

Le package `tnslog` : rédiger de la logique simple

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

Version 1.1.0-beta développée et testée sur Mac OS X.

Christophe BAL

2021-03-04

Table des matières

I.	Introduction	3
II.	Beta-dépendance	3
III.	Packages utilisés	3
IV.	Espace après la négation logique	3
V.	Personnaliser les opérateurs de comparaison	3
	1. Décorer avec du texte	3
	2. Quelques écritures symboliques nouvelles	4
VI.	Équivalences et implications	4
	1. Des symboles logiques supplémentaires	4
	2. Équivalences et implications verticales	4
VII.	Des versions alternatives du quantificateur existentiel	5
	1. Quantifier l'existence	5
	2. Versions négatives	5
VIII.	Détailler un raisonnement simple	5
	1. Version pour le lycée et après	5
	2. Version pour les collégiens	9
	3. De courts commentaires	12
IX.	Détailler un « vrai » raisonnement	13
	1. Un tableau pour le post-bac	13
	2. Un tableau sur plusieurs pages	15
	3. Un tableau pour le collège et le lycée	15
	4. Un tableau sur plusieurs pages	17
X.	Historique	18
XI.	Toutes les fiches techniques	19
	1. Espace après la négation logique	19
	2. Personnaliser les opérateurs de comparaison	19
	3. Équivalences et implications	19
	i. Des symboles logiques supplémentaires	19
	ii. Équivalences et implications verticales	19

4. Des versions alternatives du quantificateur existentiel	19
5. Détailler un raisonnement simple	20
i. Détailler un raisonnement simple	20
ii. Détailler un raisonnement simple – Mise en forme du texte	20
6. Détailler un « vrai » raisonnement	21
i. Détailler un « vrai » raisonnement via un tableau	21
ii. Détailler un « vrai » raisonnement via un tableau - Textes utilisés	22

I. Introduction

Le package `tnslog` facilite la rédaction de preuves formelles basiques via un codage sémantique simple.

II. Beta-dépendance

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

III. Packages utilisés

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

- `amssymb`
- `etoolbox`
- `mathtools`
- `witharrows`
- `centernot`
- `longtable`
- `simplekv`

IV. Espace après la négation logique

`\neg` a été redéfinie pour ajouter un peu d'espace après le symbole. Le comportement par défaut se retrouve en utilisant la macro `\stdneg`. Voici un exemple.

<code>\neg A = \stdneg A</code>	$\neg A = \neg A$
<code>\neg\neg A = \stdneg \stdneg A</code>	$\neg\neg A = \neg\neg A$

V. Personnaliser les opérateurs de comparaison

1. Décorer avec du texte

D'un point de vue pédagogique il peut être intéressant de disposer de différentes façons d'écrire une égalité, une non égalité ou une inégalité en lui ajoutant un texte descriptif. Bien entendu on tord les règles de typographie avec ce type de pratique mais c'est pour le bien de la communauté éducative. Pour cela on utilisera l'une des macros suivantes ou leur version négative obtenue en préfixant le nom d'un n.

1. `\eq` donne $=$ tandis que `\neq` donne \neq .
2. `\less` et `\gtr` donnent $<$ et $>$ tandis que `\nless` et `\ngtr` donnent \nless et \ngtr .
3. `\leq` et `\geq` donnent \leq et \geq tandis que `\nleq` et `\ngeq` donnent \nleq et \ngeq .

Voici un exemple où l'utilisation d'arguments optionnels permet de décorer des opérateurs.

<code>f(x) \eq[def] x^3 + 1</code>	$f(x) \stackrel{\text{def}}{=} x^3 + 1$
<code>si \$x \leq[cond] 2</code>	$\text{si } x \stackrel{\text{cond}}{\leq} 2$
<code>f(x) \neq[cons] x^3 + 1</code>	$f(x) \stackrel{\text{cons}}{\neq} x^3 + 1$
<code>car \$x \nleq[hyp] 2</code>	$\text{car } x \stackrel{\text{hyp}}{\nleq} 2$

Remarque. La mise en forme du texte est faite par la macro `\txtdecoope` qui utilise `\coldecoope` pour la coloration (*il est donc facile de changer la couleur du texte mais aussi la mise en forme du texte si besoin*).

2. Quelques écritures symboliques nouvelles

Certaines décorations fournissent une écriture symbolique obtenue via la version étoilée de la macro d'un opérateur. Si une telle écriture n'existe pas, le package utilisera silencieusement la version non étoilée. Pour le moment, seule la macro `\eq` propose ceci ainsi que sa version négative. Voici tous les cas possibles.

<code>\$a \eq*[def] 1\$</code> ou <code>\$a \neq*[def] 1\$</code>	$a := 1$ ou $a \neq 1$
<code>\$b \eq*[id] 2\$</code> ou <code>\$b \neq*[id] 2\$</code>	$b \Rightarrow 2$ ou $b \nRightarrow 2$

VI. Équivalences et implications

1. Des symboles logiques supplémentaires

Exemple 1 – Implication réciproque

En plus des opérateurs `\iff` et `\implies` proposés par L^AT_EX, il a été ajouté l'opérateur `\becauseof` un opérateur pour obtenir \Leftarrow ¹ ainsi que leurs versions négatives. Voici un exemple d'utilisation.

<code>\$(A \implies B)</code> <code>\iff (B \becauseof A)\$</code>	$(A \Rightarrow B) \iff (B \Leftarrow A)$
<code>\$(A \implies B)</code> <code>\niff (A \nimplies B)\$</code>	$(A \Rightarrow B) \niff (A \nRightarrow B)$

Exemple 2 – Des opérateurs décorés

Tout comme pour les comparaisons, pour décorer il suffit de passer via un argument optionnel.

