

Le package `tnssets` : théorie générale des ensembles et des applications

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

Version 0.0.0-beta développée et testée sur Mac OS X.

Christophe BAL

2020-07-10

Table des matières

| | |
|---|----------|
| 1. Introduction | 3 |
| 2. Ensembles | 3 |
| a. Ensembles versus accolades | 3 |
| b. Ensembles pour la géométrie | 3 |
| c. Ensembles probabilistes | 3 |
| d. Ensembles pour l'algèbre générale | 4 |
| e. Ensembles classiques en mathématiques et en informatique théorique | 4 |
| i. La liste complète | 4 |
| f. Ensembles classiques suffixés | 4 |
| g. Des suffixes à la carte | 5 |
| 3. Intervalles | 5 |
| a. Intervalles réels - Notation française (?) | 5 |
| b. Intervalles réels – Notation américaine | 6 |
| c. Intervalles discrets d'entiers | 6 |
| 4. Unions et intersections en mode ligne | 6 |
| 5. Applications | 7 |
| a. Cardinal, image et compagnie | 7 |
| b. Application totale, partielle, injective, surjective et/ou bijective | 7 |
| 6. Historique | 9 |

| | |
|--|-----------|
| 7. Toutes les fiches techniques | 10 |
| a. Ensembles | 10 |
| i. Ensembles versus accolades | 10 |
| ii. Ensembles pour la géométrie | 10 |
| iii. Ensembles probabilistes | 10 |
| iv. Ensembles pour l'algèbre générale | 10 |
| v. Ensembles classiques suffixés | 11 |
| vi. Des suffixes à la carte | 12 |
| b. Intervalles | 12 |
| i. Intervalles réels - Notation française (?) | 12 |
| ii. Intervalles réels – Notation américaine | 12 |
| iii. Intervalles discrets d'entiers | 13 |
| c. Unions et intersections en mode ligne | 13 |
| d. Applications | 14 |
| i. Cardinal, image et compagnie | 14 |
| ii. Application totale, partielle, injective, surjective et/ou bijective | 14 |

1. Introduction

Le package `tnssets` propose des macros utiles pour la rédaction de texte sur la théorie basique et générale des ensembles et des applications via un codage sémantique simple.

Remarque. Ce package s'appuie sur `tnscom` disponible sur <https://github.com/typensee-latex/tnscom.git>.

2. Ensembles

a. Ensembles versus accolades

Exemple 1

| | |
|---|-------------|
| <code>$\setgene{1 ; 3 ; 5}$</code> | $\{1;3;5\}$ |
|---|-------------|

Exemple 2

Dans l'exemple suivant on utilise l'option `sb` pour `s`-mall `b`-races soit « *petites accolades* » en anglais.

| | |
|--|---|
| <code>$\setgene{\dfrac{1}{3} ; \dfrac{5}{7} ; \dfrac{9}{11}}$</code> | $\left\{ \frac{1}{3} ; \frac{5}{7} ; \frac{9}{11} \right\}$ |
| <code>$\setgene*{\dfrac{1}{3} ; \dfrac{5}{7} ; \dfrac{9}{11}}$</code> | $\left\{ \frac{1}{3} ; \frac{5}{7} ; \frac{9}{11} \right\}$ |

b. Ensembles pour la géométrie

Exemple 1

| | |
|---|--------------------------------------|
| <code>\setgeo{C}</code> , <code>\setgeo{D}</code> ou <code>\setgeo{d}</code> | \mathcal{C} , \mathcal{D} ou d |
|---|--------------------------------------|

Remarque. Pour le moment, il n'est pas possible de taper `\setgeo{ABC}` avec plusieurs lettres.

