

生命保険アクチュアリー記号 $\mathrm{T}_\mathrm{E}X$ パッケージ

v3.1

Ueda, H.

<http://space.geocities.jp/funasking/>

2012/01/15

このパッケージは生命保険数学に必要なアクチュアリー記号を出力するためのコマンドを提供しています。基本的には $\mathrm{L}^A\mathrm{T}_\mathrm{E}X 2_\epsilon$ 標準のパッケージや `amsmath` 等のパッケージですべて補える記号ですが、入力 of 省略と同時に、記号の意味を著者が一目で判断できるようにしておく必要があります。したがって、ここでアクチュアリー記号を与えるコマンドを定義する意義は大いにありと考えております。

なお、このパッケージで与えるコマンドはすべて数式モード内で入力してください。また、出力例ではわざとおかしな出力を与えている箇所がありますが、それは注意のためのものであって、ミスではありません。

また、このパッケージが与えるコマンドの定義は、マクロ初心者の著者がスパイラル方式で作成したものばかりです。したがって、その定義はスマートでもエレガントでもない、泥臭い定義ばかりです。その点が気になる方は、適宜変更して頂いても構いません。その際は、著者に一言アドバイスを頂けるとありがたく存じます。

1 コマンド

1.1 デフォルトコマンド

1.1.1 デフォルトコマンド

予定利率は通常、 i と書かれます。また、現価率は v で表されます。その他の金利計算等の基本的な記号は、特別に定義する必要のない記号ばかりです。以下に一例を示しておきます。

出力例					
意味	出力	入力	意味	出力	入力
予定利率	i	<code>i</code>	現価率	v	<code>v</code>
割引率	d	<code>d</code>	利力	δ	<code>\delta</code>
生存者数	l_x, ℓ_x	<code>l_{x}, \ell_{x}</code>	死亡者数	d_x	<code>d_{x}</code>

出力例		
意味	出力	入力
生存率	p_x	<code>p_{x}</code>
	${}_np_x$	<code>{ }_{n}p_{x}</code>
死亡率	q_x	<code>q_{x}</code>
	${}_nq_x$	<code>{ }_{n}q_{x}</code>
死力	μ_x	<code>\mu_{x}</code>
略算平均余命	e_x	<code>e_{x}</code>
完全平均余命	\mathring{e}_x	<code>\mathring{e}_{x}</code>
絶対 A 脱退率	q_x^{A*}	<code>q_{x}^{\wedge A*}</code>
就業者数	l_x^{aa}	<code>l_{x}^{\wedge aa}</code>
絶対就業不能率	$q_x^{(i)*}$	<code>q_{x}^{\wedge (i)*}</code>

1.1.2 装飾コマンド

L^AT_EX 2_εには標準でさまざまな装飾系のコマンドがあります。このドキュメントでは、アクチュアリー記号に必要であろうコマンドを紹介しておきます。

出力例		
<code>\ddot{a}</code>	引数の上にツードット記号を出力する。	\ddot{a}
<code>\mathring{a}</code>	引数の上にリング記号を出力する。	\mathring{a}
<code>\bar{a}</code>	引数の上にバー記号を出力する。	\bar{a}
<code>\overline{xyz}</code>	引数の上に可変長バー記号を出力する。	\overline{xyz}
<code>\hat{a}</code>	引数の上にハット記号を出力する。	\hat{a}
<code>\widehat{xyz}</code>	引数の上に可変長ハット記号を出力する。	\widehat{xyz}
<code>\smash{a_{\frac{1}{2}}}</code>	引数の高さや深さを 0pt にする。	$a_{\frac{1}{2}} \rightarrow a_{\frac{1}{2}}$

これらの装飾コマンドを複数施すと、記号が正しい位置に付かないことがあります。例えば、`\hat{\hat{A}}`と入力すると、 $\hat{\hat{A}}$ と出力されます。これを解決するためには、`amsmath` パッケージを使用することをお勧めします。`amsmath` パッケージを使用した場合、複数の装飾を施しても正しい位置に記号が出力されるようになっています。

1.2 補助コマンド

`\fracpay` `\fracpay{#1}`

分割払い (fractional payment) を表す記号を出力します。例えば、 $P^{\frac{1}{12}}$ と書くと、 $P^{(12)}$ のように出力されます。また、省略形である `\fp` も用意しました。

使用例			
出力	入力	出力	入力
$P^{(k)}$	<code>P^{\{k\}}</code>	$P^{(k)}$	<code>P^{\fracpay{k}}</code>
$P^{(\infty)}$	<code>P^{\{\infty\}}</code>	$P^{(\infty)}$	<code>P^{\fracpay{\infty}}</code>
$P^{(k)}$	<code>P^{\{fp{k}\}}</code>	$P^{(\infty)}$	<code>P^{\{fp{\infty}\}}</code>

`\step` `\step{#1}`

保険期間を表す記号 \bar{n} を出力します。例えば、 $a_{\frac{1}{2}\bar{n}}$ と書くと、 $a_{\bar{n}}$ のように出力します。