<code>\$A \iff[ssi] B \niff[i.e.] C\$</code>	$A \overset{\text{ssi}}{\iff} B \overset{\text{i.e.}}{\niff} C$
<code>\$A \implies[donc] B \nimplies[donne] C\$</code>	$A \overset{\text{donc}}{\implies} B \overset{\text{donne}}{\nimplies} C$
<code>\$A \becauseof[car] B \nbecauseof[d'après] C\$</code> <code>\rightarrow C\$</code>	$A \overset{\text{car}}{\Leftarrow} B \overset{\text{d'après}}{\nLeftarrow} C$

2. Équivalences et implications verticales

À quoi cela sert-il ?

Les sections ?? et ?? présentent deux environnements pour détailler les étapes d'un raisonnement. Avec ces outils il devient utile d'avoir des versions verticales non décorées des symboles d'équivalence et d'implication. Voici comment les obtenir (*tous les cas possibles ont été indiqués*). Bien entendu le préfixe `v` est pour `v`-ertical.

1. Penser aussi aux preuves d'équivalence par double implication.

```

\begin{tabular}{cccccc}
  $A$          & & & & & & $B$ \\
  & $C$          & & & & & $D$ \\
  & $E$          & & & & & $F$ \\
  \\
  & $\viff$      & & & & & $\vimplies$ \\
  & $\vbecauseof$ & & & & & $\nviff$ \\
  & $\nvimplies$ & & & & & $\nvbecauseof$ \\
  \\
  $A$          & & & & & & $B$ \\
  & $C$          & & & & & $D$ \\
  & $E$          & & & & & $F$ \\
\end{tabular}

```

A	B	C	D	E	F
\Downarrow	\Downarrow	\Uparrow	\Downarrow	\Downarrow	\Downarrow
A	B	C	D	E	F

VII. Des versions alternatives du quantificateur existentiel

1. Quantifier l'existence

Voici deux versions, l'une classique, et l'autre beaucoup moins, permettant de préciser le quantificateur \exists .

```

$\exists\text{sone } x \text{ \textit{in } } E$ pour
\og il existe un seul $x$ \fg.

$\exists\text{multi}\{\leq 1\} y \text{ \textit{in } } F$ pour
\og il existe au plus un $y$ \fg.

$\exists\text{multi}\{=1\} \text{ \textit{eq*} [def] \exists\text{sone} }$

```

$\exists! x \in E$ pour « il existe un seul x ».
 $\exists_{\leq 1} y \in F$ pour « il existe au plus un y ».
 $\exists_{=1} := \exists!$

2. Versions négatives

```

$\nexists\text{sone}$ pour
\og il n'existe pas un unique \fg.

$\exists\text{multi}\{\neq 1\} \text{ \textit{eq*} [def] }$
 $\rightarrow$  $\nexists\text{sone}$

$\nexists\text{multi}\{>4\}$ pour
\og il n'existe pas plus de quatre \fg.

$\nexists\text{sts}$ vient de \textit{verb+amssymb+}.

```

$\nexists!$ pour « il n'existe pas un unique ».
 $\exists_{\neq 1} := \nexists!$
 $\nexists_{>4}$ pour « il n'existe pas plus de quatre ».
 \nexists vient de amssymb.

VIII. Détailler un raisonnement simple

1. Version pour le lycée et après

Exemple 1 – Avec les réglages par défaut

L'environnement `stepcalc` permet de détailler les étapes principales d'un calcul ou d'un raisonnement simple en s'appuyant sur la macro `\explnext` dont le nom vient de « *expl-ain next step* » soit « *expliquer la prochaine étape* » en anglais². On dispose aussi de `\explnext*` pour des explications descendantes et/ou montantes³.

Ci-dessous se trouve un exemple, très farfelu vers la fin, où l'on utilise les réglages par défaut. Notons au passage que ce type de présentation n'est sûrement pas bien adaptée à un jeune public pour lequel une 2^e façon de détailler des calculs et/ou un raisonnement simple est proposée plus bas dans la section 2..

```
\begin{stepcalc}
(a + b)^2
\explnext{On utilise  $x^2 = x \cdot x$ .}
(a + b) (a + b)
\explnext*{Double développement depuis la parenthèse gauche.}%
\explnext*{Double factorisation pas facile.}
a^2 + a b + b a + b^2
\explnext*{}%
\explnext*{Commutativité du produit.}
a^2 + 2 a b + b^2
\explnext*{Commutativité de l'addition.}%
{}
a^2 + b^2 + 2 a b
\end{stepcalc}
```

$$\begin{aligned}
 & (a + b)^2 \\
 = & \quad \{ \text{On utilise } x^2 = x \cdot x. \} \\
 & (a + b)(a + b) \\
 = & \quad \left\{ \begin{array}{c} \downarrow \text{Double développement depuis la parenthèse gauche.} \\ \uparrow \quad \quad \quad \text{Double factorisation pas facile.} \end{array} \right. \\
 & a^2 + ab + ba + b^2 \\
 = & \quad \{ \uparrow \text{Commutativité du produit.} \} \\
 & a^2 + 2ab + b^2 \\
 = & \quad \{ \downarrow \text{Commutativité de l'addition.} \} \\
 & a^2 + b^2 + 2ab
 \end{aligned}$$

Remarque. Il faut savoir que la mise en forme est celle d'une formule ce qui peut rendre service comme dans l'exemple suivant.