Exemple 2 – Avec des indices

| | |
|---|------------------------------------|
| <code>$\setgeo*{C}{1}$</code> ou <code>$\setgeo*{C}{2}$</code> | \mathcal{C}_1 ou \mathcal{C}_2 |
|---|------------------------------------|

c. Ensembles probabilistes

Exemple 1

| | |
|---|--------------------------------|
| <code>\setproba{E}</code> ou <code>\setproba{G}</code> | \mathcal{E} ou \mathcal{G} |
|---|--------------------------------|

Remarque. Pour le moment, il n'est pas possible de taper `\setproba{ABC}` avec plusieurs lettres.

Exemple 2 – Avec des indices

| | |
|--|------------------------------------|
| <code>\setproba*{E}{1}</code> ou <code>\setproba*{E}{2}</code> | \mathcal{E}_1 ou \mathcal{E}_2 |
|--|------------------------------------|

d. Ensembles pour l’algèbre générale

Exemple 1

| | |
|---|--|
| <code>\setalge{A}</code> , <code>\setalge{K}</code> , <code>\setalge{h}</code> ou <code>\setalge{k}</code> | \mathbb{A} , \mathbb{K} , \mathbb{h} ou \mathbb{k} |
|---|--|

Remarque. Pour le moment, il n’est pas possible de taper `\setalge{ABC}` avec plusieurs lettres.

Exemple 2 – Avec des indices

| | |
|--|----------------------------------|
| <code>\setalge*{k}{1}</code> ou <code>\setalge*{k}{2}</code> | \mathbb{k}_1 ou \mathbb{k}_2 |
|--|----------------------------------|

e. Ensembles classiques en mathématiques et en informatique théorique

i. La liste complète

Dans l’exemple suivant, \mathbb{P} désigne l’ensemble des nombres premiers, \mathbb{H} celui des quaternions, \mathbb{O} celui des octonions et \mathbb{F} un ensemble de nombres flottants (*notation à préciser suivant le contexte*).

| | |
|---|---|
| <code>\nullset</code> | \emptyset |
| <code>\NN</code> , <code>\ZZ</code> , <code>\PP</code> | \mathbb{N} , \mathbb{Z} , \mathbb{P} |
| <code>\DD</code> , <code>\QQ</code> , <code>\RR</code> , <code>\CC</code> | \mathbb{D} , \mathbb{Q} , \mathbb{R} , \mathbb{C} |
| <code>\HH</code> , <code>\OO</code> | \mathbb{H} , \mathbb{O} |
| <code>\FF</code> | \mathbb{F} |

f. Ensembles classiques suffixés

L’ensemble \mathbb{R} nous permet de voir tous les cas possibles.

| | |
|---|--|
| <code>\RRn</code> , <code>\RRp</code> , <code>\RRs</code> | \mathbb{R}_- , \mathbb{R}_+ , \mathbb{R}^* |
| <code>\RRsn</code> , <code>\RRsp</code> | \mathbb{R}_-^* , \mathbb{R}_+^* |

Nous avons utilisé les suffixes **n** pour **n**-égatif, **p** pour **p**-ositif et **s** pour **s**-tar soit « étoile » en anglais. Il y a aussi les suffixes composites **sn** et **sp**.

Notez qu'il est interdit d'utiliser \mathbb{C}_n pour \mathbb{C}_- car l'ensemble \mathbb{C} ne possède pas de structure ordonnée standard. Jetez un oeil à la section suivante pour apprendre à taper \mathbb{C}_- si vous en avez besoin. L'interdiction est ici purement sémantique !

Remarque. La table 1 de la présente page montre les associations autorisées entre ensembles classiques et suffixes.

