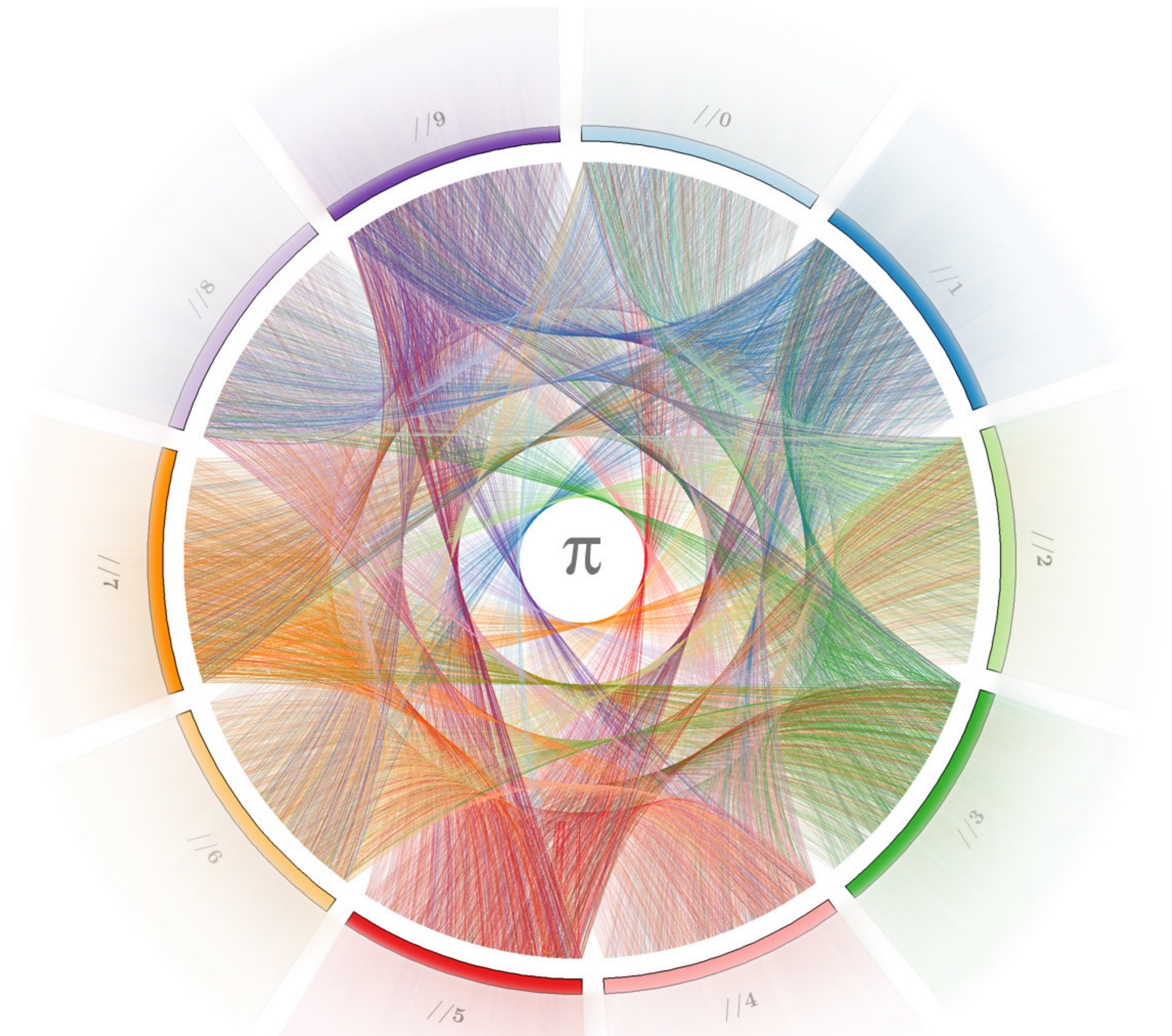
Représentation des rationnels



GIF-1001 Ordinateurs : Structure et Applications

Jean-François Lalonde

Source image

Représenter des rationnels en décimal

- Rappel: un nombre **rationnel** peut être exprimé par un ratio d’entiers relatifs (positifs ou négatifs):

- Nous allons représenter les nombres rationnels en version « nombres à virgule »:

- **Attention**: seuls les rationnels qui peuvent se représenter avec un **nombre fini de décimales** après la virgule pourront être stockés dans un ordinateur

Représenter des rationnels en décimal

Décortiquons 6 431,986…

Position	3	2	1	0	,	-1	-2	-3
Valeur	10^3	10^2	10^1	10^0	,	10^{-1}	10^{-2}	10^{-3}
Symbole	6	4	3	1	,	9	8	6
=	6000	400	30	1		0,9	0,08	6

Nombres rationnels en binaire?

