Programmation et algèbre

INFO1 - semaines 45 à 48

Guillaume CONNAN

novembre 2015

IUT de Nantes - Dpt d'informatique

Sommaire

1. Monoïdes
2. Map-Reduce
3. Groupe
   - Le groupe (Z/nZ,+)
4. Anneaux et corps
   - Anneaux
   - Corps
5. Les Complexes

Sommaire

1 Monoïdes
2 Map-Reduce
3 Groupe
- Le groupe (Z/nZ,+)

4 Anneaux et corps
- Anneaux
- Corps

5 Les Complexes

- `1 + 1 = 2`
- `"Doctor " + "Who" = "Doctor Who"`
- `panier.ajoute_contenu(autre_panier) = gros_panier`

Parlons types

- `int + int = int`
- `string + string = string`
- `panier.ajoute_contenu(panier) = panier`

- `1 + 1 = 2`
- `"Doctor " + "Who" = "Doctor Who"`
- `panier.ajoute_contenu(autre_panier) = gros_panier`

Parlons types

- `int + int = int`
- `string + string = string`
- `panier.ajoute_contenu(panier) = panier`

- `1 + 1 = 2`
- `"Doctor " + "Who" = "Doctor Who"`
- `panier.ajoute_contenu(autre_panier) = gros_panier`

Parlons types

- `int + int = int`
- `string + string = string`
- `panier.ajoute_contenu(panier) = panier`

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

IUT NANTES Informatique

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

IUT NANTES Informatique

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

Premier commandemant : STABILITÉ

UN MONOÏDE PREND DU TYPE T ET RETOURNE DU TYPE T

- `3 / 2 = 1.5` (`int / int = float`) ✗
- `True and False = False` (`Bool and Bool = Bool`) ✓
- `[1, 2] + [3, 4] = [1, 2, 3, 4]` (`list + list = list`) ✓
- `3 == 4 = False` (`int == int = Bool`) ✗

Pourquoi la stabilité ?

Pour pouvoir enchaîner des actions (... composer des fonctions !)

- `connexion = host + ":" + port + "/" + chemin`
- `email.To(['doctor@who.future']).From('The@Master.vilain')`
- `if Dalek or The Master or Wheeping Angel : run`

Pourquoi la stabilité ?

Pour pouvoir enchaîner des actions (... composer des fonctions !)

- `connexion = host + ":" + port + "/" + chemin`
- `email.To(['doctor@who.future']).From('The@Master.vilain')`
- `if Dalek or The Master or Wheeping Angel : run`

Définition 1 (Loi de composition interne)

Soit E un ensemble. Une loi de composition interne définie sur E est une **fonction totale** de $E \otimes E$ dans E.
On dit alors que le couple (E, loi) a une structure de **magma**.

Définition 2 (Stabilité)

Un ensemble E est stable par une loi $\dagger$ si, et seulement si :

$$\forall(x,y)\cdot\big((x,y)\in E\otimes E \Longrightarrow x\dagger y\in E\big)$$

Définition 1 (Loi de composition interne)

Soit E un ensemble. Une loi de composition interne définie sur E est une **fonction totale** de $E \otimes E$ dans E.
On dit alors que le couple (E, loi) a une structure de **magma**.

Définition 2 (Stabilité)

Un ensemble E est **stable** par une loi $\dagger$ si, et seulement si :

$$\forall (x, y) \cdot ((x, y) \in E \otimes E \Longrightarrow x \dagger y \in E)$$

Définition 1 (Loi de composition interne)

Soit E un ensemble. Une loi de composition interne définie sur E est une **fonction totale** de $E \otimes E$ dans E.
On dit alors que le couple (E, loi) a une structure de **magma**.

Définition 2 (Stabilité)

Un ensemble E est **stable** par une loi $\dagger$ si, et seulement si :

$$\forall (x, y) \cdot \big((x, y) \in E \otimes E \Longrightarrow x \dagger y \in E\big)$$

Deuxième commandement : NEUTRALITÉ

- `x + 0 = x`
- `x * 1 = x`
- `b and True = b`
- `b or False = b`
- `"Dalek" + "" = "Dalek"`
- `panier.ajoute_contenu(panier_vide) = panier`
- `f o (lambda x: x) = f`

Définition 3 (Élement neutre)

Soit $(E, \dagger)$ un magma. Un élément neutre de $(E, \dagger)$ est un élément e vérifiant :

$$\forall x \cdot (x \dagger e = e \dagger x = x)$$

Vous aurez bien noté le $\forall$...

Définition 3 (Élement neutre)

Soit $(E, \dagger)$ un magma. Un élément neutre de $(E, \dagger)$ est un élément e vérifiant :

$$\forall x \cdot (x \dagger e = e \dagger x = x)$$

Vous aurez bien noté le $\forall$...

Pourquoi la neutralité ?

Exemple : calculer la somme des éléments d'une liste.

```
def somme(xs) :
    som_actuelle = NEUTRE
    for x in xs :
        som_actuelle = som_actuelle + x
    return som_actuelle
```

Pourquoi la neutralité ?

Exemple : calculer la somme des éléments d'une liste.

```
def somme(xs) :
    som_actuelle = NEUTRE
    for x in xs :
        som_actuelle = som_actuelle + x
    return som_actuelle
```

Pourquoi la neutralité ?

Exemple : calculer le produit des éléments d'une liste.

```
def produit(xs) :
    prod_actuel = NEUTRE
    for x in xs :
        prod_actuel = prod_actuel * x
    return prod_actuel
```

Pourquoi la neutralité ?

Exemple : calculer le produit des éléments d'une liste.

```
def produit(xs) :
    prod_actuel = NEUTRE
    for x in xs :
        prod_actuel = prod_actuel * x
    return prod_actuel
```

Pourquoi la neutralité ?

Exemple : étant donnée une liste de booléens, sont-ils tous vrais ?

```
def tous_vrais(bs) :
    bool_actuel = NEUTRE
    for b in bs :
        bool_actuel = bool_actuel and b
    return bool_actuel
```

Pourquoi la neutralité ?

Exemple : étant donnée une liste de booléens, sont-ils tous vrais ?

```
def tous_vrais(bs) :
    bool_actuel = NEUTRE
    for b in bs :
        bool_actuel = bool_actuel and b
    return bool_actuel
```

Pourquoi la neutralité ?

Exemple : étant donnée une liste de caractères, concaténez-les en une unique chaîne.

```
def concat(cs) :
    chaine_actuelle = NEUTRE
    for c in cs :
        chaine_actuelle = chaine_actuelle + c
    return chaine_actuelle
```

Pourquoi la neutralité ?

