Le package `tnslog` : rédiger de la logique simple

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

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

Christophe BAL

2020-07-10

Table des matières

1 Introduction

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

2 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.

```
$\neg A = \stdneg A$

$\neg\neg A = \stdneg \stdneg A$
```

$\neg A = \neg A$
$\neg\neg A = \neg\neg A$

3 Différents types de comparaisons « standard »

D'un point de vue pédagogique, il peut être intéressant de disposer de différentes façon d'écrire une égalité, une non égalité ou une inégalité. Bien entendu on tord les règles de typographie avec ce type de pratique mais c'est pour le bien de la communauté éducative.

3.1 Définir quelque chose

L'exemple suivant montre deux façons de rédiger une égalité signifiant une définition *(la section 3.9 explique comment est défini le texte* « déf »*)*.

```
$f(x) \eqdef x^3 + 1$

$f(x) \eqdef* x^3 + 1$
```

$f(x) \overset{\text{déf}}{=} x^3 + 1$
$f(x) \coloneqq x^3 + 1$

3.2 Indiquer une identité

L'exemple suivant montre deux façons de rédiger des identités avec une notation symbolique non standard *(la section 3.9 explique comment est défini le texte* « id »*)*.

```
$(a + b)^2 \eqid a^2 + b^2 + 2 a b$

$(a + b)^2 \eqid* a^2 + b^2 + 2 a b$
```

$(a + b)^2 \overset{\text{id}}{=} a^2 + b^2 + 2ab$
$(a + b)^2 \rightleftharpoons a^2 + b^2 + 2ab$

3.3 Une égalité à vérifier ou non, une hypothèse, une condition

Se reporter à la section 3.9 pour savoir comment sont définis les textes *« cons »* , *« cond »* et *« hyp »*.

```
$(a + b)^3 \eqtest a^3 + b^3 + 3 a b$

$(a + b)^3 \neqid a^3 + b^3 + 3 a b$

$x \neqhyp 0$  ou
$x \neqcond 0$ ou
$x \eqcons 0$
```

$(a+b)^3 \overset{?}{=} a^3+b^3+3ab$
$(a+b)^3 \overset{\text{id}}{\neq} a^3+b^3+3ab$
$x \overset{\text{hyp}}{\neq} 0$ ou $x \overset{\text{cond}}{\neq} 0$ ou $x \overset{\text{cons}}{=} 0$

3.4 Une égalité indiquant le choix d'une valeur ou l'application d'une relation

La section 3.9 permet de savoir comment les textes *« choix »* et *« appli »* sont définis.

```
$x \geqcond 4$ implique
$x^2 \geqcons 16$.

Donc $x \eqchoice 123$ donne
$123^2 \geqappli 16$.
```

$x \overset{\text{cond}}{\geq} 4$ implique $x^2 \overset{\text{cons}}{\geq} 16$.
Donc $x \overset{\text{choix}}{=} 123$ donne $123^2 \overset{\text{appli}}{\geq} 16$.

3.5 Une égalité indiquant l'équation d'une courbe

La section 3.9 permet de savoir comment les texte *« graph »* est défini.

```
$M \in C: y \eqplot x^2 + 3$
donne
$y_M \eqappli x_M^2 + 3$.
```

$M \in C : y \overset{\text{graph}}{=} x^2+3$ donne $y_M \overset{\text{appli}}{=} x_M^2+3$.

3.6 Différents types d'inéquations

Le principe reste le même pour les symboles d'équations excepté qu'il n'y a ici aucune écriture purement symbolique. Voici un code « fourre-tout » montrant quelques exemples.

```
$x \leqtest x^2$ ou $x \lesscons x^2$ ou
$x \geqhyp 1$    ou $x \gtrcond 2$.
```

$x \overset{?}{\leq} x^2$ ou $x \overset{\text{cons}}{<} x^2$ ou $x \overset{\text{hyp}}{\geq} 1$ ou $x \overset{\text{cond}}{>} 2$.