- Une possibilité (sur 16 bits):
  - mettons la virgule au milieu:

Valeur	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	,	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
Position	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	,	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

    - 8 premiers bits: 2^0 à 2^7 (0 à 255)
    - 8 derniers bits: 2^{-1} à 2^{-8} (1/256 à 255/256, 0.00390625 à 0.99609375)

0 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0

Nombres rationnels en binaire?

- Une possibilité (sur 16 bits):
  - mettons la virgule au milieu:

Valeur	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	,	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
Position	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	,	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

    - 8 premiers bits: 2^0 à 2^7 (0 à 255)
    - 8 derniers bits: 2^{-1} à 2^{-8} (1/256 à 255/256, 0.00390625 à 0.99609375)

- Problèmes?
  - très limité!
    - quelle est la valeur maximale?
      - 255,99609375 (255 255/256)
    - quelle est la précision (ou la plus petite différence entre deux nombres)?
      - 1/256

Rappel: décimal, notation scientifique

$6\,500 = (+)\ 6{,}5 \times 10^3$

Rappel: décimal, notation scientifique

$$6\,500 = (\mathbf{+})\ 6{,}5 \times 10^3$$

signe (+): indique si le nombre est positif ou négatif

Rappel: décimal, notation scientifique

$$6\ 500 = (+)\ 6{,}5 \times \mathbf{10}^3$$

signe (+): indique si le nombre est positif ou négatif

base (10): décimale

Rappel: décimal, notation scientifique

$$6\ 500 = (+)\ 6{,}5 \times 10^{\mathbf{3}}$$

signe (+): indique si le nombre est positif ou négatif

base (10): décimale

exposant (3): indique l'ordre de grandeur

Rappel: décimal, notation scientifique

6 500 = (+) **6,5** x 10^3

signe (+): indique si le nombre est positif ou négatif

base (10): décimale

exposant (3): indique l'ordre de grandeur

significande (6,5): détermine la précision de la valeur représentée

divisée en deux: **caractéristique** (6) et **mantisse** (0,5)

Rappel: décimal, notation scientifique

$$6\ 500 = (+)\ 6{,}5 \times 10^{3}$$

signe (+): indique si le nombre est positif ou négatif

base (10): décimale

exposant (3): indique l'ordre de grandeur

significande (6,5): détermine la précision de la valeur représentée

divisée en deux: **caractéristique** (6) et **mantisse** (0,5)

$$(\text{signe})\ \text{caractéristique},\text{mantisse} \times \text{base}^{\text{exposant}}$$

Nombres rationnels, en binaire

- La norme IEEE 754 a été adoptée en 2008 pour les nombres rationnels sur 32, 64 et 128 bits
- Très similaire à la notation scientifique:

$$(\text{signe})\ 1{,}\text{mantisse} \times 2^{(\text{exposant}-127)}$$

- Sur 32 bits:
  - signe: un bit
  - base: 2, donc binaire. Comme cette base est toujours 2, on n'a pas besoin de la stocker (c'est implicite)
  - caractéristique: toujours 1. On n'a pas besoin de la stocker (implicite également)
  - exposant (décalé): 8 bits (donc de 0 à 255), mais on soustrait 127, donc de -127 à +128

1 bit	8 bits	23 bits
signe	exposant	mantisse

Exercice #1: IEEE754 vers décimal

Convertir 0x411A0000 (écrit en IEEE754) en décimal.

(signe) 1,mantisse x $2^{(\text{exposant}-127)}$

1 bit	8 bits	23 bits
signe	exposant	mantisse

Ex. 1, étape 1: hexadécimal vers binaire

Convertir 0x411A0000 (écrit en IEEE754) en décimal.

4	1	1	A	0	0	0	0

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Ex. 1, étape 2: binaire vers décimal

Convertir 0x411A0000 (écrit en IEEE754) en décimal.

0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(signe) 1,mantisse x $2^{(\text{exposant}-127)}$

1 bit	8 bits	23 bits
signe	exposant	mantisse

Ex. 1, étape 2: binaire vers décimal

Convertir 0x411A0000 (écrit en IEEE754) en décimal.