Exemple : étant donnée une liste de caractères, concaténez-les en une unique chaîne.

```
def concat(cs) :
    chaine_actuelle = NEUTRE
    for c in cs :
        chaine_actuelle = chaine_actuelle + c
    return chaine_actuelle
```

Troisième commandement : ASSOCIATIVITÉ

- `1 + (2 + 3) = (1 + 2) + 3`
- `1 * (2 * 3) = (1 * 2) * 3`
- `True and (False and True) = (True and False) and True`
- `"Crush "+("your "+"ennemies") = ("Crush "+"your ")+"ennemies"`
- `3 - (2 - 1) != (3 - 2) - 1`

IUT NANTES Informatique

Troisième commandement : ASSOCIATIVITÉ

- `1 + (2 + 3) = (1 + 2) + 3`
- `1 * (2 * 3) = (1 * 2) * 3`
- `True and (False and True) = (True and False) and True`
- `"Crush "+("your "+"ennemies") = ("Crush "+"your ")+"ennemies"`
- `3 - (2 - 1) != (3 - 2) - 1`

Troisième commandement : ASSOCIATIVITÉ

- `1 + (2 + 3) = (1 + 2) + 3`
- `1 * (2 * 3) = (1 * 2) * 3`
- `True and (False and True) = (True and False) and True`
- `"Crush "+("your "+"ennemies") = ("Crush "+"your ")+"ennemies"`
- `3 - (2 - 1) != (3 - 2) - 1`

Troisième commandement : ASSOCIATIVITÉ

- `1 + (2 + 3) = (1 + 2) + 3`
- `1 * (2 * 3) = (1 * 2) * 3`
- `True and (False and True) = (True and False) and True`
- `"Crush "+("your "+"ennemies") = ("Crush "+"your ")+"ennemies"`
- `3 - (2 - 1) != (3 - 2) - 1`

Troisième commandement : ASSOCIATIVITÉ

- `1 + (2 + 3) = (1 + 2) + 3`
- `1 * (2 * 3) = (1 * 2) * 3`
- `True and (False and True) = (True and False) and True`
- `"Crush "+("your "+"ennemies") = ("Crush "+"your ")+"ennemies"`
- `3 - (2 - 1) != (3 - 2) - 1` ✗

Troisième commandement : ASSOCIATIVITÉ

- `1 + (2 + 3) = (1 + 2) + 3`
- `1 * (2 * 3) = (1 * 2) * 3`
- `True and (False and True) = (True and False) and True`
- `"Crush "+("your "+"ennemies") = ("Crush "+"your ")+"ennemies"`
- `3 - (2 - 1) != (3 - 2) - 1` ✗

Troisième commandement : ASSOCIATIVITÉ

- `1 + (2 + 3) = (1 + 2) + 3`
- `1 * (2 * 3) = (1 * 2) * 3`
- `True and (False and True) = (True and False) and True`
- `"Crush "+("your "+"ennemies") = ("Crush "+"your ")+"ennemies"`
- `3 - (2 - 1) != (3 - 2) - 1` ✗

Définition 4 (Monoïde)

Soit $(E, \dagger)$ un magma. Si de plus ce magma admet un élément neutre et si la loi $\dagger$ est associative sur E alors $(E, \dagger)$ a une structure de **monoïde**.

Pourquoi l'associativité ?

- Diviser pour régner
- Algorithme incrémental
- Parallélisation

Pourquoi l'associativité ?

- Diviser pour régner
- Algorithme incrémental
- Parallélisation

Pourquoi l'associativité ?

- Diviser pour régner
- Algorithme incrémental
- Parallélisation


L'ASSOCIATIVITÉ ?
JE T'EXPLIQUE !
memegen.fr

Batman t'explique l'associativité

Tu veux tuer cent méchants :

- je les tue l'un après l'autre
- j'en tue 10 aujourd'hui, 10 demain, et 80 après-demain
- avec Robin, Conan et sept autres copains on en tue dix chacun

Batman t'explique l'associativité

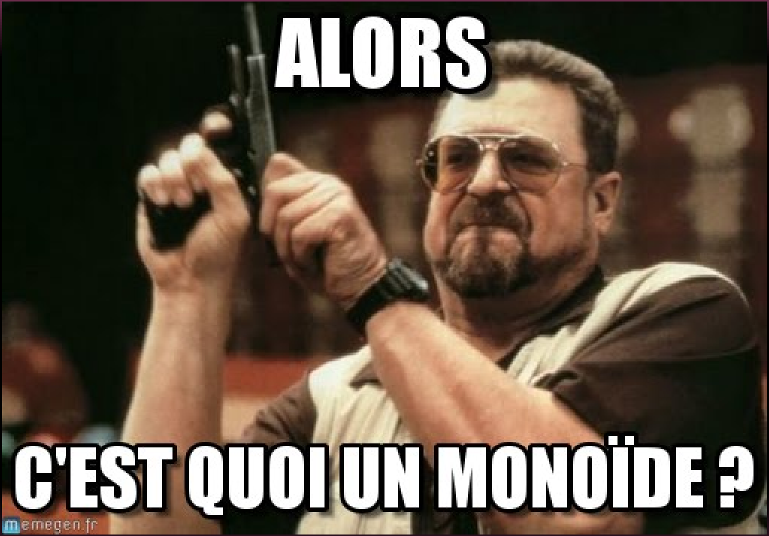
Tu veux tuer cent méchants :

- je les tue l'un après l'autre
- j'en tue 10 aujourd'hui, 10 demain, et 80 après-demain
- avec Robin, Conan et sept autres copains on en tue dix chacun

Batman t'explique l'associativité

Tu veux tuer cent méchants :

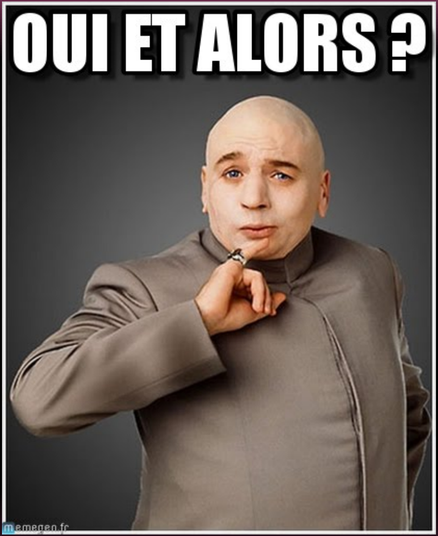
- je les tue l'un après l'autre
- j'en tue 10 aujourd'hui, 10 demain, et 80 après-demain
- avec Robin, Conan et sept autres copains on en tue dix chacun


ALORS
C'EST QUOI UN MONOÏDE ?
memegen.fr

1. Stabilité : un type T donne un type T
2. Neutre : une sorte de rien
3. Associativité : je groupe comme je veux

1. Stabilité : un type T donne un type T
2. Neutre : une sorte de rien
3. Associativité : je groupe comme je veux

1. Stabilité : un type T donne un type T
2. Neutre : une sorte de rien
3. Associativité : je groupe comme je veux


OUI ET ALORS ?
memegen.fr

Sommaire

1 Monoïdes
2 Map-Reduce
3 Groupe
- Le groupe (Z/nZ,+)

4 Anneaux et corps
- Anneaux
- Corps

5 Les Complexes


MAP-REDUCE
TU VOIS, C'EST SUPER SIMPLE
QUAND ON PENSE AUX MONOÏDES
memegen.com

La réduction, on a déjà vu...

```
def tous_vrais(bs : list bool) -> bool :
    bool_actuel = NEUTRE
    for b in bs :
        bool_actuel = bool_actuel and b
    return bool_actuel
```

```
def concat(cs : list car ) -> car :
    chaine_actuelle = NEUTRE
    for c in cs :
        chaine_actuelle = chaine_actuelle + c
    return chaine_actuelle
```

```
def reduit(collection : list t) -> t :
    reduc_actuelle = NEUTRE
    for element in collection :
        reduc_actuelle = reduc_actuelle LOI element
    return reduc_actuelle
```

```
reduce(LOI, collection, NEUTRE)
```


AVEC DES MONOÏDES
JE PEUX FAIRE MES COURSES SUR INTERNET
memegen.com
quickmeme.com

Mes courses : `panier = [1, 12, 99, 75, 10]`

Commandement 1 $1 + 12 + 99 + 75 + 10$ est un entier.