3.7 Des formes négatives aussi pour les inéquations

Tous les opérateurs de comparaison ont une forme négative qui s'obtient en préfixant le nom de l'opérateur par `n`. Voici quelques exemples d'utilisation.

```
$x \nlesshyp 3$ ou
$y \nleqtest 4$ ou
$z \ngeqcons 5$
```

$x \overset{\text{hyp}}{\not<} 3$ ou $y \overset{?}{\not\leq} 4$ ou $z \overset{\text{cons}}{\not\geq} 5$

3.8 Une table récapitulative

La table 1 page 6 fournit toutes les associations autorisées entre opérateurs de comparaison et décorations.

3.9 Textes utilisés

Voici les macros définissant les textes utilisés qui tiennent compte de l'utilisation ou non de l'option `french` de `babel`. Nous ne donnons que les versions françaises.

`\textopappli` donne *« appli »*
`\textopchoice` donne *« choix »*
`\textopcond` donne *« cond »*
`\textopcons` donne *« cons »*
`\textopdef` donne *« déf »*
`\textophyp` donne *« hyp »*
`\textopid` donne *« id »*
`\textopplot` donne *« graph »*
`\textoptest` donne *« ? »*

4 Équivalences et implications

4.1 Des symboles logiques supplémentaires

Exemple 1 – Implication réciproque

En plus des opérateurs `\iff` et `\implies` proposés par LaTeX, il a été ajouté l'opérateur `\liesimp`, où l'on a inversé les groupes syllabiques de `\implies`, un opérateur pour pour obtenir $\Longleftarrow$[1], ainsi que des versions négatives. Voici un exemple d'utilisation.

```
$(A \implies B)
 \iff (B \liesimp A)$

$(A \implies B)
 \niff (A \nimplies B)$
```

$$(A \implies B) \iff (B \Longleftarrow A)$$
$$(A \implies B) \nLeftrightarrow (A \nRightarrow B)$$

Exemple 2 – Des opérateurs décorés

Tout comme pour les comparaisons, il existe des versions décorées de type test, hypothèse, condition ... Elles sont toutes présentes dans l'exemple suivant.

```
$A \iffappli B \niffchoice C$

$A \impliescond B \nimpliescons C$

$A \liesimphyp B \nliesimptest C$
```

$$A \overset{\text{appli}}{\iff} B \overset{\text{choix}}{\nLeftrightarrow} C$$
$$A \overset{\text{cond}}{\implies} B \overset{\text{cons}}{\nRightarrow} C$$
$$A \overset{\text{hyp}}{\Longleftarrow} B \overset{?}{\nLeftarrow} C$$

4.2 Une table récapitulative

La table 1 page suivante montre toutes les associations autorisées entre opérateurs logiques et décorations.

4.3 Équivalences et implications verticales

À quoi cela sert-il ?

Les sections 6.1 et 6.2 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

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

et d'implication. Voici comment les obtenir *(tous les cas possibles ont été indiqués)*. Bien entendu le préfixe `v` est pour `v`-ertical.

```
\begin{tabular}{cccccc}
    $A$           & $B$
  & $C$           & $D$
  & $E$           & $F$
  \\
    $\viff$       & $\vimplies$
  & $\vliesimp$   & $\nviff$
  & $\nvimplies$ & $\nvliesimp$
  \\
    $A$           & $B$
  & $C$           & $D$
  & $E$           & $F$
\end{tabular}
```

$$\begin{array}{cccccc} A & B & C & D & E & F \\ \Updownarrow & \Downarrow & \Uparrow & \not\Updownarrow & \not\Downarrow & \not\Uparrow \\ A & B & C & D & E & F \end{array}$$

4.4 Tables des décorations possibles des opérateurs

La table 1 de la présente page donne toutes les associations autorisées entre opérateurs et décorations. Il faut retenir que les versions étoilées produisent des écritures symboliques.