TABLE 1 – Suffixes

| | n | p | s | sn | sp |
|-----------------------|---|---|---|----|----|
| $\backslash\text{NN}$ | | | × | | |
| $\backslash\text{PP}$ | | | | | |
| $\backslash\text{ZZ}$ | | | | | |
| $\backslash\text{DD}$ | × | × | × | × | × |
| $\backslash\text{QQ}$ | | | | | |
| $\backslash\text{RR}$ | | | | | |
| $\backslash\text{CC}$ | | | | | |
| $\backslash\text{HH}$ | | | × | | |
| $\backslash\text{OO}$ | | | | | |
| $\backslash\text{FF}$ | × | × | × | × | × |

g. Des suffixes à la carte

Dans l'exemple suivant, il faut savoir que le 2^e argument ne peut prendre que les valeurs **n**, **p**, **s**, **sn** ou **sp**.

| | |
|--|--|
| \mathbb{C}_n , \mathbb{H}_s ou $\mathbb{P}_{\leq 0}$ | \mathbb{C}_- , \mathbb{H}_+^* ou $\mathbb{P}_{\leq 0}$ |
|--|--|

3. Intervalles

a. Intervalles réels - Notation française (?)

Exemple 1

Dans cet exemple, la syntaxe fait référence à **O**-pened et **C**-losed pour « *ouvert et fermé* » en anglais. Nous verrons que **CC** et **OO** sont contractés en **C** et **O**. Notez au passage que la macro utilisée résout un problème d'espacement vis à vis du signe = .

| | |
|---|-------------------------|
| $I =]a ; b] = \text{intervalOC}\{a\}\{b\}$ | $I =]a ; b] =]a ; b]$ |
|---|-------------------------|

Exemple 2

Les crochets s'étendent verticalement automatiquement. Pour empêcher cela, il suffit d'utiliser la version étoilée de la macro. Dans ce cas, les crochets restent tout de même un peu plus grands que des crochets utilisés directement. Voici un exemple.

```

 $\backslash intervalC{\frac{1}{2}}{1^{2^3}}$ 
=
 $[\frac{1}{2}; 1^{2^3}]$ 
=
 $\backslash intervalC*{\frac{1}{2}}{1^{2^3}}$ 

```

$$\left[\frac{1}{2}; 1^{2^3}\right] = \left[\frac{1}{2}; 1^{2^3}\right] = \left[\frac{1}{2}; 1^{2^3}\right]$$

b. Intervalles réels – Notation américaine

Dans l'exemple suivant la syntaxe fait référence à P-arenthèse. Cette notation est utilisée aux États Unis.

```

 $\backslash intervalPC{a}{b} = \backslash intervalOC{a}{b}$ 
et
 $\backslash intervalP{a}{b} = \backslash intervalO{a}{b}$ 

```

$$(a; b) =]a; b[\text{ et } (a; b) =]a; b[.$$

c. Intervalles discrets d'entiers

Dans l'exemple suivant la syntaxe fait référence à \mathbb{Z} l'ensemble des entiers relatifs.

```

 $\backslash ZintervalC{-1}{4} =$ 
 $\{-1; 0; 1; 2; 3; 4\}$ 
 $\backslash ZintervalC{-1}{4} =$ 
 $\backslash ZintervalO{-2}{5}$ 

```

$$\llbracket -1; 4 \rrbracket = \{-1; 0; 1; 2; 3; 4\}$$

$$\llbracket -1; 4 \rrbracket = \llbracket -2; 5 \rrbracket.$$

4. Unions et intersections en mode ligne

L^AT_EX permet d'afficher sans souci $\bigcup_{k=1}^n$ mais ne propose pas \bigcup^n . Les macros `\dcap`, `\dcup` et `\dsqcup` donnent accès à ce type de fonctionnalité pour \cap , \cup et \sqcup respectivement. Voici des exemples d'utilisation.