Commandement 2 0 est le neutre

Commandement 3 $((((1 + 12) + 99) + 75) + 10)$ peut-être différemment...

IUT NANTES Informatique

Mes courses : `panier = [1, 12, 99, 75, 10]`

Commandement 1 $1 + 12 + 99 + 75 + 10$ est un entier.

Commandement 2 0 est le neutre

Commandement 3 $((((1 + 12) + 99) + 75) + 10)$ peut-être différemment...

Le calcul peut être effectué par exemple...

... comme ceci : $((1 + 12) + (99 + 75)) + (10 + 0)$

... ou comme cela :

- serveur 1 : $1 + 12$

Le calcul peut être effectué par exemple...
... comme ceci : $((1+12)+(99+75))+(10+0)$
... ou comme cela :

serveur 1 $1+12$

serveur 2 $99+75$

serveur 3 $10+0$

Le calcul peut être effectué par exemple...
... comme ceci : $((1 + 12) + (99 + 75)) + (10 + 0)$
... ou comme cela :

serveur 1	$1 + 12$
serveur 2	$99 + 75$
serveur 3	$10 + 0$

Le calcul peut être effectué par exemple...
... comme ceci : $((1 + 12) + (99 + 75)) + (10 + 0)$
... ou comme cela :

serveur 1 $1 + 12$

serveur 2 $99 + 75$

serveur 3 $10 + 0$

Le calcul peut être effectué par exemple...
... comme ceci : $((1+12)+(99+75))+(10+0)$
... ou comme cela :

serveur 1	$1+12$
serveur 2	$99+75$
serveur 3	$10+0$

Oui mais en fait je n'ai pas un panier de prix mais un panier de produits :

`panier = [pizza, coca, finger, ketchup, casquette]`

On a une fonction `prix(produit) -> int`

Laissons la magie opérer

Oui mais en fait je n'ai pas un panier de prix mais un panier de produits :

`panier = [pizza, coca, finger, ketchup, casquette]`

On a une fonction `prix(produit) -> int`

Laissons la magie opérer

Oui mais en fait je n'ai pas un panier de prix mais un panier de produits :

`panier = [pizza, coca, finger, ketchup, casquette]`

On a une fonction `prix(produit) -> int`

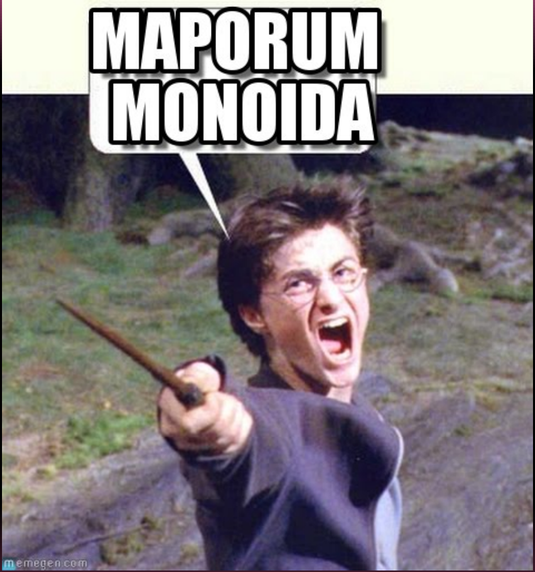
Laissons la magie opérer

Oui mais en fait je n'ai pas un panier de prix mais un panier de produits :

`panier = [pizza, coca, finger, ketchup, casquette]`

On a une fonction `prix(produit) -> int`

Laissons la magie opérer


MAPORUM MONOIDA
memegen.com

`map` va transformer chaque objet de notre panier en un prix.

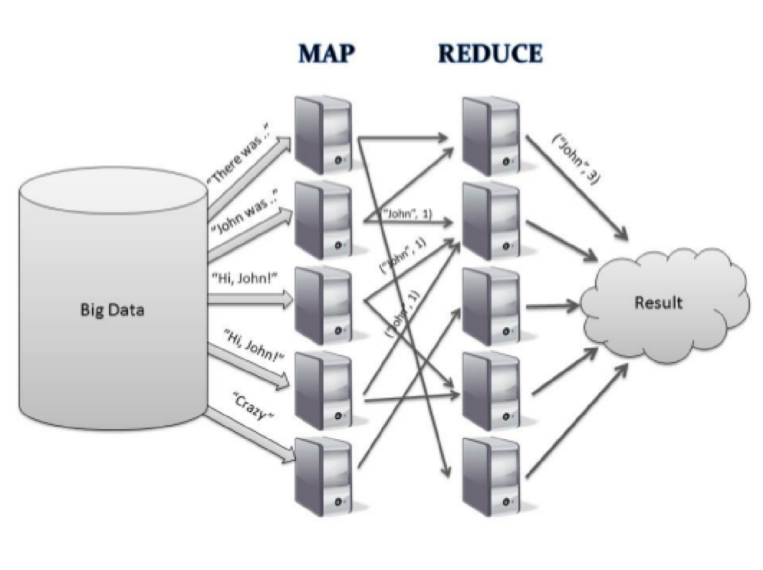
`map` nous permet de transférer nos objets minables dans un monoïde `map` va transformer chaque objet de notre panier en un prix.

`map` **nous permet de transférer nos objets minables dans un monoïde**


UTILISE LE MONOÏDE
memegen.com

- `panier = [pizza, biere, finger, ketchup, casquette]`
- `prix_panier = map(prix, panier)`
- `total = reduce(lambda x, y: x + y, prix_panier, 0)`


· MAP LA COLLECTION SUR UN MONOÏDE
· REDUCE LA COLLECTION OBTENUE
memegen.com


MAP
REDUCE
Big Data
"There was.."
"John was.."
"Hi, John!"
"Hi, John!"
"Crazy"
"John", 1)
("John", 1)
("John", 1)
("John", 3)
Result

Sommaire

1 Monoïdes
2 Map-Reduce
3 Groupe
- Le groupe (Z/nZ,+)

4 Anneaux et corps
- Anneaux
- Corps

5 Les Complexes


IL EST PARFOIS IMPOSSIBLE
DE REVENIR EN ARRIÈRE
memegen.com

On voudrait parfois revenir en arrière.