2. Cet environnement utilise aussi le package `witharrows` qui est très sympathique pour expliquer des étapes de calcul.

3. Les explications données ne doivent pas être trop longues car ce serait contre-productif.

Un calcul avec un placement pouvant être utile :

```
\begin{stepcalc}
(a + b)^2
\explnext{Identité remarquable.}
a^2 + b^2 + 2 a b
\end{stepcalc}
```

Un calcul avec un placement pouvant être utile :

$$(a + b)^2 = \text{\textit{Identité remarquable.}} a^2 + b^2 + 2ab$$

Avec un retour à la ligne, il faudra donc si besoin gérer l'espacement vertical.

Mon calcul pas trop proche.

```
\medskip
\begin{stepcalc}
(a + b)^2
\explnext{Identité remarquable.}
a^2 + b^2 + 2 a b
\end{stepcalc}
```

Mon calcul pas trop proche.

$$(a + b)^2 = \text{\textit{Identité remarquable.}} a^2 + b^2 + 2ab$$

Remarque. Voici des petites choses à connaître sur les macros `\explnext` et `\explnext*`.

1. `\expltxt` est utilisée par `\explnext` pour mettre en forme le texte d'explication.
2. `\expltxtup` et `\expltxtdown` sont utilisées par `\explnext*` décorer les textes d'explication juste avant leur mise en forme finale via `\expltxtupdown`.
3. `\explnext` et `\explnext*` utilisent la macro constante `\expltxtspacein` pour l'espacement entre le symbole et la courte explication. Par défaut, cette macro vaut `2em`.

Exemple 2 – Utiliser un autre symbole globalement

L'environnement `stepcalc` possède plusieurs options dont l'une est `ope` qui vaut `{=}` par défaut. Ceci permet de faire ce qui suit sans effort.

```
\begin{stepcalc}[ope = \viff]
x^2 + 10 x + 25 = 0
\explnext{Identité remarquable.}
(x + 5)^2 = 0
\explnext{$P^2 = 0$ si et
seulement si $P = 0$.}
x = -5
\end{stepcalc}
```

$$x^2 + 10x + 25 = 0$$

$$\Updownarrow \text{\textit{Identité remarquable.}}$$

$$(x + 5)^2 = 0$$

$$\Updownarrow \text{\textit{$P^2 = 0$ si et seulement si $P = 0$.}}$$

$$x = -5$$

Exemple 3 – Juste utiliser des symboles

Si l'argument obligatoire de la macro `\explnext` est vide alors seul le symbole est affiché (*ne pas oublier les accolades vides*). Voici un court exemple de ceci.

```
\begin{stepcalc}[ope = \viff]
a^2 = b^2
\explnext{}
a = \pm b
\end{stepcalc}
```

$$a^2 = b^2$$

$$\Updownarrow$$

$$a = \pm b$$

Exemple 4 – Utiliser un autre symbole localement

La macro `\explnext` possède un argument optionnel qui utilise par défaut celui de l’environnement. En utilisant cette option, on choisit alors localement le symbole à employer. Voici un exemple d’utilisation complètement farfelu bien que correct.

<pre> \begin{stepcalc}[ope = \viff] 0 \leq a < b \explnext[\vimplies]% {Croissance de \$x^2\$ sur \$R_{+}\$} a^2 < b^2 \explnext{} a^2 - b^2 < 0 \explnext{Identité remarquable.} (a - b)(a + b) < 0 \explnext[\vimplies]{} a \neq b \end{stepcalc> </pre>	$0 \leq a < b$ $\Downarrow \quad \{ \textit{Croissance de } x^2 \text{ sur } R_{+}. \}$ $a^2 < b^2$ \Updownarrow $a^2 - b^2 < 0$ $\Updownarrow \quad \{ \textit{Identité remarquable.} \}$ $(a - b)(a + b) < 0$ \Downarrow $a \neq b$
--	---

Exemple 5 – Choisir la mise en forme des explications

Pour la mise en forme des explications à double sens, la macro `\explnext` fait appel à la macro `\expltxt`. Par défaut, le package utilise la définition suivante.

```

\newcommand\expltxt[1]{%
  \text{\color{blue}\footnotesize \{\,\,\{\itshape #1\}\,\,\} }%
}

```

Pour la mise en forme des explications à sens unique, la macro `\explnext*` fait appel aux macros `\expltxtup`, `\expltxtupdown` et `\expltxtupdown`. Par défaut le package utilise les définitions suivantes.

```

\newcommand\expltxtup[1]{%
  $\uparrow$ #1 $\uparrow$%
}

\newcommand\expltxtupdown[1]{%
  $\downarrow$ #1 $\downarrow$%
}

\newcommand\expltxtupdown[2]{%
  \displaystyle\footnotesize\color{blue}%
  \left\{\,\,%
    \genfrac{}{}{0pt}{}%
    \text{\itshape\expltxtupdown{\samesizeas{#1}{#2}}}%
  \}%
  \text{\itshape\expltxtup{\samesizeas{#2}{#1}}}%
  \}%
  \right\}%
}

```


Nous allons expliquer comment obtenir l'afreux exemple ci-dessous montrant que l'on peut adapter si besoin la mise en forme.