Exemple 1 – Les symboles « seuls »

```

 $A \dcap B = C \cap D$ 
 $A \dcup B = C \cup D$ 
 $A \dsqcup B = C \sqcup D$ 

```

$$A \cap B = C \cap D$$

$$A \cup B = C \cup D$$

$$A \sqcup B = C \sqcup D$$

Exemple 2 – Des intersections indicées

Ci-dessous est utilisée la macro `\bigcap` proposée par le package `amssymb`.

```

 $\cap_{k=1}^n A_k$  ,
 $\cap_{k=1}^n B_k$  ,
 $\bigcap_{k=1}^n C_k$  ,
 $\displaystyle%$ 
 $\bigcap_{k=1}^n D_k$ 

```

$$\cap_{k=1}^n A_k , \cap_{k=1}^n B_k , \cap_{k=1}^n C_k , \bigcap_{k=1}^n D_k$$

Exemple 3 – Des unions indicées

Ci-dessous est utilisée la macro `\bigcup` proposée par le package `amssymb`.

```

 $\cup_{k=1}^n A_k$  ,
 $\cup_{k=1}^n B_k$  ,
 $\bigcup_{k=1}^n C_k$  ,
 $\displaystyle%$ 
 $\bigcup_{k=1}^n D_k$ 

```

$$\cup_{k=1}^n A_k , \cup_{k=1}^n B_k , \cup_{k=1}^n C_k , \bigcup_{k=1}^n D_k$$

Exemple 4 – Des unions disjointes indicées

Ci-dessous sont utilisées les macros `\sqcup` et `\bigsqcup` proposée par le package `amssymb`.

```

 $\sqcup_{k=1}^n A_k$  ,
 $\sqcup_{k=1}^n B_k$  ,
 $\bigsqcup_{k=1}^n C_k$  ,
 $\displaystyle%$ 
 $\bigsqcup_{k=1}^n D_k$ 

```

$$\sqcup_{k=1}^n A_k , \sqcup_{k=1}^n B_k , \sqcup_{k=1}^n C_k , \bigsqcup_{k=1}^n D_k$$

5. Applications

a. Cardinal, image et compagnie

Exemple 1 – Cardinal

```

 $\card E = \card E$ 

```

$$\#E = \text{card } E$$

Exemple 2 – Image et compagnie

Ci-dessous se trouve la macro `\ker` proposée par `amsmath` qui est importé par `tnssets`.

```

 $\ker f$  ,  $\dom f$  ,
 $\im f$  ou  $\codom f$ 

```

$$\ker f , \text{dom } f , \text{im } f \text{ ou } \text{codom } f$$

b. Application totale, partielle, injective, surjective et/ou bijective

Voici des symboles qui, bien que très techniques, facilitent la rédaction de documents à propos des applications totales ou partielles ¹ (*on parle aussi d'applications, sans qualificatif, et de fonctions*).

1. $a : E \rightarrow F$ est une application totale si $\forall x \in E, \exists ! y \in F$ tel que $y = a(x)$. Plus généralement, $f : E \rightarrow F$ est une application partielle si $\forall x \in E, \exists_{\leq 1} y \in F$ tel que $y = f(x)$, autrement dit soit $f(x)$ existe dans F , soit f n'est pas définie en x .

Exemple 1 – Applications totales

$f: A \rightarrow B$ est une application totale, c'est à dire définie sur A tout entier.

$i: C \rightarrow D$ est une application totale injective.

$s: E \rightarrow F$ est une application totale surjective.

$b: G \rightarrow H$ est une application totale bijective.

$f: A \rightarrow B$ est une application totale, c'est à dire définie sur A tout entier.

$i: C \rightarrow D$ est une application totale injective.

$s: E \rightarrow F$ est une application totale surjective.

$b: G \rightarrow H$ est une application totale bijective.

Exemple 2 – Applications partielles

$f: A \rightarrowtail B$ est une application partielle, c'est à dire définie sur un sous-ensemble de A .

$i: C \rightarrowtail D$ est une application partielle injective.

$s: E \rightarrowtail F$ est une application partielle surjective.

$b: G \rightarrowtail H$ est une application partielle bijective.

$f: A \rightarrowtail B$ est une application partielle, c'est à dire définie sur un sous-ensemble de A .

$i: C \rightarrowtail D$ est une application partielle injective.

$s: E \rightarrowtail F$ est une application partielle surjective.

$b: G \rightarrowtail H$ est une application partielle bijective.

