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

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

Version `0.2.0-beta` développée et testée sur Mac OS X.

Christophe BAL

2020-07-30

---

## Table des matières

---

# 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.

# 2. Beta-dépendance

`tnscom` qui est disponible sur `https://github.com/typensee-latex/tnscom.git` est un package utilisé en coulisse.

# 3. Packages utilisés

La roue ayant déjà été inventée, le package `tnssets` réutilise les packages suivants sans aucun scrupule.

- `amssymb`
- `dsfont`
- `forloop`
- `mathrsfs`
- `mathtools`
- `relsize`
- `stackengine`
- `stmaryrd`
- `xstring`
- `yhmath`

# 4. Ensembles

## a. Ensembles versus accolades

**Exemple 1**

```
$\setgene{1 ; 3 ; 5}$ .
```

$\{1\,;3\,;5\}$ .

**Exemple 2**

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

```
$\setgene {\dfrac{1}{3} ; \dfrac{5}{7} ;
          \dfrac{9}{11}}$

$\setgene*{\dfrac{1}{3} ; \dfrac{5}{7} ;
          \dfrac{9}{11}}$
```

$\left\{\dfrac{1}{3}\,;\dfrac{5}{7}\,;\dfrac{9}{11}\right\}$

$\{\dfrac{1}{3}\,;\dfrac{5}{7}\,;\dfrac{9}{11}\}$

## b. Ensembles pour la géométrie

**Exemple 1**

```
$\setgeo{C}$ ,
$\setgeo{D}$ ou
$\setgeo{d}$
```

$\mathscr{C}$ , $\mathscr{D}$ ou $\mathscr{d}$

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

**Exemple 2 – Avec des indices**

| | |
|---|---|
| `$\setgeo*{C}{1}$ ou $\setgeo*{C}{2}$` | $\mathscr{C}_1$ ou $\mathscr{C}_2$ |

## c. Ensembles probabilistes

**Exemple 1**

| | |
|---|---|
| `$\setproba{E}$ ou $\setproba{G}$` | $\mathcal{E}$ ou $\mathcal{G}$ |

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

**Exemple 2 – Avec des indices**

| | |
|---|---|
| `$\setproba*{E}{1}$ ou $\setproba*{E}{2}$` | $\mathcal{E}_1$ ou $\mathcal{E}_2$ |

## d. Ensembles pour l'algèbre générale

**Exemple 1**

| | |
|---|---|
| `$\setalge{A}$ , $\setalge{K}$ , $\setalge{h}$ ou $\setalge{k}$` | $\mathbb{A}$ , $\mathbb{K}$ , $\mathbb{h}$ ou $\mathbb{k}$ |

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

**Exemple 2 – Avec des indices**

| | |
|---|---|
| `$\setalge*{k}{1}$ ou $\setalge*{k}{2}$` | $\mathbb{k}_1$ 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)*.

```
$\nullset$

$\NN$ , $\ZZ$ , $\PP$

$\DD$ , $\QQ$ , $\RR$ , $\CC$

$\HH$ , $\OO$

$\FF$
```

$\varnothing$
$\mathbb{N}$ , $\mathbb{Z}$ , $\mathbb{P}$
$\mathbb{D}$ , $\mathbb{Q}$ , $\mathbb{R}$ , $\mathbb{C}$
$\mathbb{H}$ , $\mathbb{O}$
$\mathbb{F}$

## f. Ensembles classiques suffixés

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

```
$\RRn$ , $\RRp$ , $\RRs$

$\RRsn$ , $\RRsp$
```

$\mathbb{R}_{-}$ , $\mathbb{R}_{+}$ , $\mathbb{R}^{*}$
$\mathbb{R}_{-}^{*}$ , $\mathbb{R}_{+}^{*}$

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 `$\CCn$` 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 |
|---|---|---|---|---|---|
| `\NN` | | | × | | |
| `\PP` | | | | | |
| `\ZZ`<br>`\DD`<br>`\QQ`<br>`\RR` | × | × | × | × | × |
| `\CC`<br>`\HH`<br>`\OO` | | | × | | |
| `\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`.

| | |
|---|---|
| `$\setspecial{\CC}{n}$ ,`<br>`$\setspecial{\HH}{sp}$ ou`<br>`$\setspecial*{\setproba{P}}{n}$` | $\mathbb{C}_{-}$ , $\mathbb{H}_{+}^{*}$ ou $\mathcal{P}_{\leq 0}$ |

# 5. 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 = .

| | |
|---|---|
| `$I = ]a ; b] = \intervalOC{a}{b}$` | $I = ]a\,;b] = \left]a\,;b\right]$ |