0	1	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Bit de signe = 0, donc nombre positif

$$\boxed{(+)\ 1{,}\text{mantisse} \times 2^{(\text{exposant}-127)}}$$

- Exposant = 0b10000010 = 130. 130-127 = 3

$$\boxed{(+)\ 1{,}\text{mantisse} \times 2^{3}}$$

- Mantisse = 0b0011010…
  - = $1 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-6}$ = 0,125 + 0,0625 + 0,015625 = 0,203125

$$\boxed{(+)\ 1{,}203125 \times 2^{3} = 9{,}625}$$

(signe) 1,mantisse x $2^{(\text{exposant-127})}$

1 bit	8 bits	23 bits
signe	exposant	mantisse

Exercice #2

Convertir 0x40D00000 (écrit en IEEE754) en décimal

- Tout d'abord, convertissons en binaire

4	0	D	0	0	0	0	0

0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Séparons la chaîne de bits en sections correspondants aux champs de l'IEEE754 (signe, exposant, mantisse)

0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Analysons chaque champ indépendamment
  - Bit de signe: 0 donc positif (+)
  - Exposant: 0b10000001 = 129. 129 - 127 = 2.
  - Mantisse: 0b1010 000… $2^{-1} + 2^{-3}$ = 0,5 + 0,125 = 0,625
- Résultat:
  - (+) 1,625 x 2^{2} = 6,5

(signe) 1,mantisse x $2^{(\text{exposant}-127)}$

1 bit	8 bits	23 bits
signe	exposant	mantisse

Exercice #1: décimal vers IEEE754

Convertir 12,5 en binaire sur 32 bits avec IEEE 754.

Écrire la réponse en hexadécimal.

(signe) 1,mantisse x $2^{(\text{exposant}-127)}$

1 bit	8 bits	23 bits
signe	exposant	mantisse

Ex. 1, étape 1: décimal vers binaire (IEEE754)

Convertir 12,5 en binaire sur 32 bits avec IEEE 754

(signe) 1,mantisse x $2^{(exposant-127)}$

1 bit	8 bits	23 bits
signe	exposant	mantisse

Ex. 1, étape 1: décimal vers binaire (IEEE754)

Convertir 12,5 en binaire sur 32 bits avec IEEE 754

- 12,5 est positif, donc bit de signe = 0

0																															

- 12,5 = 0b1100,1 = 1,1001 x 2^3

- nous voulons que exposant - 127 = 3, alors exposant = 130 soit 0b10000010

0	1	0	0	0	0	0	1	0																							

- la mantisse est 1001000…

0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(signe) 1,mantisse x $2^{(exposant-127)}$

1 bit	8 bits	23 bits
signe	exposant	mantisse

Ex. 1, étape 2: binaire vers hexadécimal

Écrire votre réponse en hexadécimal

0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

donc, 0x41480000

Exercice #2: décimal vers IEEE754

Convertir -16,125 en hexadécimal (IEEE754)

- Exprimons le nombre avec des puissances de 2

$16 = 2^4 \qquad 0{,}125 = 2^{-3}$

- Convertissons en binaire

$-10000{,}001_b$

- Décalons la virgule vers la gauche

$-10000{,}001_b = -1{,}0000001_b \times 2^4$

- Identifions chacun des champs du format IEEE754
  - Signe - : bit à 1

1																															

  - Exposant: exposant - 127 = 4. 4+127 = 131 = 10000011_b

1	1	0	0	0	0	0	1	1																							

  - Mantisse: $00000010\ldots_b$

1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Décimal vers IEEE754 (suite)

Convertir -16,125 en hexadécimal (IEEE754)

1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Regroupons les bits par groupes de 4

1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Convertissons en hexadécimal

C	1	8	1	0	0	0	0

- Résultat: 0xC1810000

Considérations

- Intervalle: 1.2E-38 to 3.4E+38
  - Plus que 4G valeurs mais seulement 32 bits?
  - 1267650600228229401496703205376 Oui
  - 1267650600228229401496703205377 Non
- $(2^{60}-2^{60})+1$ donne $0+1 = 1$ car la soustraction se fait exactement;
- $(2^{60}+1)-2^{60}$ donne 0, car, avec une précision limitée à 53 bits, $2^{60}+1$ s'arrondit à 2^{60}.
- Même fonctionnement pour les autres tailles de nombres à virgule flottante mais avec plus de bits

Cas spéciaux IEEE754

- Exposant et mantisse à 0:
  - 0x00000000
  - Le nombre est zéro (+0 ou -0)
- Exposant à 255 et mantisse à 0
  - 0x7F800000 ou 0xFF800000
  - $+\infty$ ou $-\infty$
- Exposant à 255 et mantisse différente de 0
  - ex: 0xFFC00000
  - Pas un nombre («not a number», ou NaN)
  - Exemples: log(-1), 0/0.

Outils pratiques

http://www.binaryconvert.com

https://float.exposed/0x3f800000

PHIR™ #3

- A priori, **nous ne pouvons pas savoir** ce qu'une chaîne binaire signifie.
    - Ex: que veut dire 0x416C6C6F (sur 32 bits)?
    - La bonne réponse est: ça dépend!

entier non-signé	1097624687
entier signé	1097624687
réel	14,47764

- Il nous faut donc savoir quel format utiliser pour bien interpréter les données