6. Historique

Nous ne donnons ici qu'un très bref historique récent² de **tnssets** à destination de l'utilisateur principalement. Tous les changements sont disponibles uniquement en anglais dans le dossier **change-log** : voir le code source de **tnssets** sur **github**.

2020-07-10 Première version 0.0.0-beta.

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

7. Toutes les fiches techniques

a. Ensembles

i. Ensembles versus accolades

`\setgene <macro> [1 Option] (1 Argument)`

— **Option**: la valeur par défaut est `b`. Voici les différentes valeurs possibles.

1. `b` : on utilise des accolades extensibles.
2. `sb` : on utilise des accolades non extensibles.

— **Argument**: la définition de l'ensemble.

— **Argument**: la définition de l'ensemble.

ii. Ensembles pour la géométrie

`\setgeo <macro> (1 Argument)`

— **Argument**: un seul caractère ASCII indiquant un ensemble géométrique.

`\setgeo* <macro> (2 Arguments)`

— **Argument 1**: un seul caractère ASCII indiquant \mathcal{U} dans le nom \mathcal{U}_d d'un ensemble géométrique.

— **Argument 2**: un texte donnant d dans le nom \mathcal{U}_d d'un ensemble géométrique.

iii. Ensembles probabilistes

`\setproba <macro> (1 Argument)`

— **Argument**: un seul caractère ASCII majuscule indiquant un ensemble probabiliste.

`\setproba* <macro> (2 Arguments)`

— **Argument 1**: un seul caractère ASCII majuscule indiquant \mathcal{U} dans le nom \mathcal{U}_d d'un ensemble probabiliste.

— **Argument 2**: un texte donnant d dans le nom \mathcal{U}_d d'un ensemble probabiliste.

iv. Ensembles pour l'algèbre générale

`\setalge <macro> (1 Argument)`

— **Argument**: soit l'une des lettres `h` et `k`, soit un seul caractère ASCII majuscule indiquant un ensemble de type anneau ou corps.

`\setalge* <macro> (2 Arguments)`

— **Argument 1**: un seul caractère ASCII indiquant \mathbb{U} dans le nom \mathbb{U}_d d'un ensemble de type anneau ou corps.

— **Argument 2**: un texte donnant d dans le nom \mathbb{U}_d d'un ensemble de type anneau ou corps.

v. Ensembles classiques suffixés

\NN <macro> (Sans argument)
\NNs <macro> (Sans argument)

\PP <macro> (Sans argument)

\ZZ <macro> (Sans argument)
\ZZn <macro> (Sans argument)
\ZZp <macro> (Sans argument)
\ZZs <macro> (Sans argument)
\ZZsn <macro> (Sans argument)
\ZZsp <macro> (Sans argument)

\DD <macro> (Sans argument)
\DDn <macro> (Sans argument)
\DDp <macro> (Sans argument)
\DDs <macro> (Sans argument)
\DDsn <macro> (Sans argument)
\DDsp <macro> (Sans argument)

\QQ <macro> (Sans argument)
\QQn <macro> (Sans argument)
\QQp <macro> (Sans argument)
\QQs <macro> (Sans argument)
\QQsn <macro> (Sans argument)
\QQsp <macro> (Sans argument)

\RR <macro> (Sans argument)
\RRn <macro> (Sans argument)
\RRp <macro> (Sans argument)
\RRs <macro> (Sans argument)
\RRsn <macro> (Sans argument)
\RRsp <macro> (Sans argument)

\CC <macro> (Sans argument)
\CCs <macro> (Sans argument)

\HH <macro> (Sans argument)
\HHs <macro> (Sans argument)

\OO <macro> (Sans argument)
\OOs <macro> (Sans argument)

vi. Des suffixes à la carte

`\setspecial <macro> (2 Arguments)`
`\setspecial* <macro> (2 Arguments)`

- Argument 1: l'ensemble à "suffixer".
- Argument 2: l'un des suffixes `n`, `p`, `s`, `sn` ou `sp`.

b. Intervalles

i. Intervalles réels - Notation française (?)