TABLE 1 – Décorations

Préfixe	appli	choice	cond	cons	def	def*	hyp	id	id*	plot	test
\eq	×	×	×	×	×	×	×	×	×	×	×
\neq	×	×	×	×			×	×		×	×
\less \nless \leq \nleq \gtr \ngtr \geq \ngeq	×	×	×	×			×			×	×
\iff \niff \implies \nimplies \liesimp \nliesimp	×	×	×	×			×				×

5 Des versions alternatives du quantificateur existentiel

5.1 Quantifier l'existence

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

```
$\existsone x \in E$ pour
\og il existe un seul $x$ \fg.

$\existmulti{\leq 1} y \in F$ pour
\og il existe au plus un $y$ \fg.

$\existmulti{=1} \eqdef* \existsone$
```

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

5.2 Versions négatives

```
$\nexistsone$ pour
\og il n'existe pas un unique \fg.

$\existmulti{\neq1} \eqdef* \nexistsone$

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

$\nexists$ vient de \verb+amssymb+.
```

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

6 Détailler un raisonnement simple

6.1 Version pour le lycée et après

Exemple 1 – Avec les réglages par défaut

L'environnement `explain` 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 anglais [2]. On dispose aussi de `\explnext*` pour des explications descendantes et/ou montantes [3].

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 2e façon de détailler des calculs et/ou un raisonnement simple est proposée plus bas dans la section 6.2.

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.

```
\begin{explain}
    (a + b)^2
        \explnext{On utilise $x^2 = x \cdot x$.}
    (a + b) (a + b)
        \explnext*{Double développement depuis la parenthèse gauche.}%
                  {Double factorisation pas facile.}
    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{explain}
```

$(a+b)^2$
$=$ { *On utilise* $x^2 = x \cdot x$. }
$(a+b)(a+b)$
$=$ { ↓ *Double développement depuis la parenthèse gauche.* ↓ / ↑ *Double factorisation pas facile.* ↑ }
$a^2 + ab + ba + b^2$
$=$ { ↑ *Commutativité du produit.* ↑ }
$a^2 + 2ab + b^2$
$=$ { ↓ *Commutativité de l'addition.* ↓ }
$a^2 + b^2 + 2ab$

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

```
Un calcul avec un placement pouvant être
utile :
\begin{explain}
    (a + b)^2
        \explnext{Identité remarquable.}
    a^2 + b^2 + 2 a b
\end{explain}
```

Un calcul avec un placement pouvant être utile : $(a+b)^2$
$=$ { *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{explain}
    (a + b)^2
        \explnext{Identité remarquable.}
    a^2 + b^2 + 2 a b
\end{explain}
```

Mon calcul pas trop proche.

$(a+b)^2$
$=$ { *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 `explain` possède plusieurs options dont l'une est `ope` qui vaut `{=}` par défaut. Ceci permet de faire ce qui suit sans effort.

```
\begin{explain}[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{explain}
```

$$
\begin{aligned}
& x^2 + 10x + 25 = 0 \\
\Updownarrow \quad & \{\textit{ Identité remarquable. }\} \\
& (x+5)^2 = 0 \\
\Updownarrow \quad & \{\, P^2 = 0 \textit{ si et seulement si } P = 0. \,\} \\
& x = -5
\end{aligned}
$$

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

$$
\begin{aligned}
& a^2 = b^2 \\
\Updownarrow & \\
& a = \pm b
\end{aligned}
$$

Exemple 4 – Utiliser un autre symbole localement

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

```
\begin{explain}[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{explain}
```

$$
\begin{aligned}
& 0 \leq a < b \\
\Downarrow \quad & \{\textit{ Croissance de } x^2 \textit{ sur } R_+. \,\} \\
& a^2 < b^2 \\
\Updownarrow & \\
& a^2 - b^2 < 0 \\
\Updownarrow \quad & \{\textit{ Identité remarquable. }\} \\
& (a-b)(a+b) < 0 \\
\Downarrow & \\
& a \neq b
\end{aligned}
$$

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`, `\expltxtdown` et `\expltxtupdown`. Par défaut le package utilise les définitions suivantes.

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

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

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

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