Par exemple on a rajouté un glaçon dans son whiskey...vite il faut l'enlever.

On voudrait parfois revenir en arrière.
Par exemple on a rajouté un glaçon dans son whiskey...vite il faut l'enlever.


PAS DE GLACE
DANS MON WHISKEY !
memegen.com

On a rajouté 2 à un entier :

$$x \longrightarrow x + 2$$

Oups, enlevons-le

$$x + 2 \longrightarrow x + 2 + (-2)$$

On a rajouté 2 à un entier :

$$x \longrightarrow x + 2$$

Oups, enlevons-le

$$x + 2 \longrightarrow x + 2 + (-2)$$

On a rajouté 2 à un entier :

$$x \longrightarrow x + 2$$

Oups, enlevons-le

$$x + 2 \longrightarrow x + 2 + (-2)$$

Définition 5 (Élément inversible)

Soit x un élément de E. Il est inversible (ou symétrisable) par $\star$ si, et seulement si,

$$(\exists y)(x \star y = y \star x = e_\star)$$

Pouvez-vous démontrer que si x admet un symétrique, alors ce symétrique est unique ?
On note souvent ce symétrique $x^{-1_\star}$ ou bien $-_\star x$.

Définition 5 (Élément inversible)

Soit x un élément de E. Il est inversible (ou symétrisable) par $\star$ si, et seulement si,

$$(\exists y)(x \star y = y \star x = e_\star)$$

Pouvez-vous démontrer que si x admet un symétrique, alors ce symétrique est unique ?
On note souvent ce symétrique $x^{-1_\star}$ ou bien $-_\star x$.

Définition 5 (Élément inversible)

Soit x un élément de E. Il est inversible (ou symétrisable) par $\star$ si, et seulement si,

$$(\exists y)(x \star y = y \star x = e_\star)$$

Pouvez-vous démontrer que si x admet un symétrique, alors ce symétrique est unique ?

On note souvent ce symétrique $x^{-1_\star}$ ou bien $-_\star x$.

Définition 5 (Élément inversible)

Soit x un élément de E. Il est inversible (ou symétrisable) par $\star$ si, et seulement si,

$$(\exists y)(x \star y = y \star x = e_\star)$$

Pouvez-vous démontrer que si x admet un symétrique, alors ce symétrique est unique ?
On note souvent ce symétrique $x^{-1_\star}$ ou bien $-_\star x$.

Des fois ça n'est pas possible...

On a rajouté une chaîne à une autre :

```
s = "Use the " + "Force"
```

Oups, je voulais mettre **`Monoid`** à la place de **`Force`**...

C'est trop tard : je ne peux rien concaténer à s pour revenir à `"Use the"`.

Des fois ça n'est pas possible...

On a rajouté une chaîne à une autre :

```
s = "Use the " + "Force"
```

Oups, je voulais mettre **Monoid** à la place de **Force**...

C'est trop tard : je ne peux rien concaténer à s pour revenir à "Use the".

Des fois ça n'est pas possible...

On a rajouté une chaîne à une autre :

```
s = "Use the " + "Force"
```

Oups, je voulais mettre `Monoid` à la place de `Force`...

C'est trop tard : je ne peux rien concaténer à `s` pour revenir à `"Use the"`.

IUT Nantes Informatique

Des fois ça n'est pas possible...

On a rajouté une chaîne à une autre :

`s = "Use the " + "Force"`

Oups, je voulais mettre **`Monoid`** à la place de **`Force`**...

C'est trop tard : je ne peux rien concaténer à `s` pour revenir à `"Use the"`.

Des fois ça n'est pas possible...

On a rajouté une chaîne à une autre :

`s = "Use the " + "Force"`

Oups, je voulais mettre `Monoid` à la place de `Force`...

C'est trop tard : je ne peux rien concaténer à `s` pour revenir à `"Use the"`.


INVERSIBLE ?
memegen.fr

Définition 6 (Groupe)

Un groupe est un monoïde tel que tout élément est inversible.


Evariste Galois
25.10.1811
31.05.1832

Sommaire

1 Monoïdes
2 Map-Reduce
3 **Groupe**
- Le groupe $(\mathbb{Z}/n\mathbb{Z},+)$
4 Anneaux et corps
- Anneaux
- Corps
5 Les Complexes

Définition 7 (Congruence des entiers)

Soit n un entier naturel non nul.
Deux éléments a et b du groupe $(\mathbb{Z}, +)$ sont **congrus modulo n** si, et seulement si, ils ont le même reste dans la division par n. On note

$$a \equiv b \pmod{n}$$

ou

$$a \equiv_n b$$

et on lit « *a est congru à b modulo n* ».

« Well, LISP seems to work okay now, modulo that GC bug. »

Définition 7 (Congruence des entiers)

Soit n un entier naturel non nul.
Deux éléments a et b du groupe $(\mathbb{Z}, +)$ sont **congrus modulo n** si, et seulement si, ils ont le même reste dans la division par n. On note

$$a \equiv b \pmod{n}$$

ou

$$a \equiv_n b$$

et on lit « *a est congru à b modulo n* ».

« Well, LISP seems to work okay now, modulo that GC bug. »
« I feel fine today modulo a slight headache. »

Définition 7 (Congruence des entiers)

Soit n un entier naturel non nul.
Deux éléments a et b du groupe $(\mathbb{Z}, +)$ sont **congrus modulo n** si, et seulement si, ils ont le même reste dans la division par n. On note

$$a \equiv b \pmod{n}$$

ou

$$a \equiv_n b$$

et on lit « *a est congru à b modulo n* ».

> *« Well, LISP seems to work okay now, modulo that GC bug. »*
> *« I feel fine today modulo a slight headache. »*
>
> *in « The New Hacker's Dictionnary »*
> *http://outpost9.com/reference/jargon/jargon_28.html#SEC35*

Définition 7 (Congruence des entiers)

Soit n un entier naturel non nul.
Deux éléments a et b du groupe $(\mathbb{Z}, +)$ sont **congrus modulo n** si, et seulement si, ils ont le même reste dans la division par n. On note

$$a \equiv b \pmod{n}$$

ou

$$a \equiv_n b$$

et on lit « *a est congru à b modulo n* ».

« Well, LISP seems to work okay now, modulo that GC bug. »
« I feel fine today modulo a slight headache. »

in « The New Hacker's Dictionnary »
http://outpost9.com/reference/jargon/jargon_28.html#SEC35

Définition 7 (Congruence des entiers)

Soit n un entier naturel non nul.
Deux éléments a et b du groupe $(\mathbb{Z}, +)$ sont **congrus modulo** n si, et seulement si, ils ont le même reste dans la division par n. On note

$$a \equiv b \pmod{n}$$

ou

$$a \equiv_n b$$

et on lit « *a est congru à b modulo n* ».

> « *Well, LISP seems to work okay now, modulo that GC bug.* »
> « *I feel fine today modulo a slight headache.* »

in « The New Hacker's Dictionnary »
http://outpost9.com/reference/jargon/jargon_28.html#SEC35

IUT Nantes Informatique

Définition 7 (Congruence des entiers)

Soit n un entier naturel non nul.
Deux éléments a et b du groupe $(\mathbb{Z}, +)$ sont **congrus modulo** n si, et seulement si, ils ont le même reste dans la division par n. On note

$$a \equiv b \pmod{n}$$

ou

$$a \equiv_n b$$

et on lit « *a est congru à b modulo n* ».