<pre> \begin{stepcalc} (a + b) (a + b) \explnext{Se souvenir de \$P\cdot \hookrightarrow P = P^2\$.}% (a + b)^2 \explnext*{Id. Rm. - D��v.}% {Id. Rm. - Facto.}% a^2 + 2 a b + b^2 \end{stepcalc} </pre>	$ \begin{aligned} & (a + b)(a + b) \\ &= \quad \Downarrow \textit{Se souvenir de } P \cdot P = P^2. \quad \Uparrow \\ & (a + b)^2 \\ &= \quad \left\langle \begin{array}{c} \Downarrow \textit{Id. Rm. - D��v.} \Downarrow \\ \Uparrow \textit{Id. Rm. - Facto.} \Uparrow \end{array} \right\rangle \\ & a^2 + 2ab + b^2 \end{aligned} $
--	--

La mise en forme a   t   obtenue en utilisant le code L^AT_EX suivant o   la macro `\samesizeas{#1}{#2}` rend le texte #1 aussi large que #2 en ajoutant des espaces suppl  mentaires tout en centrant le r  sultat final si besoin (*ne pas oublier de passer en mode texte via `\text`*).

```

\newcommand\myexpltxt[2]{%
  \text{\color{#1} \footnotesize \itshape \bfseries #2}%
}

\renewcommand\expltxt[1]{%
  \myexpltxt{gray}{\Downarrow$ #1 $\Uparrow$}%
}

\renewcommand\expltxtup[1]{%
  \myexpltxt{orange}{\Uparrow$ #1 $\Uparrow$}%
}

\renewcommand\expltxttdown[1]{%
  \myexpltxt{red}{\Downarrow$ #1 $\Downarrow$}%
}

\renewcommand\expltxtupdown[2]{%
  \displaystyle\color{blue!20!black!30!green}%
  \genfrac{\langle}{\rangle}{1pt}{}{%
    \expltxttdown{\samesizeas{#1}{#2}}%
  }{%
    \expltxtup{\samesizeas{#2}{#1}}%
  }%
}

```

2. Version pour les coll  giens

L'environnement `stepcalc`⁴ avec les options `style = ar` et `style = ar*` imprime une r  daction plus classique tout en utilisant des fl  ches pour indiquer les explications (*ar est pour ar-row soit « fl  che » en anglais*). Dans ce cas d'utilisation, la macro `\explnext*` permet d'avoir une fl  che unidirectionnelle, vers le haut ou le bas au choix, ou bien d'  crire deux indications dont l'une est montante et l'autre descendante.

4. Cet environnement utilise aussi le package `witharrows`.

Il existe aussi l'option `style = sar` lorsque la toute 1^{re} étape n'est pas expliquée (*s est pour s-hort soit « court » en anglais*). Voir l'exemple 4 page 11.

Exemple 1 – L'opérateur dans la marge... ou pas

Considérons le code suivant qui utilise `style = ar`.

<pre>\begin{stepcalc}[style = ar] (a + b)^2 \explnext{} a^2 + 2 a b + b^2 \end{stepcalc}</pre>	$(a + b)^2$ $= a^2 + 2ab + b^2$
--	---------------------------------

Intégré dans un paragraphe, on obtient le rendu suivant qui vaut ce qu'il vaut ici.

$$(a + b)^2$$

$$= a^2 + 2ab + b^2$$

En utilisant `style = ar*` au lieu de `style = ar`, on aboutit à ce qui suit qui semble mieux.

$$(a + b)^2$$

$$= a^2 + 2ab + b^2$$

Dans la suite, on va se concentrer sur le style `ar` mais tout ce qui est dit pour ce style s'applique aussi à `ar*`.

Exemple 2 – Des flèches à double sens

<pre>\begin{stepcalc}[style = ar] (a + b)^2 \explnext{Identité remarquable} a^2 + 2 a b + b^2 \explnext{} a^2 + b^2 + 2 a b \end{stepcalc}</pre>	$(a + b)^2$ $= a^2 + 2ab + b^2$ $= a^2 + b^2 + 2ab$ <div style="position: relative; height: 40px;"> ↔ Identité remarquable </div>
--	---

Exemple 3 – Des flèches unidirectionnelles

Ce qui suit est juste là comme démo. car les explications y sont un peu farfelues.

```
\begin{stepcalc}[style = ar]
  (a + b)^2
  \explnext*{Via $P^2 = P \cdot P$.}
  {Via $P \cdot P = P^2$.}
  (a + b) (a + b)
  \explnext*{Double développement.}%
  {Double factorisation (pas simple).}
  a^2 + a b + b a + b^2
  \explnext*{Commutativité du produit.}%
  {}
  a^2 + 2 a b + b^2
  \explnext*{}%
  {Commutativité de l'addition.}
```

```
a^2 + b^2 + 2 a b
\end{stepcalc}
```

$$\begin{aligned}
 & (a+b)^2 \\
 = & (a+b)(a+b) && \left. \begin{array}{l} \text{Via } P^2 = P \cdot P. \\ \text{Double développement.} \end{array} \right\} \text{Via } P \cdot P = P^2. \\
 = & a^2 + ab + ba + b^2 && \left. \begin{array}{l} \text{Double factorisation (pas simple).} \\ \text{Commutativité du produit.} \end{array} \right\} \\
 = & a^2 + 2ab + b^2 && \left. \begin{array}{l} \text{Commutativité de l'addition.} \end{array} \right\} \\
 = & a^2 + b^2 + 2ab
 \end{aligned}$$

Exemple 4 – Ne pas expliquer le tout début

L'environnement `stepcalc` avec l'option `style = sar` débute différemment la mise en forme. Bien entendu ici le tout premier `\explnext` doit avoir un argument vide !

```
\begin{stepcalc}[style = sar]
(a + b) (a + b)
\explnext{}
(a + b)^2
\explnext{Identité remarquable.}
a^2 + b^2 + 2 a b
\end{stepcalc}
```

$$\begin{aligned}
 (a+b)(a+b) &= (a+b)^2 \\
 &= a^2 + b^2 + 2ab \quad \left. \begin{array}{l} \text{Identité remarquable.} \end{array} \right\}
 \end{aligned}$$

Exemple 5 – Choisir son symbole

Voici comment faire où l'implication finale est juste là pour la démonstration (*on notera une petite bidouille un peu sale à faire pour avoir un alignement à peu près correct*).