```
\begin{explain}
    (a + b) (a + b)
        \explnext{Se souvenir de $P\cdot
    P = P^2$.}%
    (a + b)^2
        \explnext*{Id. Rm. - Dév.}%
                  {Id. Rm. - Facto.}
    a^2 + 2 a b + b^2
\end{explain}
```

$$
\begin{aligned}
&(a+b)(a+b)\\
&= \quad \Downarrow \textit{ Se souvenir de } P\cdot P = P^2\textit{.} \Uparrow\\
&(a+b)^2\\
&= \quad \left\langle \begin{array}{c} \Downarrow \textit{ Id. Rm. - Dév. } \Downarrow \\ \hline \Uparrow \textit{ Id. Rm. - Facto. } \Uparrow \end{array} \right\rangle\\
&a^2+2ab+b^2
\end{aligned}
$$

La mise en forme a été obtenue en utilisant le code LaTeX suivant où la macro `\samesizeas{#1}{#2}` rend le texte `#1` aussi large que `#2` en ajoutant des espaces supplémentaires tout en centrant le résultat 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\expltxtdown[1]{%
    \myexpltxt{red}{$\Downarrow$ #1 $\Downarrow$}%
}

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

6.2 Version pour les collégiens

L'environnement `explain` avec l'option `style = ar`[4] utilise des flèches pour indiquer les explications *(`ar` est pour `ar`-row soit* « flèche » *en anglais)*. Dans ce cas d'utilisation, la macro `\explnext*` permet d'avoir une flèche unidirectionnelle, vers le haut ou le bas au choix, ou bien d'écrire deux indications dont l'une est montante et l'autre descendante.

Il existe aussi l'option `style = sar` lorsque la toute 1re étape n'est pas expliquée *(`s` est pour `s`-hort soit* « court » *en anglais)*. Attention car forcément ceci nécessite au tout début de l'environnement l'usage de la macro `\explnext` sans aucun contenu !

Exemple 1 – Des flèches à double sens

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

$$
\begin{aligned}
&(a+b)^2 \\
&= a^2 + 2ab + b^2 \quad \text{\textit{Identité remarquable}} \\
&= a^2 + b^2 + 2ab
\end{aligned}
$$

Exemple 2 – Des flèches unidirectionnelles

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

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

```
\begin{explain}[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{explain}
```

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

Exemple 3 – Ne pas expliquer le tout début

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

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

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

Exemple 4 – 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)*.

```
\begin{explain}[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{explain}
```

$$\begin{aligned} & a^2 + 2ab + b^2 = 0 \\ \iff\ & (a+b)^2 = 0 \\ \implies\ & a + b = 0 \end{aligned} \quad \text{⤵ } P^2 = 0 \text{ ssi } P = 0.$$

Avec la version courte, on obtient ce qui suit.

```
\begin{explain}[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{explain}
```

$$\begin{aligned} a^2 + 2ab + b^2 = 0 & \iff (a+b)^2 = 0 \\ & \implies a + b = 0 \end{aligned} \quad \text{⤵ } P^2 = 0 \text{ ssi } P = 0.$$

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

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

Remarque. 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{explain}[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{explain}
```

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

Exemple 3 – Le meilleur des deux mondes

Dans d'autres situations, utilisez 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.

```
\begin{explain}[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{explain}
```

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

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.

```
\begin{explain}[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{explain}
```

$$
\begin{aligned}
&(a+b)(a+b) && [\ \textit{Forme facto.}\ ] && \big\} \textit{Via } x^2 = x\cdot x.\\
&= (a+b)^2 \qquad [\ \textit{Au passage...}\ ] && && \big\} \textit{Id.Rq - Dév.} \quad \big\{ \textit{Id.Rq - Facto.}\\
&= a^2+2ab+b^2 && [\ \textit{Forme dév.}\ ]
\end{aligned}
$$