« Well, LISP seems to work okay now, modulo that GC bug. »
« I feel fine today modulo a slight headache. »

in « The New Hacker's Dictionnary »
http://outpost9.com/reference/jargon/jargon_28.html#SEC35

Théorème 8 (Division euclidienne)

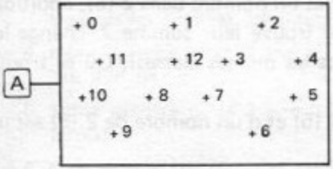
Soit a un entier relatif et b un entier naturel non nul.
Il existe un unique couple d'entiers (q, r) tels que

$$a = bq + r \quad \text{avec} \quad 0 \leqslant r < b$$

Déterminer q et r, c'est effectuer la division euclidienne de a par b.
On appelle a le dividende, b le diviseur, q le quotient et r le reste.

I Restes dans la division par 5

1. Effectue les divisions de 21 par 5 et 46 par 5.
Tu trouves le même reste : 1
Nous dirons que 46 donne dans la division par 5, le même reste que 21. A est l'ensemble des nombres de zéro à 12. Trace les traits fléchés signifiant "... donne le même reste, dans la division par 5 que ..."
Recommence le schéma de façon que les flèches ne se coupent pas.



Est-ce possible ? Quel classement peux-tu effectuer dans l'ensemble A ?

2. Classons les nombres entiers.

Comme nous venons de classer les nombres entiers de zéro à 12, nous allons essayer de classer tous les nombres entiers.

Quels sont les restes possibles dans la division d'un nombre par 5 ?

Traçons 5 colonnes. Nous mettrons dans la première les nombres qui donnent pour reste 0, dans le second ceux qui donnent pour reste 1 et ainsi de suite ...

restes	0	1	2	3	4
	0	1	2	3	4
nombres	5	6	7	8	9
	10	11	12	13	14
	..	..	..	..	..

Continue à placer les nombres suivants de la même manière (par exemple jusqu'à 30).

Quels sont les nombres de la 1 ère colonne ? de la 2ème ? la 3 ème ?

Fais la différence entre deux nombres quelconques de la même colonne. Qu'observes-tu ?

3. Notation et vocabulaire.

André, Jean et Pierre font partie de la même équipe. Pour désigner cette équipe on peut aussi bien dire : équipe d'André, équipe de Jean ou bien équipe de Pierre.

L'ensemble des nombres de la 1ère colonne sera appelé la classe de zéro par 5 et noté $\dot{0}\ (5)$, ou bien classe de 5 par 5 et noté $\dot{5}(5)$, ou bien classe de tout autre nombre de la colonne $\dot{10}\ (5)$, $\dot{15}(5)$ ...) :

Egalité : $\dot{0}\ (5)$, $\dot{5}\ (5)$ désignent le même ensemble, tu peux écrire : $\dot{0}(5) = \dot{5}\ (5)$ ou bien, plus simplement $\dot{0} = \dot{5}\ (5)$.

De la même manière les nombres de la deuxième colonne constituent : $\dot{1}\ (5)$ ou bien $\dot{6}\ (5)$, ... et $\dot{1} = \dot{6} = \dot{11}\ (5)$.

II Addition des classes

Prends un nombre dans $\dot{1}$ (5) et un nombre dans $\dot{2}$ (5), additionne ces deux nombres. Où se trouve leur somme ? change les nombres (en les prenant dans les mêmes classes). Où se trouve la nouvelle somme ?

La somme d'un nombre de $\dot{1}$ (5) et d'un nombre de $\dot{2}$ (5) est un nombre de $\dot{3}$ (5)

Nous écrivons ceci $\dot{1} \dot{+} \dot{2} = \dot{3}$ (5) (nous mettons un point sur +, car ce n'est pas l'addition ordinaire et nous lisons un pointé plus deux pointé égale trois pointé).

Complète les égalités : $\dot{0} \dot{+} \dot{2} = \square$; $\dot{1} \dot{+} \dot{3} = \square$; $\dot{3} \dot{+} \dot{2} = \square$

Informatique

Pour donner tous les résultats on dresse une table d'addition.

Complète la avec les notations les plus simples.

Remarque. Effectue $4+3$; peux-tu écrire $\dot{4}\,\dot{+}\,\dot{3} = \dot{7}$?

$\dot{4}\,\dot{+}\,\dot{4}$; peux-tu écrire $4+4 = 3$?

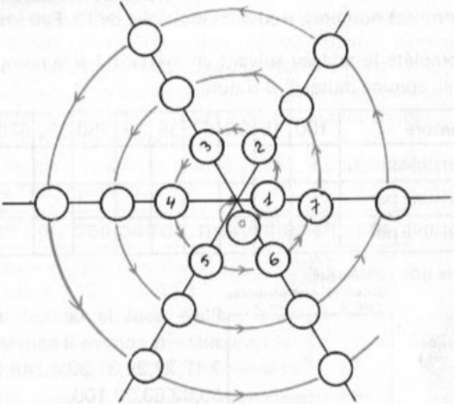
Il ne faut pas confondre l'addition des classes avec celle des entiers.

Multiplication des classes : ce que tu viens de faire pour l'addition, tu peux le faire de la même manière pour la multiplication et dresser une table de multiplication.

$\dot{+}$	$\dot{0}$	$\dot{1}$	$\dot{2}$	$\dot{3}$	$\dot{4}$
$\dot{0}$			$\dot{2}$		
$\dot{1}$			$\dot{3}$	$\dot{4}$	
$\dot{2}$	$\dot{2}$	$\dot{3}$		$\dot{0}$	
$\dot{3}$		$\dot{4}$	$\dot{0}$		
$\dot{4}$					

I Classement

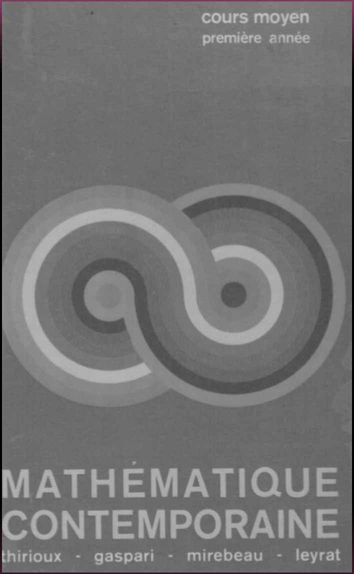
1.



Ecris la suite des nombres en suivant le sens des flèches.

Place ensuite les nombres de chaque demi-ligne droite sur ce tableau. Les nombres naturels ne sont-ils pas ainsi classés ?