<pre>\begin{stepcalc}[style = ar, ope = \iff] a^2 + 2 a b + b^2 = 0 \explnext{} (a + b)^2 = 0 \explnext[\:\implies] % {\\$P^2 = 0\\$ ssi \\$P = 0\\$} a + b = 0 \end{stepcalc}</pre>	$ \begin{aligned} & a^2 + 2ab + b^2 = 0 \\ \iff & (a+b)^2 = 0 && \left. \begin{array}{l} \text{P}^2 = 0 \text{ ssi } P = 0. \end{array} \right\} \\ \implies & a + b = 0 \end{aligned} $
--	--

Avec la version courte, on obtient ce qui suit.

```
\begin{stepcalc}[style = sar, ope = \iff]
a^2 + 2 a b + b^2 = 0
\explnext{}
(a + b)^2 = 0
\explnext[\:\implies] %
{\$P^2 = 0\$ ssi \$P = 0\$}
a + b = 0
\end{stepcalc}
```

$$a^2 + 2ab + b^2 = 0 \iff (a + b)^2 = 0 \iff a + b = 0 \quad \left. \vphantom{a^2 + 2ab + b^2 = 0} \right\} P^2 = 0 \text{ ssi } P = 0.$$

3. De courts commentaires

Exemple 1 – Sans alignement

Il est possible d'ajouter de petits commentaires via `\comthis` où `comthis` est pour `com-ment this` soit « *commenter ceci* » en anglais.

```
\begin{stepcalc}
(a + b)^2
\comthis{Forme facto.}
\explnext*{Id.Rq. -- Dév.}%
{Id.Rq. -- Facto.}
a^2 + 2 a b + b^2
\comthis{Forme dév.}
\end{stepcalc}
```

$$\begin{aligned} (a + b)^2 & \quad [\textit{Forme facto.}] \\ = & \quad \left\{ \begin{array}{l} \downarrow \textit{Id.Rq. - Dév.} \downarrow \\ \uparrow \textit{Id.Rq. - Facto.} \uparrow \end{array} \right\} \\ a^2 + 2ab + b^2 & \quad [\textit{Forme dév.}] \end{aligned}$$

Remarque. La mise en forme du texte des commentaires est fait via la macro personnalisable `\explcom`. Quant à l'espacement ajouté entre le texte et son commentaire il est défini par la macro `\expltxtspacein` qui est égale à 2em par défaut.

Exemple 2 – Tout aligner

Il peut être utile d'aligner tous les commentaires. Ceci s'obtient via l'option `com = al` où `al` est pour `al-igné` (par défaut `com = nal` avec le préfixe *n* pour *n-on*).

```
\begin{stepcalc}[com = al]
(a + b)^2
\comthis{Forme facto.}
\explnext*{Id.Rq - Dév.}%
{Id.Rq - Facto.}
a^2 + 2 a b + b^2
\comthis{Forme dév.}
\end{stepcalc}
```

$$\begin{aligned} (a + b)^2 & \quad [\textit{Forme facto.}] \\ = & \quad \left\{ \begin{array}{l} \downarrow \textit{Id.Rq - Dév.} \downarrow \\ \uparrow \textit{Id.Rq - Facto.} \uparrow \end{array} \right\} \\ a^2 + 2ab + b^2 & \quad [\textit{Forme dév.}] \end{aligned}$$

Exemple 3 – Le meilleur des deux mondes

Dans d'autres situations, utiliser les deux types d'alignement peut faire sens. Ceci s'obtient via l'option `com = al` et l'emploi de la macro étoilée `\comthis*` à chaque fois que l'on souhaite "coller" un commentaire le plus à gauche possible.

<pre> \begin{stepcalc}[com = al] (a + b) (a + b) \comthis{Forme facto.} \explnext{Via $x^2 = x \cdot x$.} (a + b)^2 \comthis*{Au passage...} \explnext*{Id.Rq - Dév.}% {Id.Rq - Facto.} a^2 + 2 a b + b^2 \comthis{Forme dév.} \end{stepcalc} </pre>	$ \begin{aligned} & (a + b)(a + b) && [\textit{Forme facto.}] \\ = & \{ \textit{Via } x^2 = x \cdot x. \} \\ & (a + b)^2 && [\textit{Au passage...}] \\ = & \left\{ \begin{array}{c} \downarrow \textit{Id.Rq - Dév.} \downarrow \\ \uparrow \textit{Id.Rq - Facto.} \uparrow \end{array} \right\} \\ & a^2 + 2ab + b^2 && [\textit{Forme dév.}] \end{aligned} $
---	--

Remarque. Si l’alignement n’est pas activé, les macros `\comthis*` et `\comthis` auront toutes les deux le même effet.

Exemple 4 – Ceci marche aussi avec le style « fléché »

Voici ce que donne le mode mixte lorsque des flèches sont utilisées pour les explications. Il semble moins pertinent ici de mixer les modes « alignement » et « non alignement » mais chacun pris séparément peut avoir son utilité.

<pre> \begin{stepcalc}[style = ar, com = al] (a + b) (a + b) \comthis{Forme facto.} \explnext{Via $x^2 = x \cdot x$.} (a + b)^2 \comthis*{Au passage...} \explnext*{Id.Rq - Dév.}% {Id.Rq - Facto.} a^2 + 2 a b + b^2 \comthis{Forme dév.} \end{stepcalc} </pre>	$ \begin{aligned} & (a + b)(a + b) && [\textit{Forme facto.}] \\ = & (a + b)^2 && [\textit{Au passage...}] \\ = & a^2 + 2ab + b^2 && [\textit{Forme dév.}] \end{aligned} $
	$ \begin{array}{l} \curvearrowright \textit{Via } x^2 = x \cdot x. \\ \curvearrowright \textit{Id.Rq - Dév.} \quad \curvearrowright \textit{Id.Rq - Facto.} \end{array} $

Remarque. Bien entendu il est impossible de commenter le tout début en mode fléché court.

