

Le package `tnsarith` : un peu d'arithmétique élémentaire

Code source disponible sur <https://github.com/typensee-latex/tnsarith.git>.

Version 0.1.0-beta développée et testée sur Mac OS X.

Christophe BAL

2020-07-12

Table des matières

1	Introduction	3
2	Ensembles classiques	3
3	Opérateurs de base	3
4	Fonctions nommées spéciales	3
4.1	Sans paramètre	3
4.2	Toutes les fonctions nommées en plus	3
5	Fractions continuées	3
5.1	Fractions continuées standard	3
5.2	Fractions continuées généralisées	4
5.3	Comme une fraction continuée isolée	4
5.4	L'opérateur \mathcal{K}	4
6	Historique	6
7	Toutes les fiches techniques	7
7.1	Introduction	7
7.2	Ensembles classiques	7
7.3	Opérateurs de base	7
7.3.1	Arithmétique – Opérateurs de base	7
7.4	Fonctions nommées spéciales	7
7.4.1	Fonctions nommées sans paramètre	7
7.5	Fractions continuées	7
7.5.1	Fractions continuées standard	7
7.5.2	Fractions continuées généralisées	7
7.5.3	Comme une fraction continuée isolée	7
7.5.4	Fractions continuées – L'opérateur \mathcal{K}	7

1 Introduction

Le package `tnsarith` propose quelques macros pour rédiger un peu d'arithmétique via un codage sémantique simple.

2 Ensembles classiques

Le package `tnssets` propose les macros `\NN`, `\ZZ`, `\QQ` ainsi que `\PP` pour indiquer l'ensemble des naturels, celui des entiers relatifs, celui des fractions rationnelles et enfin celui des nombres premiers. Se rendre sur <https://github.com/typensee-latex/tnssets.git> si cela vous intéresse.

3 Opérateurs de base

Pour des raisons d'expressivité des codes \LaTeX , les opérateurs binaires `\divides`, `\ndivides` et `\modulo` ont été ajoutés comme alias respectifs de `\mid`, `\nmid` et `\bmod` qui sont proposés par le package `amssymb`. Un opérateur `\nequiv` a été aussi ajouté.

```
$10 \divides 150$ au lieu de  
$10 \mid 150$
```

```
$10 \ndivides 154$ au lieu de  
$10 \not\mid 154$
```

```
$a \nequiv b \modulo p  
\iff  
p \ndivides (a - b)$.
```

$10 \mid 150$ au lieu de $10|150$
 $10 \nmid 154$ au lieu de $10 \not\mid 154$
 $a \not\equiv b \bmod p \iff p \nmid (a - b).$

4 Fonctions nommées spéciales

4.1 Sans paramètre

Deux fonctions nommées `\pgcd` et `\ppcm` utiles au francophone ont été ajoutées ainsi que la fonction `\lcm` pour les anglophones car cette dernière n'est pas disponible par défaut. La liste complète des fonctions nommées est donnée ci-dessous dans la section 4.2.

```
$\pgcd x = \gcd x$ et $\ppcm x = \lcm x$
```

$\text{pgcd } x = \gcd x$ et $\text{ppcm } x = \text{lcm } x$

4.2 Toutes les fonctions nommées en plus

`pgcd` : `pgcd...`
`ppcm` : `ppcm...`

`lcm` : `lcm...`

5 Fractions continuées

5.1 Fractions continuées standard

Dans l'exemple suivant, la notation en ligne semble être due à Alfred Pringsheim. La notation à gauche utilise toujours le maximum d'espace pour améliorer la lisibilité.

```


$$\frac{u_0}{u_1 + \frac{1}{u_2 + \frac{1}{\dots + \frac{1}{u_n}}}}$$


```

$$u_0 + \frac{1}{u_1 + \frac{1}{u_2 + \frac{1}{\dots + \frac{1}{u_n}}}} = u_0 + \left\lfloor \frac{1}{u_1} \right\rfloor + \left\lfloor \frac{1}{u_2} \right\rfloor + \left\lfloor \frac{1}{\dots} \right\rfloor + \left\lfloor \frac{1}{u_n} \right\rfloor$$