0	1	2	3	4	5
6	7				


cours moyen
première année
MATHÉMATIQUE
CONTEMPORAINE
thirioux - gaspari - mirebeau - leyrat

On note $\mathbb{Z}/_{n\mathbb{Z}}$ l'ensemble des classes modulo n.
Par exemple, on notera :

$$\mathbb{Z}/_{8\mathbb{Z}} = \left\{\overline{0}^8, \overline{1}^8, \overline{2}^8, \overline{3}^8, \overline{4}^8, \overline{5}^8, \overline{6}^8, \overline{7}^8\right\}$$

iut NANTES Informatique

On note $\mathbb{Z}/_{n\mathbb{Z}}$ l'ensemble des classes modulo n.
Par exemple, on notera :

$$\mathbb{Z}/_{8\mathbb{Z}} = \left\{\overline{0}^8, \overline{1}^8, \overline{2}^8, \overline{3}^8, \overline{4}^8, \overline{5}^8, \overline{6}^8, \overline{7}^8\right\}$$

Attention !

Notez bien que $\overline{0}^8$, $\overline{1}^8$, etc. sont des ensembles d'entiers !...

Ainsi $\overline{0}^8 = \{\ldots, -16, -8, 0, 8, 16, 24, 32, \ldots\}$

Attention !

Notez bien que $\overline{0}^8$, $\overline{1}^8$, etc. sont des ensembles d'entiers !...

Ainsi $\overline{0}^8 = \{\ldots, -16, -8, 0, 8, 16, 24, 32, \ldots\}$

Attention !

Notez bien que $\overline{0}^8$, $\overline{1}^8$, etc. sont des ensembles d'entiers !...

Ainsi $\overline{0}^8 = \{\ldots, -16, -8, 0, 8, 16, 24, 32, \ldots\}$

Exercice 1

Écrivez par exemple les lois d'addition et de multiplication de $\mathbb{Z}/_{7\mathbb{Z}}$ *et* $\mathbb{Z}/_{8\mathbb{Z}}$ *: des remarques ?*

Est-ce que $(\mathbb{Z}/_{n\mathbb{Z}}, +)$ *et* $(\mathbb{Z}/_{n\mathbb{Z}}, \cdot)$ *sont des groupes ?*

Exercice 1

Écrivez par exemple les lois d'addition et de multiplication de $\mathbb{Z}/_{7\mathbb{Z}}$ *et* $\mathbb{Z}/_{8\mathbb{Z}}$ *: des remarques ?*
Est-ce que $(\mathbb{Z}/_{n\mathbb{Z}}, +)$ *et* $(\mathbb{Z}/_{n\mathbb{Z}}, \cdot)$ *sont des groupes ?*

Exercice 1

Écrivez par exemple les lois d'addition et de multiplication de $\mathbb{Z}/_{7\mathbb{Z}}$ *et* $\mathbb{Z}/_{8\mathbb{Z}}$ *: des remarques ?*
Est-ce que $(\mathbb{Z}/_{n\mathbb{Z}}, +)$ *et* $(\mathbb{Z}/_{n\mathbb{Z}}, \cdot)$ *sont des groupes ?*

PGCD

Définition 9 (PGCD)

Soit a et b deux entiers. On appelle PGCD de a et b et on note $a \wedge b$ l'entier défini de la manière suivante :

- $0 \wedge 0 = 0$
- Si a et b ne sont pas simultanément nuls, $a \wedge b$ est le plus grand entier naturel qui divise simultanément a et b.

$$D(a) \cap D(b)$$

PGCD

Définition 9 (PGCD)

Soit a et b deux entiers. On appelle PGCD de a et b et on note $a \wedge b$ l'entier défini de la manière suivante :

- $0 \wedge 0 = 0$
- Si a et b ne sont pas simultanément nuls, $a \wedge b$ est le plus grand entier naturel qui divise simultanément a et b.

$$D(a) \cap D(b)$$

PGCD

Définition 9 (PGCD)

Soit a et b deux entiers. On appelle PGCD de a et b et on note $a \wedge b$ l'entier défini de la manière suivante :

- $0 \wedge 0 = 0$
- Si a et b ne sont pas simultanément nuls, $a \wedge b$ est le plus grand entier naturel qui divise simultanément a et b.

$$D(a) \cap D(b)$$

PGCD

Définition 9 (PGCD)

Soit a et b deux entiers. On appelle PGCD de a et b et on note $a \wedge b$ l'entier défini de la manière suivante :

- $0 \wedge 0 = 0$
- Si a et b ne sont pas simultanément nuls, $a \wedge b$ est le plus grand entier naturel qui divise simultanément a et b.

$$D(a) \cap D(b)$$

IUT NANTES Informatique

PGCD

Définition 9 (PGCD)

Soit a et b deux entiers. On appelle PGCD de a et b et on note $a \wedge b$ l'entier défini de la manière suivante :

- $0 \wedge 0 = 0$
- Si a et b ne sont pas simultanément nuls, $a \wedge b$ est le plus grand entier naturel qui divise simultanément a et b.

$$D(a) \cap D(b)$$

iut NANTES Informatique

Algorithme d'Euclide I

```
Fonction euclide( a, b : entiers ) : entier
Si b = 0 Alors
  | Retourner a
Sinon
  | Retourner euclide(b,rem(a,b))
FinSi
```

Algorithme d'Euclide II

k	u_k	v_k	r_k	q_k
0	1	0	$r_0 = a$	/
1	0	1	$r_1 = b$	q_1
2	$u_0 - u_1 q_1$	$v_0 - v_1 q_1$	$r_2 = r_0 - r_1 q_1$	q_2
3	$u_1 - u_2 q_2$	$v_1 - v_2 q_2$	$r_3 = r_1 - r_2 q_2$	q_3
$\vdots$			$\vdots$	$\vdots$
$p-1$			$r_{p-1} = a \wedge b$	q_{p-1}
p			$r_p = 0$	

k	u_k	v_k	r_k	q_k	
0	1	0	19	/	L_0
1	0	1	15	1	L_1
2	1	-1	4	3	$L_2 \leftarrow L_0 - \boxed{1} \times L_1$
3	-3	4	3	1	$L_3 \leftarrow L_1 - 3 \times L_2$
4	4	-5	1	3	$L_4 \leftarrow L_2 - 1 \times L_3$
5			0		$L_5 \leftarrow L_3 - 3 \times L_4$

Sommaire

1 Monoïdes
2 Map-Reduce
3 Groupe
- Le groupe (Z/nZ,+)

4 Anneaux et corps
- Anneaux
- Corps

5 Les Complexes

Sommaire

1 Monoïdes
2 Map-Reduce
3 Groupe
- Le groupe (Z/nZ,+)

4 Anneaux et corps
- Anneaux
- Corps

5 Les Complexes


I DON'T WANNA SAY IT'S HOT
IN MY ROOM...
BUT TWO HOBBITS JUST CAME
AROUND AND THREW A RING
IN IT

Définition 10 (Anneau)

Soit A un ensemble muni de deux lois $\boxdot$ et $\boxplus$.

On dit que $\langle A, \boxplus, \boxdot \rangle$ est un anneau si, et seulement si :

- $\langle A, \boxplus \rangle$ est un groupe commutatif ;
- $\boxdot$ est associative ;
- $\boxdot$ est distributive sur $\boxplus$.