IX. Détailler un « vrai » raisonnement

1. Un tableau pour le post-bac

Exemple 1 – Le minimum avec les réglages par défaut

Prenons un exemple utile à la logique formelle en informatique théorique mais qui a complètement sa place en mathématiques plus classiques (*voir la section ?? pour un autre type de présentation plus adapté à un public de collège ou de lycée*). Ci-dessous l’environnement `demotab` facilite la mise en page⁵ et la macro étoilée `\explref*` permet d’indiquer une référence interne au raisonnement⁶.

5. En coulisse est utilisé l’environnement `longtable` du package éponyme.

6. Les indications peuvent être numérotées jusqu’à 99 ce qui est bien au-delà des besoins pratiques.

Dans cet exemple en deux morceaux, pour montrer au passage comment continuer la numérotation là où elle s'était arrêtée, on utilise « *m.p.* » comme abréviation de « *modus ponens* ».

```

\begin{demotab}
  \demostep
    Hypothèse &  $A$ 
  \demostep
    Axiome 1 &  $A \implies B$ 
  \demostep
    m.p. sur
      \explref*{1} et \explref*{2}
    &  $B$ 
  \demostep
    \explref*{1} et \explref*{3}
    &  $A \wedge B$ 
\end{demotab}

```

1	Hypothèse	A
2	Axiome 1	$A \implies B$
3	m.p. sur 1 et 2	B
4	1 et 3	$A \wedge B$

Il est possible de couper sa démonstration en morceaux en indiquant à l'environnement la valeur du 1^{er} numéro de justification via la clé `start` : la valeur spéciale `last` indique de continuer la numérotation à la suite.

<pre>\begin{demotab}[start = last] \demostep Axiome 3 & \$(A \wedge B) \implies C\$ \demostep m.p. sur \explref*[4] et \explref*[5] & \$\$ \end{demotab}</pre>	<div><div>5</div>Axiome 3<div>$(A \wedge B) \implies C$</div></div> <div><div>6</div>m.p. sur<div><div>4</div> et <div>5</div></div><div>C</div></div>
--	--

Exemple 2 – Référencer une indication

L'argument optionnel de `\demostep` permet de définir un label qui ensuite facilitera le référencement d'une justification de façon pérenne via la macro non étoilée `\explref`.

```

\begin{demotab}
  \demostep[demo-my-hyp]
    Hypothèse &  $A$ 
  \demostep[demo-use-axiom-1]
    Axiome 1 &  $A \implies B$ 
  \demostep
    m.p. sur
    \explref{demo-my-hyp}
    et
    \explref{demo-use-axiom-1}
    &  $B$ 
\end{demotab}

```

1	Hypothèse	A
2	Axiome 1	$A \implies B$
3	m.p. sur 1 et 2	B

Remarque. Prendre bien garde au fait que ce mécanisme utilise les macros `\label` et `\ref` de L^AT_EX. On travaille donc avec des références globalement au document compilé.

Exemple 3 – Indiquer ce que l’on cherche à faire

Les clés optionnelles `hyps` pour plusieurs hypothèses, `hyp` pour une seule hypothèse et `ccl` pour la conclusion permettent d’expliquer ce que l’on démontre et sous quel contexte.

<pre> \begin{demotab}[hyp = \$A\$, ccl = \$B\$] \demostep Hypothèse & \$A\$ \demostep Axiome 1 & \$A\$ \implies \$B\$ \demostep m.p. sur \explref*{1} et \explref*{2} & \$B\$ \end{demotab} </pre>	<p>Démonstration sous l'hypothèse : A</p> <table> <tr> <td>1</td> <td>Hypothèse</td> <td>A</td> </tr> <tr> <td>2</td> <td>Axiome 1</td> <td>$A \implies B$</td> </tr> <tr> <td>3</td> <td>m.p. sur 1 et 2</td> <td>B</td> </tr> </table> <p>Conclusion : B</p>	1	Hypothèse	A	2	Axiome 1	$A \implies B$	3	m.p. sur 1 et 2	B
1	Hypothèse	A								
2	Axiome 1	$A \implies B$								
3	m.p. sur 1 et 2	B								

Remarque. Aucune des clés `hyps`, `hyp` et `ccl` n’est obligatoire. Par contre il n’est pas possible d’utiliser à la fois les clés `hyps` et `hyp`.

2. Un tableau sur plusieurs pages

Un tableau devant utiliser plusieurs pages sera scindé comme ci-dessous sans perte d’information⁷.

1	Hypothèse	A
2	Axiome 1	$A \implies B$
Suite de la démo. page suivante...		
1		
3	m.p. sur 1 et 2	B
4	1 et 3	$A \wedge B$

3. Un tableau pour le collège et le lycée

Exemple 1 – Avec les réglages par défaut

L’environnement étoilé `demotab*` est différent de l’environnement `demotab` puisqu’il sert à indiquer trois choses et non juste deux comme le montre l’exemple suivant⁸. Par contre, la syntaxe est très similaire. Notez au passage la possibilité d’utiliser `\newline` pour forcer un retour à la ligne dans une cellule et aussi la nécessité d’écrire les accolades de la macro sans argument `\demostep` lorsque la 1^{re} case est vide (*ceci est inutile lorsque l’argument optionnel est renseigné comme nous allons le vérifier dans l’exemple juste après*).