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

6.4 Un mini hack très utile pour des *« étapes alignées »*

Vous pouvez écrire très facilement des calculs ou raisonnement simples alignés comme suit sans trop vous fatiguez.

```
\begin{explain}[style = sar]
    (a + b) (a + b)
        \explnext{}
    (a + b)^2
        \explnext{}
    a^2 + b^2 + 2 a b
        \comthis{Pourquoi ?}
        \explnext{}
    a^2 + 2 a b + b^2
\end{explain}
```

$$\begin{aligned}(a+b)(a+b) &= (a+b)^2 \\ &= a^2+b^2+2ab \qquad [\textit{ Pourquoi ? }] \\ &= a^2+2ab+b^2\end{aligned}$$

On a accès à une autre mise en forme *(ceci peut rendre aussi service)*.

```
\begin{explain}[style = ar]
    (a + b) (a + b)
        \explnext{}
    (a + b)^2
        \explnext{Pourquoi ?}
    a^2 + b^2 + 2 a b
        \explnext{}
    a^2 + 2 a b + b^2
\end{explain}
```

$$\begin{aligned}&(a+b)(a+b) \\ &= (a+b)^2 \\ &= a^2+b^2+2ab \quad \textit{Pourquoi ?} \\ &= a^2+2ab+b^2\end{aligned}$$

Enfin dans le cadre de calculs à faire expliquer par des élèves, ce qui suit peut être utile.

```
Donner les justifications J1, J2 et J3.

\medskip
\begin{explain}
    (a + b) (a + b)
        \explnext{J1}
    (a + b)^2
        \explnext{J2}
    a^2 + b^2 + 2 a b
        \explnext{J3}
    a^2 + 2 a b + b^2
\end{explain}
```

Donner les justifications J1, J2 et J3.

$$\begin{aligned}&(a+b)(a+b) \\ &= \quad \{\textit{ J1}\} \\ &(a+b)^2 \\ &= \quad \{\textit{ J2}\} \\ &a^2+b^2+2ab \\ &= \quad \{\textit{ J3}\} \\ &a^2+2ab+b^2\end{aligned}$$

6.5 Un conseil de mise en forme

Voici un style de codage que nous trouvons très facile à relire et maintenir.

```
\begin{explain}[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{explain}
```

$(a+b)(a+b)$ [*Forme facto.*]
$=$ { *Via* $x^2 = x \cdot x$. }
$(a+b)^2$ [*Au passage...*]
$=$ { ↓ *Id.Rq - Dév.* ↓ ; ↑ *Id.Rq - Facto.* ↑ }
$a^2 + 2ab + b^2$ [*Forme dév.*]

7 Détailler un « vrai » raisonnement

7.1 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 7.3 pour un autre type de présentation plus adapté à un public de collège ou de lycée)*. Ci-dessous l'environnement `demoexplain` facilite la mise en page [5] et la macro étoilée `\explref*` permet d'indiquer une référence interne au raisonnement [6]. 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{demoexplain}
    \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{demoexplain}
```

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 1er numéro de justification via la clé `start` : la valeur spéciale `last` indique de continuer la numérotation à la suite.

5. En coulisse est utilisé l'environnement `longtable` du package éponyme.
6. Les indications peuvent être numérotes jusqu'à 99 ce qui est bien au-delà des besoins pratiques.

