em módulo: $m = (0.100) \times 10^{-5} = 10^{-6}$

maior número, em módulo: $M = (0.999) \times 10^{5} = 99900$

Outros exemplos

Ex: (4)

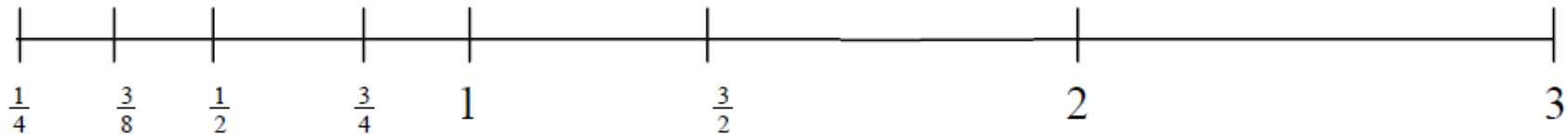
Considere $F(2,2,-1,2)$, com número normalizado, isto é, $d_1 \neq 0$. Os números serão:
$\pm .10 \times 2^e$ ou $\pm .11 \times 2^e$, sendo $-1 \leq e \leq 2$.

Convertendo para decimal, temos:
$.10 = ½$ e $.11 = ¾$

Com isso, os únicos números positivos representáveis nesse computador são:

Mantissa		Expoentes
$1/2 \times 2^e$		
$3/4 \times 2^e$	para	e= -1, 0, 1 e 2

Ou seja, ¼, ½, 1, 2, 3/8, 3/4, 3/2 e 3, que podem ser representados na reta numerada:



Alem desses números, os seus respectivos números negativos e o numero zero também serão representados.

Representação dos números

O conjunto de números de números reais é infinito, entretanto, a sua representação em um sistema de ponto flutuante é limitada, pois é um sistema **finito**. Essa limitação tem duas origens:

A) a faixa dos expoentes é limitada ($e_{min} \leq e \leq e_{max}$);

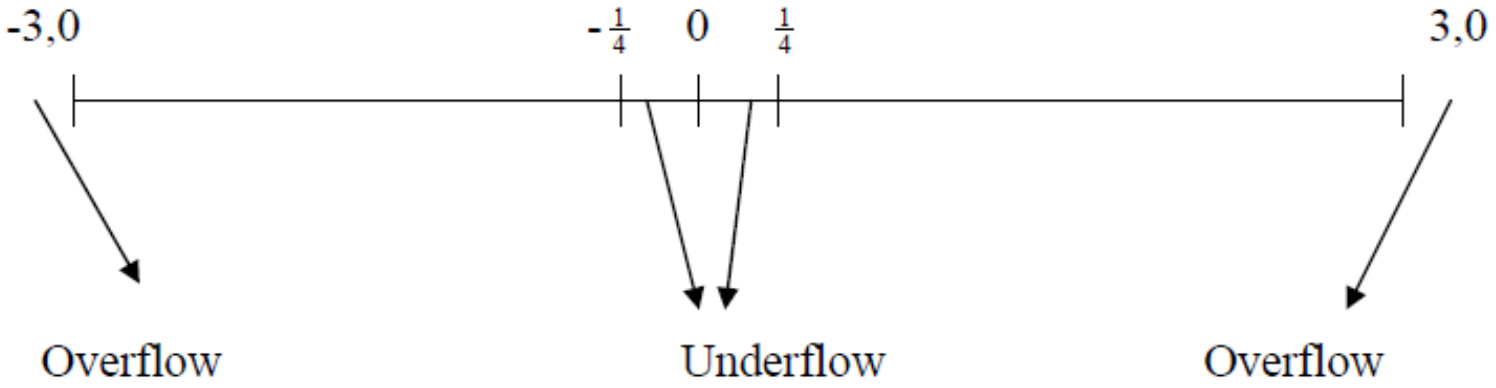
B) a mantissa representa um número finito de números ($\beta^{t-1} \leq m \leq \beta^{t-1}$)

Faixa dos expoentes é limitada ($e_{min} \leq e \leq e_{max}$);

Sempre que uma operação aritmética produz um número com expoente superior ao expoente máximo, tem-se o fenômeno de "**overflow**". De forma similar, operações que resultem em expoente inferior ao expoente mínimo tem-se o fenômeno de "**underflow**".

Reta numerada do ex. (3)

No caso do exemplo dado, pode-se observar qual as regiões que ocorrem o overflow e o underflow. Neste caso, considera-se a parte positiva e negativa da aritmética do exemplo.



Exercícios

- Exercícios
  - Resolver a lista 01 - exercícios 1, 2, 6 e 7

SURPRESAAAAA!!!

Número total de elementos de uma aritmética de ponto flutuante

$$n.\ de\ elementos = 2(\beta - 1)\beta^{t-1}(e_{\max} - e_{\min} + 1) + 1$$

Para contabiliza os números negativos

Para o numero zero

Linguagem de programação

Em qualquer linguagem é possível especificar a representação que deve ser usada para os números a serem armazenados em uma dada variável. Na linguagem C, para **números inteiros** - *int* 2 bytes => (-32768 , 32767).

No caso da representação de **ponto flutuante** - *float* => número máximo de dígitos na base binária igual a 24 (*t=24*) e expoente entre –126 e 127. Portanto, uma variável declarada como *float* pode armazenar números reais entre $\sim 10^{-38}$ e $\sim 10^{38}$.