5.2 Fractions continuées généralisées

Voici comment écrire une fraction continuée généralisée.

```


$$a + \frac{b}{c + \frac{d}{e + \frac{f}{\dots + \frac{y}{z}}}}$$


```

$$a + \frac{b}{c + \frac{d}{e + \frac{f}{\dots + \frac{y}{z}}}} = a + \left\lfloor \frac{b}{c} \right\rfloor + \left\lfloor \frac{d}{e} \right\rfloor + \left\lfloor \frac{f}{\dots} \right\rfloor + \left\lfloor \frac{y}{z} \right\rfloor$$

5.3 Comme une fraction continuée isolée

La raison d'être de la macro ci-dessous vient juste de son usage en interne.

```


$$\frac{a}{b}$$


```

$\left\lfloor \frac{a}{b} \right\rfloor$ pour les fous... :-)

5.4 L'opérateur \mathcal{K}

Exemple 1

La notation suivante est proche de celle qu'utilisait Carl Friedrich Gauss.

```


$$\mathcal{K}_{k=1}^n(b_k : c_k) = \frac{b_1}{c_1 + \frac{b_2}{c_2 + \frac{b_3}{\dots + \frac{b_n}{c_n}}}}$$


```

$$\mathcal{K}_{k=1}^n(b_k : c_k) = \frac{b_1}{c_1 + \frac{b_2}{c_2 + \frac{b_3}{\dots + \frac{b_n}{c_n}}}}$$

Remarque. La lettre \mathcal{K} vient de "kettenbruch" qui signifie "fraction continuée" en allemand.

Exemple 2

```

$$u_0 + \cfrac{\cfrac{1}{u_1 + \cfrac{1}{u_2 + \cfrac{1}{\dots + \cfrac{1}{u_n}}}}}{1 : u_k}$$

```

$$u_0 + \mathcal{K}_{k=1}^n(1 : u_k) = u_0 + \frac{1}{u_1 + \frac{1}{u_2 + \frac{1}{\dots + \frac{1}{u_n}}}}$$

6 Historique

Nous ne donnons ici qu'un très bref historique récent ¹ de `tnsarith` à destination de l'utilisateur principalement. Tous les changements sont disponibles uniquement en anglais dans le dossier `change-log` : voir le code source de `tnsarith` sur `github`.

2020-07-12 Nouvelle version mineure `0.1.0-beta`.

- **FONCTIONS NOMMÉES** : ajout de `\pgcd`, `\ppcm` et `\lcm`.
-

2020-07-10 Première version `0.0.0-beta`.

1. On ne va pas au-delà de un an depuis la dernière version.

7 Toutes les fiches techniques

7.1 Introduction

7.2 Ensembles classiques

7.3 Opérateurs de base

7.3.1 Arithmétique – Opérateurs de base

`\divides <macro>` (Sans argument)
`\ndivides <macro>` (Sans argument)
`\nequiv <macro>` (Sans argument)
`\modulo <macro>` (Sans argument)

7.4 Fonctions nommées spéciales

7.4.1 Fonctions nommées sans paramètre

`\pgcd <macro>` (Sans argument)
`\ppcm <macro>` (Sans argument)

`\lcm <macro>` (Sans argument)

7.5 Fractions continuées

7.5.1 Fractions continuées standard

`\contfrac <macro>` (1 Argument)
`\contfrac* <macro>` (1 Argument)

— Argument: tous les éléments de la fraction continuée séparés par des |.

7.5.2 Fractions continuées généralisées

`\contfracgene <macro>` (1 Argument)
`\contfracgene* <macro>` (1 Argument)

— Argument: tous les éléments de la fraction continuée généralisée séparés par des |.

7.5.3 Comme une fraction continuée isolée

`\singlecontfrac <macro>` (2 Arguments)

— Argument 1: le pseudo numérateur.

— Argument 2: le pseudo dénominateur.

7.5.4 Fractions continuées – L’opérateur \mathcal{K}

La macro suivante sans argument a un comportement spécifique vis à vis des mises en index et en exposant.

`\contfracope <macro>` (Sans argument)