```
\begin{demoexplain}[start = last]
    \demostep
        Axiome 3
      & $(A \wedge B) \implies C$
    \demostep
        m.p. sur
        \explref*{4} et \explref*{5}
      & $C$
\end{demoexplain}
```

5	Axiome 3	$(A \wedge B) \implies C$
6	m.p. sur 4 et 5	C

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{demoexplain}
    \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{demoexplain}
```

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 LaTeX. 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.

```
\begin{demoexplain}[hyp = $A$, ccl = $B$]
    \demostep
        Hypothèse & $A$
    \demostep
        Axiome 1  & $A \implies B$
    \demostep
        m.p. sur
        \explref*{1} et \explref*{2}
      & $B$
\end{demoexplain}
```

Démonstration sous l'hypothèse : A

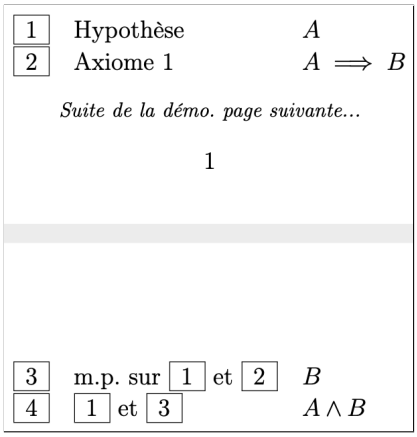
1	Hypothèse	A
2	Axiome 1	$A \implies B$
3	m.p. sur 1 et 2	B

Conclusion : 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`.

7.2 Un tableau sur plusieurs pages

Un tableau devant utiliser plusieurs pages sera scindé comme ci-dessous sans perte d'information [7].



7.3 Un tableau pour le collège et le lycée

Exemple 1 – Avec les réglages par défaut

L'environnement étoilé `demoexplain*` est différent de l'environnement `demoexplain` puisqu'il sert à indiquer trois choses et non juste deux comme le montre l'exemple suivant [8]. 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 1re case est vide *(ceci est inutile lorsque l'argument optionnel est renseigné comme nous allons le vérifier dans l'exemple juste après)*.

```
\begin{demoexplain*}
    \demostep
        $ABC$ est un triangle \newline équilatéral
      & Définition d'un triangle \newline équilatéral.
      & $AB = BC = AC$
    \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{demoexplain*}
```

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.	$AB = 10\,cm$
3	Voir les conséquences 1 et 2 .	Simple calcul.	ABC a pour périmètre $30\,cm$.

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.

Exemple 2 – Avec toutes les options

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

```
\begin{demoexplain*}[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{demoexplain*}
```

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.	$AB = 10\,cm$
6	Voir les conséquences 4 et 5 .	Simple calcul.	ABC a pour périmètre $30\,cm$.

7.4 Un tableau sur plusieurs pages

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

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

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.	$AB = 10\,cm$
3	Voir les conséquences 1 et 2 .	Simple calcul.	ABC a pour périmètre $30\,cm$.

Suite de la démo. page suivante...

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.	$AB = 10\,cm$
6	Voir les conséquences 4 et 5 .	Simple calcul.	ABC a pour périmètre $30\,cm$.

8 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`.

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

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

9 Toutes les fiches techniques

9.1 Espace après la négation logique

\neg <macro> (Sans argument)
\stdneg <macro> (Sans argument)

9.2 Différents types de comparaisons « standard »

9.2.1 Opérateurs décorés – Les textes

\textopappli <macro> (Sans argument)
\textopchoice <macro> (Sans argument)
\textopcond <macro> (Sans argument)
\textopcons <macro> (Sans argument)
\textopdef <macro> (Sans argument)
\textophyp <macro> (Sans argument)
\textopid <macro> (Sans argument)
\textopplot <macro> (Sans argument)
\textoptest <macro> (Sans argument)

9.2.2 Opérateurs de comparaison supplémentaires

\eqdef <macro> (Sans argument)
\eqdef* <macro> (Sans argument)
\eqid <macro> (Sans argument)
\eqid* <macro> (Sans argument)
\eqplot <macro> (Sans argument)
\eqappli <macro> (Sans argument)
\eqchoice <macro> (Sans argument)
\eqcond <macro> (Sans argument)
\eqcons <macro> (Sans argument)
\eqhyp <macro> (Sans argument)
\eqtest <macro> (Sans argument)