Si $\boxdot$ est commutative, alors l'anneau est dit *commutatif*.
Si $\boxdot$ admet un élément neutre, l'anneau est dit *unitaire*.
Les éléments d'un anneau unitaire qui admettent un symétrique par $\boxdot$ sont dits *inversibles*. On note A^* l'ensemble des éléments inversibles de A.

IUT Nantes Informatique

Définition 10 (Anneau)

Soit A un ensemble muni de deux lois $\boxdot$ et $\boxplus$.

On dit que $\langle A, \boxplus, \boxdot \rangle$ est un anneau si, et seulement si :

- $\langle A, \boxplus \rangle$ est un groupe commutatif ;
- $\boxdot$ est associative ;
- $\boxdot$ est distributive sur $\boxplus$.

Si $\boxdot$ est commutative, alors l'anneau est dit *commutatif*.
Si $\boxdot$ admet un élément neutre, l'anneau est dit *unitaire*.
Les éléments d'un anneau unitaire qui admettent un symétrique par $\boxdot$ sont dits *inversibles*. On note A^* l'ensemble des éléments inversibles de A.

Définition 10 (Anneau)

Soit A un ensemble muni de deux lois $\boxdot$ et $\boxplus$.

On dit que $\langle A, \boxplus, \boxdot \rangle$ est un anneau si, et seulement si :

- $\langle A, \boxplus \rangle$ est un groupe commutatif ;
- $\boxdot$ est associative ;
- $\boxdot$ est distributive sur $\boxplus$.

Si $\boxdot$ est commutative, alors l'anneau est dit *commutatif*.
Si $\boxdot$ admet un élément neutre, l'anneau est dit *unitaire*.
Les éléments d'un anneau unitaire qui admettent un symétrique par $\boxdot$ sont dits *inversibles*. On note A^* l'ensemble des éléments inversibles de A.

Définition 10 (Anneau)

Soit A un ensemble muni de deux lois $\boxdot$ et $\boxplus$.

On dit que $\langle A, \boxplus, \boxdot \rangle$ est un anneau si, et seulement si :

- $\langle A, \boxplus \rangle$ est un groupe commutatif ;
- $\boxdot$ est associative ;
- $\boxdot$ est distributive sur $\boxplus$.

Si $\boxdot$ est commutative, alors l'anneau est dit *commutatif*.
Si $\boxdot$ admet un élément neutre, l'anneau est dit *unitaire*.
Les éléments d'un anneau unitaire qui admettent un symétrique par $\boxdot$ sont dits *inversibles*. On note A^* l'ensemble des éléments inversibles de A.

Définition 10 (Anneau)

Soit A un ensemble muni de deux lois $\boxdot$ et $\boxplus$.

On dit que $\langle A, \boxplus, \boxdot \rangle$ est un anneau si, et seulement si :

- $\langle A, \boxplus \rangle$ est un groupe commutatif ;
- $\boxdot$ est associative ;
- $\boxdot$ est distributive sur $\boxplus$.

Si $\boxdot$ est commutative, alors l'anneau est dit *commutatif*.
Si $\boxdot$ admet un élément neutre, l'anneau est dit *unitaire*.
Les éléments d'un anneau unitaire qui admettent un symétrique par $\boxdot$ sont dits *inversibles*. On note A^* l'ensemble des éléments inversibles de A.

Définition 10 (Anneau)

Soit A un ensemble muni de deux lois $\boxdot$ et $\boxplus$.

On dit que $\langle A, \boxplus, \boxdot \rangle$ est un anneau si, et seulement si :

- $\langle A, \boxplus \rangle$ est un groupe commutatif ;
- $\boxdot$ est associative ;
- $\boxdot$ est distributive sur $\boxplus$.

Si $\boxdot$ est commutative, alors l'anneau est dit *commutatif*.
Si $\boxdot$ admet un élément neutre, l'anneau est dit *unitaire*.
Les éléments d'un anneau unitaire qui admettent un symétrique par $\boxdot$ sont dits *inversibles*. On note A^* l'ensemble des éléments inversibles de A.

Définition 10 (Anneau)

Soit A un ensemble muni de deux lois $\boxdot$ et $\boxplus$.

On dit que $\langle A, \boxplus, \boxdot \rangle$ est un anneau si, et seulement si :

- $\langle A, \boxplus \rangle$ est un groupe commutatif ;
- $\boxdot$ est associative ;
- $\boxdot$ est distributive sur $\boxplus$.

Si $\boxdot$ est commutative, alors l'anneau est dit *commutatif*.
Si $\boxdot$ admet un élément neutre, l'anneau est dit *unitaire*.
Les éléments d'un anneau unitaire qui admettent un symétrique par $\boxdot$ sont dits *inversibles*. On note A^* l'ensemble des éléments inversibles de A.

IUT Nantes Informatique

Définition 10 (Anneau)

Soit A un ensemble muni de deux lois $\boxdot$ et $\boxplus$.

On dit que $\langle A, \boxplus, \boxdot \rangle$ est un anneau si, et seulement si :

- $\langle A, \boxplus \rangle$ est un groupe commutatif ;
- $\boxdot$ est associative ;
- $\boxdot$ est distributive sur $\boxplus$.

Si $\boxdot$ est commutative, alors l'anneau est dit *commutatif*.
Si $\boxdot$ admet un élément neutre, l'anneau est dit *unitaire*.
Les éléments d'un anneau unitaire qui admettent un symétrique par $\boxdot$ sont dits *inversibles*. On note A^* l'ensemble des éléments inversibles de A.

Sommaire

1 Monoïdes
2 Map-Reduce
3 Groupe
- Le groupe (Z/nZ,+)

4 Anneaux et corps
- Anneaux
- Corps

5 Les Complexes

Définition 11 (Corps)

Soit $\mathbb{K}$ un ensemble muni de deux LCI $\boxplus$ et $\boxdot$. Le triplet $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ possède une structure de **corps** si, et seulement si,

- $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ a une structure d'anneau unitaire ;
- $\langle \mathbb{K} \setminus \{e_{\boxplus}\}, \boxdot \rangle$ a une structure de groupe.

Combien d'éléments a au minimum un corps ?

Définition 11 (Corps)

Soit $\mathbb{K}$ un ensemble muni de deux LCI $\boxplus$ et $\boxdot$. Le triplet $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ possède une structure de **corps** si, et seulement si,

- $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ a une structure d'anneau unitaire ;
- $\langle \mathbb{K} \setminus \{e_{\boxplus}\}, \boxdot \rangle$ a une structure de groupe.

Combien d'éléments a au minimum un corps ?

Définition 11 (Corps)

Soit $\mathbb{K}$ un ensemble muni de deux LCI $\boxplus$ et $\boxdot$. Le triplet $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ possède une structure de **corps** si, et seulement si,

- $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ a une structure d'anneau unitaire ;
- $\langle \mathbb{K} \smallsetminus \{e_{\boxplus}\}, \boxdot \rangle$ a une structure de groupe.

Combien d'éléments a au minimum un corps ?