### 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.

| | |
|---|---|
| `$\displaystyle`<br>` \intervalC{ \frac{1}{2} }{ 1^{2^{3}} }`<br>` =`<br>` [ \frac{1}{2} ; 1^{2^{3}} ]`<br>` =`<br>` \intervalC*{ \frac{1}{2} }{ 1^{2^{3}} }$` | $\displaystyle \left[\frac{1}{2}\,;1^{2^{3}}\right] = [\frac{1}{2}\,;1^{2^{3}}] = \big[\frac{1}{2}\,;1^{2^{3}}\big]$ |

## b. Intervalles réels – Notation américaine

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

| | |
|---|---|
| `$\intervalPC{a}{b} = \intervalOC{a}{b}$`<br>`et`<br>`$\intervalP{a}{b} = \intervalO{a}{b}$.` | $(a\,;b] = ]a\,;b]$ 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.

| | |
|---|---|
| `$\ZintervalC{-1}{4} =`<br>` \{ -1 ; 0 ; 1 ; 2 ; 3 ; 4 \}$`<br><br>`$\ZintervalC{-1}{4} =`<br>` \ZintervalO{-2}{5}$.` | $[\![-1\,;4]\!] = \{-1\,;0\,;1\,;2\,;3\,;4\}$<br>$[\![-1\,;4]\!] = ]\!]-2\,;5[\![$. |

# 6. Unions et intersections en mode ligne

LaTeX permet d'afficher sans souci $\cup_{k=1}^{n}$ mais ne propose pas $\displaystyle\bigcup_{k=1}^{n}$. 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 »**

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

$A \dcup B = C \cup D$

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

$A \bigcap B = C \cap D$
$A \bigcup B = C \cup D$
$A \bigsqcup B = C \sqcup D$

**Exemple 2 – Des intersections indicées**

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

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

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

**Exemple 3 – Des unions indicées**

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

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

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

**Exemple 4 – Des unions disjointes indicées**

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

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

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

# 7. Applications

## a. Cardinal, image et compagnie

### Exemple 1 – Cardinal

| | |
|---|---|
| `$\card* E = \card E$` | $\# E = \operatorname{card} E$ |

### Exemple 2 – Image et compagnie

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

| | |
|---|---|
| `$\ker f$ , $\dom f$ ,`<br>`$\im f$ ou $\codom f$` | $\ker f$ , $\operatorname{dom} f$ , $\operatorname{im} f$ ou $\operatorname{codom} f$ |

## 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).*

### Exemple 1 – Applications totales

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

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

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

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

$f : A \to B$ est une application totale, c'est à dire définie sur $A$ tout entier.
$i : C \rightarrowtail D$ est une application totale injective.
$s : E \twoheadrightarrow F$ est une application totale surjective.
$b : G \rightarrowtail\!\!\!\!\twoheadrightarrow H$ est une application totale bijective.

1. $a : E \to F$ est une application totale si $\forall x \in E$, $\exists! y \in F$ tel que $y = a(x)$. Plus généralement, $f : E \to F$ 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 2 – Applications partielles**

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

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

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

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

$f : A \nrightarrow B$ est une application partielle, c'est à dire définie sur un sous-ensemble de $A$.
$i : C \rightarrowtail\!\!\!\!+ D$ est une application partielle injective.
$s : E \twoheadrightarrow\!\!\!\!+ F$ est une application partielle surjective.
$b : G \rightarrowtail\!\!\!\!\twoheadrightarrow\!\!\!\!+ H$ est une application partielle bijective.

## c. Composition

**Exemple 1 – Opérateur**

La macro `\compo` est juste une version un peu plus petite de `\circ`.

```
$f \compo g$ et non
$f \circ g$
```

$f \circ g$ et non $f \circ g$

**Exemple 2 – Compositions successives**

La macro `\multicompo` sert à indiquer la composition d'une application plusieurs fois de suite par elle-même. Voici toutes les mises en forme disponibles où l'option `exp` nécessite que le nombre d'applications composées soit un naturel non nul connu. Vous noterez que l'écriture par défaut, qui n'est pas standard, n'est pas $f^{(p)}$ car cette notation est traditionnellement utilisée pour indiquer la dérivée p^e d'une application.

```
 $\multicompo     {g}{5}
= \multicompo[exp]{g}{5}$

 $\multicompo     {f}{p}
= \multicompo[dot]{f}{p}$
```

$g^{\langle 5 \rangle} = g \circ g \circ g \circ g \circ g$
$f^{\langle p \rangle} = f \circ \cdots \circ f$

**Remarque.** La convention retenue est analogue à ce que l'on fait avec les puissances de nombres réels. En particulier, $f^{\langle 1 \rangle} = f$.

# 8. Historique

Nous ne donnons ici qu'un très bref historique récent [2] 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`.