Pour toutes les macros ci-dessous, la version non étoilée produit des délimiteurs qui s'étirent si besoin verticalement, tandis que la version étoilée ne le fait pas.

`\intervalC0 <macro> (2 Arguments)`
`\intervalC0* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $[a; b[$.
- Argument 2: borne supérieure b de l'intervalle $[a; b[$.

`\intervalC <macro> (2 Arguments)`
`\intervalC* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $[a; b]$.
- Argument 2: borne supérieure b de l'intervalle $[a; b]$.

`\interval0 <macro> (2 Arguments)`
`\interval0* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $]a; b[$.
- Argument 2: borne supérieure b de l'intervalle $]a; b[$.

`\interval0C <macro> (2 Arguments)`
`\interval0C* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $]a; b]$.
- Argument 2: borne supérieure b de l'intervalle $]a; b]$.

ii. Intervalles réels – Notation américaine

Pour toutes les macros ci-dessous, la version non étoilée produit des délimiteurs qui s'étirent si besoin verticalement, tandis que la version étoilée ne le fait pas.

`\intervalCP <macro> (2 Arguments)`
`\intervalCP* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $[a; b)$.
- Argument 2: borne supérieure b de l'intervalle $[a; b)$.

`\intervalP <macro> (2 Arguments)`
`\intervalP* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $(a; b)$.
 - Argument 2: borne supérieure b de l'intervalle $(a; b)$.
-

`\intervalPC <macro> (2 Arguments)`
`\intervalPC* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $(a; b]$.
- Argument 2: borne supérieure b de l'intervalle $(a; b]$.

iii. Intervalles discrets d'entiers

Pour toutes les macros ci-dessous, la version non étoilée produit des délimiteurs qui s'étirent si besoin verticalement, tandis que la version étoilée ne le fait pas.

`\ZintervalCO <macro> (2 Arguments)`
`\ZintervalCO* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $\llbracket a; b \llbracket$.
 - Argument 2: borne supérieure b de l'intervalle $\llbracket a; b \llbracket$.
-

`\ZintervalC <macro> (2 Arguments)`
`\ZintervalC* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $\llbracket a; b \rrbracket$.
 - Argument 2: borne supérieure b de l'intervalle $\llbracket a; b \rrbracket$.
-

`\ZintervalO <macro> (2 Arguments)`
`\ZintervalO* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $\llbracket a; b \llbracket$.
 - Argument 2: borne supérieure b de l'intervalle $\llbracket a; b \llbracket$.
-

`\ZintervalOC <macro> (2 Arguments)`
`\ZintervalOC* <macro> (2 Arguments)`

- Argument 1: borne inférieure a de l'intervalle $\llbracket a; b \rrbracket$.
- Argument 2: borne supérieure b de l'intervalle $\llbracket a; b \rrbracket$.

c. Unions et intersections en mode ligne

`\dcap <macro> (Sans argument)`
`\dcup <macro> (Sans argument)`
`\dsqcup <macro> (Sans argument)`

d. Applications

i. Cardinal, image et compagnie

`\card <macro> (Sans argument)`
`\card* <macro> (Sans argument)`

`\dom <macro> (Sans argument)`
`\codom <macro> (Sans argument)`
`\im <macro> (Sans argument)`

ii. Application totale, partielle, injective, surjective et/ou bijective

| | |
|--|---|
| <code>\to <macro> (Sans argument)</code> | <code>\pto <macro> (Sans argument)</code> |
| <code>\onetoone <macro> (Sans argument)</code> | <code>\ponetoone <macro> (Sans argument)</code> |
| <code>\onto <macro> (Sans argument)</code> | <code>\ponto <macro> (Sans argument)</code> |
| <code>\bijet <macro> (Sans argument)</code> | <code>\pbijet <macro> (Sans argument)</code> |