\neqid <macro> (Sans argument)
\neqplot <macro> (Sans argument)
\neqappli <macro> (Sans argument)
\neqchoice <macro> (Sans argument)
\neqcond <macro> (Sans argument)
\neqcons <macro> (Sans argument)
\neqhyp <macro> (Sans argument)
\neqtest <macro> (Sans argument)

\lessplot <macro> (Sans argument)
\lessappli <macro> (Sans argument)
\lesschoice <macro> (Sans argument)
\lesscond <macro> (Sans argument)
\lesscons <macro> (Sans argument)

\lesshyp <macro> (Sans argument)
\lesstest <macro> (Sans argument)

\nlessplot <macro> (Sans argument)
\nlessappli <macro> (Sans argument)
\nlesschoice <macro> (Sans argument)
\nlesscond <macro> (Sans argument)
\nlesscons <macro> (Sans argument)
\nlesshyp <macro> (Sans argument)
\nlesstest <macro> (Sans argument)

\leqplot <macro> (Sans argument)
\leqappli <macro> (Sans argument)
\leqchoice <macro> (Sans argument)
\leqcond <macro> (Sans argument)
\leqcons <macro> (Sans argument)
\leqhyp <macro> (Sans argument)
\leqtest <macro> (Sans argument)

\nleqplot <macro> (Sans argument)
\nleqappli <macro> (Sans argument)
\nleqchoice <macro> (Sans argument)
\nleqcond <macro> (Sans argument)
\nleqcons <macro> (Sans argument)
\nleqhyp <macro> (Sans argument)
\nleqtest <macro> (Sans argument)

\gtrplot <macro> (Sans argument)
\gtrappli <macro> (Sans argument)
\gtrchoice <macro> (Sans argument)
\gtrcond <macro> (Sans argument)
\gtrcons <macro> (Sans argument)
\gtrhyp <macro> (Sans argument)
\gtrtest <macro> (Sans argument)

\ngtrplot <macro> (Sans argument)
\ngtrappli <macro> (Sans argument)
\ngtrchoice <macro> (Sans argument)
\ngtrcond <macro> (Sans argument)
\ngtrcons <macro> (Sans argument)
\ngtrhyp <macro> (Sans argument)
\ngtrtest <macro> (Sans argument)

\geqplot <macro> (Sans argument)
\geqappli <macro> (Sans argument)

\geqchoice <macro> (Sans argument)
\geqcond <macro> (Sans argument)
\geqcons <macro> (Sans argument)
\geqhyp <macro> (Sans argument)
\geqtest <macro> (Sans argument)

9.3 Équivalences et implications

9.3.1 Des symboles logiques supplémentaires

\iff <macro> (Sans argument)
\iffappli <macro> (Sans argument)
\iffchoice <macro> (Sans argument)
\iffcond <macro> (Sans argument)
\iffcons <macro> (Sans argument)
\iffhyp <macro> (Sans argument)
\ifftest <macro> (Sans argument)

\niff <macro> (Sans argument)
\niffappli <macro> (Sans argument)
\niffchoice <macro> (Sans argument)
\niffcond <macro> (Sans argument)
\niffcons <macro> (Sans argument)
\niffhyp <macro> (Sans argument)
\nifftest <macro> (Sans argument)

\implies <macro> (Sans argument)
\impliesappli <macro> (Sans argument)
\implieschoice <macro> (Sans argument)
\impliescond <macro> (Sans argument)
\impliescons <macro> (Sans argument)
\implieshyp <macro> (Sans argument)
\impliestest <macro> (Sans argument)

\nimplies <macro> (Sans argument)
\nimpliesappli <macro> (Sans argument)
\nimplieschoice <macro> (Sans argument)
\nimpliescond <macro> (Sans argument)
\nimpliescons <macro> (Sans argument)
\nimplieshyp <macro> (Sans argument)
\nimpliestest <macro> (Sans argument)