**2020-07-30** Nouvelle version mineure `0.2.0-beta`.

- **Composition d'applications.**
  - `\compo` est un opérateur de composition de deux applications.
  - `\multicomp` permet d'indiquer des compositions successives d'une application par elle-même.

---

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

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

# 9. Toutes les fiches techniques

## a. Ensembles

### i. Ensembles versus accolades

`\setgene[#opt]{#1}`

— `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{#1}`

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

---

`\setgeo*{#1..#2}`

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

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

### iii. Ensembles probabilistes

`\setproba{#1}`

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

---

`\setproba*{#1..#2}`

— `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{#1}`

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

---

`\setalge*{#1..#2}`

— `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` `\NNs`

---

`\PP`

---

`\ZZ` `\ZZn` `\ZZp` `\ZZs` `\ZZsn` `\ZZsp`

---

`\DD` `\DDn` `\DDp` `\DDs` `\DDsn` `\DDsp`

---

`\QQ` `\QQn` `\QQp` `\QQs` `\QQsn` `\QQsp`

---

`\RR` `\RRn` `\RRp` `\RRs` `\RRsn` `\RRsp`

---

`\CC` `\CCs`

---

`\HH` `\HHs`

---

`\OO` `\OOs`

#### vi. Des suffixes à la carte

```
\setspecial {#1..#2}
\setspecial*{#1..#2}
```

— `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.

---

```
\intervalCO {#1..#2}
\intervalCO*{#1..#2}
```

— `Argument 1:` borne inférieure $a$ de l'intervalle $[a\,;b[$.

— `Argument 2:` borne supérieure $b$ de l'intervalle $[a\,;b[$.

---

```
\intervalC {#1..#2}
\intervalC*{#1..#2}
```

— `Argument 1` : borne inférieure $a$ de l'intervalle $[a\,;b]$.

— `Argument 2` : borne supérieure $b$ de l'intervalle $[a\,;b]$.

---

```
\intervalO {#1..#2}
\intervalO*{#1..#2}
```

— `Argument 1` : borne inférieure $a$ de l'intervalle $]a\,;b[$.

— `Argument 2` : borne supérieure $b$ de l'intervalle $]a\,;b[$.

---

```
\intervalOC {#1..#2}
\intervalOC*{#1..#2}
```

— `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 {#1..#2}
\intervalCP*{#1..#2}
```

— `Argument 1` : borne inférieure $a$ de l'intervalle $[a\,;b)$.

— `Argument 2` : borne supérieure $b$ de l'intervalle $[a\,;b)$.

---

```
\intervalP {#1..#2}
\intervalP*{#1..#2}
```

— `Argument 1` : borne inférieure $a$ de l'intervalle $(a\,;b)$.

— `Argument 2` : borne supérieure $b$ de l'intervalle $(a\,;b)$.

---

```
\intervalPC {#1..#2}
\intervalPC*{#1..#2}
```

— `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 {#1..#2}
\ZintervalCO*{#1..#2}
```

— `Argument 1` : borne inférieure $a$ de l'intervalle $[\![a\,;b[\![$.

— `Argument 2`: borne supérieure $b$ de l'intervalle $[\![a\,;b[\![$.

---

```
\ZintervalC {#1..#2}
\ZintervalC*{#1..#2}
```

— `Argument 1`: borne inférieure $a$ de l'intervalle $[\![a\,;b]\!]$.

— `Argument 2`: borne supérieure $b$ de l'intervalle $[\![a\,;b]\!]$.

---

```
\ZintervalO {#1..#2}
\ZintervalO*{#1..#2}
```

— `Argument 1`: borne inférieure $a$ de l'intervalle $]\!]a\,;b[\![$.

— `Argument 2`: borne supérieure $b$ de l'intervalle $]\!]a\,;b[\![$.

---

```
\ZintervalOC {#1..#2}
\ZintervalOC*{#1..#2}
```

— `Argument 1`: borne inférieure $a$ de l'intervalle $]\!]a\,;b]\!]$.

— `Argument 2`: borne supérieure $b$ de l'intervalle $]\!]a\,;b]\!]$.

## c. Unions et intersections en mode ligne

```
\dcap
\dcup
\dsqcup
```

## d. Applications

### i. Cardinal, image et compagnie

```
\card
\card*
```

---

```
\dom
\codom
\im
```

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

```
\to
\onetoone
\onto
\biject
```

```
\pto
\ponetoone
\ponto
\pbiject
```

### iii. Composition

`\compo` compo = compo-sition

---

`\multicompo[#opt]{#1..#2}`

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

1. `r` : écriture utilisant des chevrons. `r = r-after`.
2. `exp` : écriture développée. `exp = exp-and`.
3. `dot` : écriture faussement développée utilisant des points de suspension.

— `Argument 1:` l'application.

— `Argument 2:` le nombre d'applications composées.