<pre> \begin{demotab*} \demostep \$ABC\$ est un triangle \newline équilatéral & Définition d'un triangle \newline équilatéral. & \$AB = BC = AC\$ </pre>
--

7. Tout le travail est fait par l’environnement `longtable` du package éponyme.

8. C’est pour cela qu’est proposé une version étoilée de l’environnement et non l’utilisation d’une option de l’environnement non étoilé.

```

\demostep{} % --> Ne pas oublier ici !
& Voir l'énoncé.
&  $AB = 10$  \, cm$
\demostep
  Voir les conséquences \newline \explref*{1} et \explref*{2} .
& Simple calcul.
&  $ABC$  a pour périmètre  $30$  \, cm$.
\end{demotab*}

```

Réf.	Je sais que...	Propriété ou fait utilisé	Conséquence
1	ABC est un triangle équilatéral	Définition d'un triangle équilatéral.	$AB = BC = AC$
2		Voir l'énoncé.	$AB = 10\text{ cm}$
3	Voir les conséquences 1 et 2 .	Simple calcul.	ABC a pour périmètre 30 cm .

Exemple 2 – Avec toutes les options

Le système de référence marche ici aussi. Par contre `demotab*` ne propose que `start` comme clé optionnelle avec le même fonctionnement que pour `demotab`.

```

\begin{demotab*}[start = last]
\demostep[demo-first-geo-fact]
   $ABC$  est un triangle \newline équilatéral
& Définition d'un triangle \newline équilatéral.
&  $AB = BC = AC$ 
\demostep[known-data]
& Voir l'énoncé.
&  $AB = 10$  \, cm$
\demostep
  Voir les conséquences \newline
  \explref{demo-first-geo-fact} et \explref{known-data} .
& Simple calcul.
&  $ABC$  a pour périmètre  $30$  \, cm$.
\end{demotab*}

```

Réf.	Je sais que...	Propriété ou fait utilisé	Conséquence
4	ABC est un triangle équilatéral	Définition d'un triangle équilatéral.	$AB = BC = AC$
5		Voir l'énoncé.	$AB = 10\text{ cm}$
6	Voir les conséquences 4 et 5 .	Simple calcul.	ABC a pour périmètre 30 cm .

4. Un tableau sur plusieurs pages

Un tableau devant utiliser plusieurs pages sera scindé comme ci-dessous sans perte d'information ⁹.

Réf.	Je sais que...	Propriété ou fait utilisé	Conséquence
1	ABC est un triangle équilatéral	Définition d'un triangle équilatéral.	$AB = BC = AC$
2		Voir l'énoncé.	$AB = 10\text{ cm}$
3	Voir les conséquences 1 et 2 .	Simple calcul.	ABC a pour périmètre 30 cm .
<i>Suite de la démo. page suivante...</i>			
1			
Réf.	Je sais que...	Propriété ou fait utilisé	Conséquence
4	ABC est un triangle équilatéral	Définition d'un triangle équilatéral.	$AB = BC = AC$
5		Voir l'énoncé.	$AB = 10\text{ cm}$
6	Voir les conséquences 4 et 5 .	Simple calcul.	ABC a pour périmètre 30 cm .

9. Tout le travail est fait par l'environnement `longtable` du package éponyme.

X. Historique

Nous ne donnons ici qu'un très bref historique récent ¹⁰ de **tnslog** à destination de l'utilisateur principalement. Tous les changements sont disponibles uniquement en anglais dans le dossier **change-log** : voir le code source de **tnslog** sur **github**.

2021-03-04 Nouvelle version mineure **1.1.0-beta**.

- **LOGIQUE** : pour les macros `\eq*` et `\neq*`, on a juste gardé les versions symboliques pour la définition et l'identité.
-

2021-03-01 Nouvelle version majeure **1.0.0-beta**.

- **CALCULS ET DÉMONSTRATIONS PAS À PAS.**
 - L'environnement `demoexplain` a été renommé `demotab`.
 - L'environnement `explain` a été renommé `stepcalc`.
 - Le nouvel environnement `stepcalc` dispose d'un nouveau style `ar*` pour placer les opérateurs dans la marge.
 - **LOGIQUE.**
 - `\liesimp` a été renommée `\becauseof`.
 - La façon de décorer les opérateurs de comparaison ou de logique a changé et est plus généraliste.
 - Les macros `\eq*` et `\neq*` proposent des versions symboliques pour certains textes de décoration.
-

2020-08-08 Nouvelle version mineure **0.1.0-beta**.

- **MACROS « TEXTUELLES »** : le préfixe `txt` remplace l'ancien `text`.
-

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

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

XI. Toutes les fiches techniques

1. Espace après la négation logique

`\neg`
`\stdneg` (pour retrouver le symbole par défaut)

2. Personnaliser les opérateurs de comparaison

`\eq [#opt]`
`\neq [#opt]`

— Option: un texte de décoration éventuel. On obtient des versions symboliques si l'on utilise l'un des textes suivants.

1. ? 2. appli 3. cons 4. def 5. id

<code>\less [#opt]</code>	<code>\leq [#opt]</code>	<code>\gtr [#opt]</code>	<code>\geq [#opt]</code>
<code>\nless [#opt]</code>	<code>\nleq [#opt]</code>	<code>\ngtr [#opt]</code>	<code>\ngeq [#opt]</code>

— Option: un texte de décoration éventuel.

3. Équivalences et implications

i. Des symboles logiques supplémentaires

<code>\iff [#opt]</code>	<code>\implies [#opt]</code>	<code>\becauseof [#opt]</code>
<code>\niff [#opt]</code>	<code>\nimplies [#opt]</code>	<code>\nbecauseof [#opt]</code>

— Option: un texte de décoration éventuel.

ii. Équivalences et implications verticales

<code>\viff</code>	<code>\vimplies</code>	<code>\vbecauseof</code>
<code>\nviff</code>	<code>\nvimplies</code>	<code>\nvbecauseof</code>

4. Des versions alternatives du quantificateur existentiel

`\existmulti {#1}`
`\nexistmulti{#1}`

— Argument 1: une écriture mathématique servant à préciser la portée du quantificateur.