\liesimp <macro> (Sans argument)
\liesimpappli <macro> (Sans argument)
\liesimpchoice <macro> (Sans argument)
\liesimpcond <macro> (Sans argument)
\liesimpcons <macro> (Sans argument)

\liesimphyp <macro> (Sans argument)
\liesimptest <macro> (Sans argument)

\nliesimp <macro> (Sans argument)
\nliesimpappli <macro> (Sans argument)
\nliesimpchoice <macro> (Sans argument)
\nliesimpcond <macro> (Sans argument)
\nliesimpcons <macro> (Sans argument)
\nliesimphyp <macro> (Sans argument)
\nliesimptest <macro> (Sans argument)

9.3.2 Équivalences et implications verticales

\viff <macro> (Sans argument)
\nviff <macro> (Sans argument)
\vimplies <macro> (Sans argument)
\nvimplies <macro> (Sans argument)
\vliesimp <macro> (Sans argument)
\nvliesimp <macro> (Sans argument)

9.4 Des versions alternatives du quantificateur existentiel

\existmulti <macro> (1 Argument)
\nexistmulti <macro> (1 Argument)

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

\existsone <macro> (Sans argument)
\nexistsone <macro> (Sans argument)

9.5 Détailler un raisonnement simple

9.5.1 Détailler un raisonnement simple

explain <env> [1 Option]

— 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 =)*.
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) 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.

ATTENTION ! *La macro* `\explnext` *est à utiliser sans argument au tout début de l'environnement* `aexplain*`.

`\explnext <macro> [1 Option] (1 Argument)` où `expl = expl-ain`

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

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

`\explnext* <macro> [1 Option] (2 Arguments)` où `expl = expl-ain`

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

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

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

`\comthis <macro> (1 Argument)` où `com = com-ment`
`\comthis* <macro> (1 Argument)` où `com = com-ment`

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

9.5.2 Détailler un raisonnement simple – Mise en forme du texte

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

`\expltxtspacein <macro> (Sans argument)`

`\expltxt <macro> (1 Argument)` où `expl = expl-ain`

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

`\expltxtdown <macro> (1 Argument)` où `expl = expl-ain`

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

`\expltxtup <macro> (1 Argument)` où `expl = expl-ain`

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

`\expltxtupdown <macro> (2 Arguments)` où `expl = expl-ain`

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

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

`\explcom <macro> (1 Argument)` où `expl = expl-ain` et `com = com-ment`

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

9.6 Détailler un « vrai » raisonnement

9.6.1 Détailler un « vrai » raisonnement via un tableau

`demoexplain <env> [4 Options]`

— `Option "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 `demoexplain` ou `demoexplain*` utilisé.

— `Option "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`.

— `Option "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`.

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

`demoexplain* <env> [1 Option]`

— `Option "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 `demoexplain` ou `demoexplain*` utilisé.

`\demostep <macro> [1 Option] (Sans argument)`

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

`\explref <macro> (1 Argument)` où `expl = expl-ain` et `ref = ref-erence`

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

`\explref* <macro> (1 Argument)` où `expl = expl-ain` et `ref = ref-erence`

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

9.6.2 Détailler un « vrai » raisonnement via un tableau - Textes utilisés

`\textdemoID <macro> (Sans argument)` où `ID = ID-entifier`
`\textdemoKNOWN <macro> (Sans argument)`
`\textdemoPROP <macro> (Sans argument)` où `PROP = PROP-osition`
`\textdemoCONS <macro> (Sans argument)` où `CONS = CONS-equence`
`\textdemoHYPS <macro> (Sans argument)` où `HYPS = HYP-othesis (S-everal ones)`
`\textdemoHYP <macro> (Sans argument)` où `HYP = HYP-othesis (just one)`
`\textdemoCCL <macro> (Sans argument)` où `CCL = C-on-CL-usion`
`\textdemoNEXTPAGE <macro> (Sans argument)`