Définition 11 (Corps)

Soit $\mathbb{K}$ un ensemble muni de deux LCI $\boxplus$ et $\boxdot$. Le triplet $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ possède une structure de **corps** si, et seulement si,

- $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ a une structure d'anneau unitaire ;
- $\langle \mathbb{K} \setminus \{e_{\boxplus}\}, \boxdot \rangle$ a une structure de groupe.

Combien d'éléments a au minimum un corps ?

Définition 11 (Corps)

Soit $\mathbb{K}$ un ensemble muni de deux LCI $\boxplus$ et $\boxdot$. Le triplet $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ possède une structure de **corps** si, et seulement si,

- $\langle \mathbb{K}, \boxplus, \boxdot \rangle$ a une structure d'anneau unitaire ;
- $\langle \mathbb{K} \setminus \{e_{\boxplus}\}, \boxdot \rangle$ a une structure de groupe.

Combien d'éléments a au minimum un corps ?

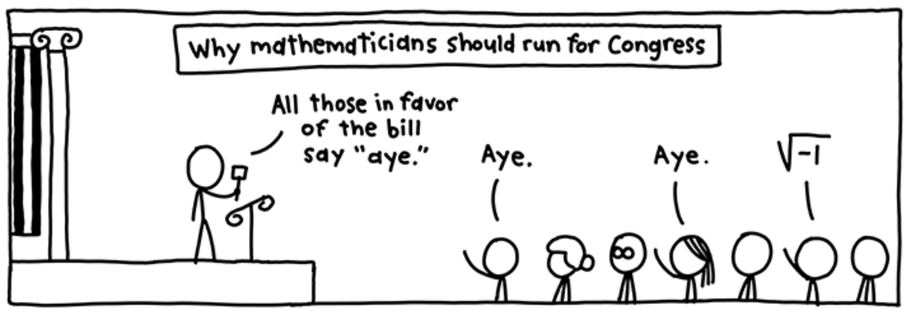
Le corps $\mathbb{F}_2$ ($\mathbb{F}$ comme Field) est $\mathbb{Z}/2\mathbb{Z}$ muni de deux opérations : lesquelles à votre avis ?

Le corps $\mathbb{F}_2$ ($\mathbb{F}$ comme Field) est $\mathbb{Z}/2\mathbb{Z}$ muni de deux opérations : lesquelles à votre avis ?

Le corps $\mathbb{F}_2$ ($\mathbb{F}$ comme Field) est $\mathbb{Z}/_{2\mathbb{Z}}$ muni de deux opérations : lesquelles à votre avis ?

Sommaire

1 Monoïdes
2 Map-Reduce
3 Groupe
- Le groupe (Z/nZ,+)
4 Anneaux et corps
- Anneaux
- Corps
5 Les Complexes



Why mathematicians should run for Congress
All those in favor of the bill say "aye."
Aye.
Aye.
$\sqrt{-1}$

ℂ comme...Corps des Complexes.

Quel est l'inverse de $x + iy$ pour la multiplication ?

```
In [68]: 1/(1+1j)
Out[68]: (0.5-0.5j)
```

$\mathbb{C}$ comme...Corps des Complexes.

Quel est l'inverse de $x + iy$ pour la multiplication ?

```
In [68]: 1/(1+1j)
Out[68]: (0.5-0.5j)
```

$\mathbb{C}$ comme...Corps des Complexes.
Quel est l'inverse de $x + iy$ pour la multiplication ?

```
In [68]: 1/(1+1j)
Out[68]: (0.5-0.5j)
```

$\mathbb{C}$ comme...Corps des Complexes.
Quel est l'inverse de $x + iy$ pour la multiplication ?

```
In [68]: 1/(1+1j)
Out[68]: (0.5-0.5j)
```

Définition 12 (Morphisme)

Soit $\langle E, \oplus, \otimes \rangle$ et $\langle F, \dagger, \star \rangle$ deux corps et φ une fonction totale de E dans F. On dit que φ est un morphisme de corps de E dans F si, et seulement si :

$$(\forall x)(\forall y)\big(\varphi(x \oplus y) = \varphi(x) \dagger \varphi(y)\big)$$

$$(\forall x)(\forall y)\big(\varphi(x \otimes y) = \varphi(x) \star \varphi(y)\big)$$

Si φ est bijective, on parle d'ISOMORPHISME

Définition 12 (Morphisme)

Soit $\langle E, \oplus, \otimes\rangle$ et $\langle F, \dagger, \star\rangle$ deux corps et φ une fonction totale de E dans F. On dit que φ est un morphisme de corps de E dans F si, et seulement si :

$$\left(\forall x\right)\left(\forall y\right)\Big(\varphi(x \oplus y) = \varphi(x) \dagger \varphi(y)\Big)$$

$$\left(\forall x\right)\left(\forall y\right)\Big(\varphi(x \otimes y) = \varphi(x) \star \varphi(y)\Big)$$

Si φ est bijective, on parle d'ISOMORPHISME

IUT Nantes Informatique

Définition 12 (Morphisme)

Soit $\langle E, \oplus, \otimes \rangle$ et $\langle F, \dagger, \star \rangle$ deux corps et φ une fonction totale de E dans F. On dit que φ est un morphisme de corps de E dans F si, et seulement si :

$$(\forall x)(\forall y)\big(\varphi(x \oplus y) = \varphi(x) \dagger \varphi(y)\big)$$

$$(\forall x)(\forall y)\big(\varphi(x \otimes y) = \varphi(x) \star \varphi(y)\big)$$

Si φ est bijective, on parle d'ISOMORPHISME

Imaginez un isomorphisme entre $\mathbb{C}$ et $\mathbb{R}^2$, l'ensemble des coordonnées des points de l'écran...
Quel est le plus simple à manipuler ?

Imaginez un isomorphisme entre $\mathbb{C}$ et $\mathbb{R}^2$, l'ensemble des coordonnées des points de l'écran...
Quel est le plus simple à manipuler ?

```python
with open(hpath, 'w') as h:
        h.writelines(
            ['<!DOCTYPE html>\n'
            ,'<head>\n'
            ,'<title>plot</title>\n'
            ,'</head>\n'
            ,'<body>\n'
            ,'<svg height="720" width=920 xmlns="http://www.w3.org/2000/svg">\n'
            ,'<line x1="0" y1="360" x2="720" y2="360"'
             ,'style="stroke:rgb(150,150,150);stroke-width:2"/>\n'
            ,'<line x1="360" y1="0" x2="360" y2="720"'
            ,'style="stroke:rgb(150,150,150);stroke-width:2"/>\n'])
        for pt in L:
            if isinstance(pt, Number):
                x,y = pt.real, pt.imag
            else:
                if isinstance(pt, tuple) or isinstance(pt, list):
                    x,y = pt
                else:
                    raise ValueError
            h.writelines(['<circle cx="%d" cy="%d" r="%d" fill="white"/>\n'
                          % (origin[0]+scalar*x,origin[1]-scalar*y,dot_size)])
        h.writelines(['</svg>\n</body>\n</html>'])
```

Merci à Philip Doctor