`\existsone`
`\nexistsone`

5. Détailler un raisonnement simple

i. Détailler un raisonnement simple

```
\begin{stepcalc} [#opt]
...
\end{stepcalc}
```

— **Option**: la valeur utilise une syntaxe de type clé-valeur. Voici les différentes clés disponibles.

1. **ope** sert à définir l'opérateur utilisé dans tout l'environnement qui sera rédigé en mode mathématique. La valeur par défaut est `{=}` (*et non juste = attention*).
2. **style** sert à définir le style de mise en forme. Voici les différentes valeurs possibles.
 - (a) **u**, la valeur par défaut, est pour **u**-niversity.
 - (b) **ar** est pour **ar**-row.
 - (c) **ar*** est similaire à **ar** mais avec l'opérateur dans la marge.
 - (d) **sar** est pour **s**-hort **ar**-row.
3. **com** permet de demander l'alignement ou non des commentaires non étoilés entre eux.
 - (a) **nal**, la valeur par défaut, est pour **n**-ot **al**-igned.
 - (b) **al** est pour **al**-igned.

```
\explnext [#opt] {#1} expl = expl-ain
```

— **Option**: le symbole à utiliser pour une explication, la valeur par défaut étant celle du symbole de l'environnement `stepcalc` où `\explnext` est utilisé.

— **Argument**: le texte de l'explication qui peut être vide si aucune explication n'est à afficher.

ATTENTION! La macro `\explnext` est à utiliser sans argument ni option au tout début du contenu de l'environnement `stepcalc` en cas d'utilisation du style `sar`.

```
\explnext* [#opt] {#1..#2}
```

— **Option**: le symbole à utiliser pour une explication, la valeur par défaut étant celle du symbole de l'environnement `stepcalc` où `\explnext` est utilisé.

— **Argument 1**: le texte de l'explication pour la 1^{re} ligne. Ce texte peut être vide (*voir l'environnement `astepcalc` pour la raison de ceci*).

— **Argument 2**: le texte de l'explication pour la 2^e ligne. Ce texte peut être vide (*voir l'environnement `astepcalc` pour la raison de ceci*).

```
\comthis {#1} com = com-ment
\comthis*{#1}
```

— **Argument**: le texte d'un court commentaire.

ii. Détailler un raisonnement simple – Mise en forme du texte

Les macros suivantes sont juste utilisées par l'environnement `stepcalc`.

```
\expltxtspacein{}
```

`\expltxt{#1}`

— Argument: le texte de l'explication que l'on veut mettre en forme.

`\expltxtupdown{#1}`

— Argument: le texte de l'explication du haut vers le bas que l'on veut mettre en forme.

`\expltxtup{#1}`

— Argument: le texte de l'explication du bas vers le haut que l'on veut mettre en forme.

`\expltxtupdown{#1..#2}`

— Argument 1: le texte de l'explication du haut vers le bas que l'on veut mettre en forme.

— Argument 2: le texte de l'explication du bas vers le haut que l'on veut mettre en forme.

`\explcom{#1}`

— Argument: le texte d'un court commentaire.

6. Détailler un « vrai » raisonnement

i. Détailler un « vrai » raisonnement via un tableau

`\begin{demotab} [#opts]`

...

`\end{demotab}`

— Clé "start": le début de la numérotation des identifiants des justifications. La valeur par défaut est 1 et la valeur spéciale `last` permet de reprendre la numérotation là où elle s'était arrêtée le dernier environnement `demotab` ou `demotab*` utilisé.

— Clé "hyps ": les hypothèses, au format texte, vérifiées au départ. Cet argument peut être vide et ne doit pas rentrer en conflit avec l'option `hyp`.

— Clé "hyp ": une unique hypothèse, au format texte, vérifiée au départ. Cet argument peut être vide et ne doit pas rentrer en conflit avec l'option `hyps`.

— Clé "ccl ": la conclusion, au format texte, du raisonnement détaillé. Cet argument peut être vide.

`\begin{demotab*} [#opt]`

...

`\end{demotab*}`

— Clé "start": le début de la numérotation des identifiants des justifications. La valeur par défaut est 1 et la valeur spéciale `last` permet de reprendre la numérotation là où elle s'était arrêtée le dernier environnement `demotab` ou `demotab*` utilisé.

`\demostep[#opt]`

— **Option**: un texte qui sera utilisé comme label global référençant le numéro d'une justification.

`\explref{#1}` ref = ref-erence

— **Argument**: un numéro de 1 ou 2 chiffres qui sera encadré comme le sont les numérotations des indications.

`\explref*{#1}`

— **Argument**: un texte correspondant à un label global référençant le numéro d'une justification.

ii. Détailler un « vrai » raisonnement via un tableau - Textes utilisés

<code>\txtdemoID{}</code>	ID = ID-entifier
<code>\txtdemoKNOWN{}</code>	
<code>\txtdemoPROP{}</code>	PROP = PROP-osition
<code>\txtdemoCONS{}</code>	CONS = CONS-equence
<code>\txtdemoHYPS{}</code>	HYPS = HYP-othesis (S-everal ones)
<code>\txtdemoHYP{}</code>	HYP = HYP-othesis (just one)
<code>\txtdemoCCL{}</code>	CCL = C-on-CL-usion
<code>\txtdemoNEXTPAGE{}</code>	