使用例			
出力	入力	出力	入力
$a_{\bar{n}}$	<code>a_{\step{n}}</code>	$A_{x:\bar{n}}$	<code>A_{x:\step{n}}</code>
$a_{\overline{n-f-t}}$	<code>a_{\step{n-f-t}}</code>	$s_{\frac{1}{2}}$	<code>s_{\step{\frac{1}{2}}}</code>
$x^{\frac{1}{1+\frac{1}{2}}}$	<code>x^{\step{\frac{1}{1+\frac{1}{2}}}}</code>	$a_{\overline{s_{\bar{n}}}}$	<code>a_{\step{s_{\step{n}}}}</code>

`\overbracket` `\overbracket{#1}`

引数の上部に、可変長の下向きブラケット (カギ括弧) を出力します。例： \overbracket{A} (`\overbracket{A}`)、 \overbracket{xyzw} (`\overbracket{xyzw}`)、 \overbracket{ABC} (`displaystyle`)、 \overbracket{xy} (`textstyle`)、 $\overbracket{s_{abcd_{efgh}}}$ (`(script-)``scriptstyle`)。

なお、この記号は縦線と横線がそれぞれ異なるパーツで描かれているため、ディスプレイ上では繋ぎ目がはみ出ているように見えますが、印刷や拡大して見てわかるように、実際にははみ出ていません。

`\overtortoise` `\overtortoise{#1}`

引数の上部に、可変長の下向き亀甲括弧 (tortoise shell bracket) を出力します。例： \overtortoise{A} (`\overtortoise{A}`)、 \overtortoise{xyzw} (`\overtortoise{xyzw}`)、 \overtortoise{ABC} (`displaystyle`)、 \overtortoise{xy} (`textstyle`)、 $\overtortoise{s_{abcd_{efgh}}}$ (`(script-)``scriptstyle`)。

なお、この記号は縦線と横線がそれぞれ異なるパーツで描かれているため、ディスプレイ上では繋ぎ目がはみ出ているように見えますが、印刷や拡大して見てわかるように、実際にははみ出ていません。

`\od` `\od[#1]{#2}{#3}`

常微分 (ordinary differential) を表すコマンドです。2 階以上の微分を表す場合は、`[#1]` オプションに階数を入力することで出力できます。デフォルトでは微分記号 d はローマン体となっていますが、`\setDiffSymb` コマンドで変更することができます。これについては後述します。

使用例

出力	入力	出力	入力
$\frac{dy}{dx}$	<code>\od{y}{x}</code>	$\frac{dl_{x+t}}{dt}$	<code>\od{1_{x+t}}{t}</code>
$\frac{d^2}{dx^2}f(x)$	<code>\od[2]{x}f(x)</code>	$\frac{d^ny}{dx^n}$	<code>\od[n]{y}{x}</code>

`\textod` `\textod[#1]{#2}{#3}`

テキスト形式の分数表現を用いた微分記号です。例えば、`\textod[n]{y}{x}`と入力すると d^ny/dx^n のように出力されます。

`\setDiffSymb` `\setDiffSymb{typeface}{symbol}`

このコマンドは、微分記号の書体 *typeface* と記号 *symbol* を変更することができます。書体には *roman* と *italic* を指定することができます。記号は任意ですが、立体 (*roman*) またはイタリック体 (*italic*) が用意されていない記号もありますので注意してください。デフォルトでは立体の *d*、すなわち "d" が設定されています。

入力例：

```
\[ \displaystyle
\od{y}{x},
\setDiffSymb{italic}{\od{y}{x}},
\setDiffSymb{roman}{D} \od{v}{t},
\setDiffSymb{italic}{\delta} \od{L}{t},
\setDiffSymb{roman}{\delta} \od{L}{t},
\setDiffSymb{\partial} \od{f}{x}, \textod{f}{x},
\setDiffSymb{}{\od{y}{x} \}
```

出力例：

$$\frac{dy}{dx}, \frac{dy}{dx}, \frac{Dv}{Dt}, \frac{\delta L}{\delta t}, \frac{\delta L}{\delta t}, \frac{\partial f}{\partial x}, \partial f / \partial x, \frac{dy}{dx}$$

`\pd` `\pd[#1]{#2}{#3}`

偏微分 (partial differential) を表すコマンドです。2 階以上の微分を表す場合は、[#1] オプションに階数を入力することで出力できます。例えば、`displaystyle` 中で `\pd[n]{y}{x}` と入力すると $\frac{\partial^ny}{\partial x^n}$ のように出力されます。

`\pdd` `\pdd[#1]{#2}{#3}[#4]{#5}[#6]`

2 変数偏微分を表します。例えば、`displaystyle` 中で `\pdd[n+m]{f}{x}[n]{y}[m]` と入力すると $\frac{\partial^{n+m}f}{\partial x^n \partial y^m}$ のように出力されます。

`\textpd` `\textpd[#1]{#2}{#3}`

テキスト形式の分数表現を用いた微分記号です。例えば、`\textpd[n]{y}{x}` と入力すると $\partial^ny/\partial x^n$ のように出力されます。

1.3 アクチュアリー記号

`\lisyb` `\lisyb[#1](#2)<#3>{#4}[#5](#6){#7}{#8}`

アクチュアリー記号 (life actuarial symbol) を出力する汎用的なコマンドです。
各引数やオプションの詳細は以下の表の通りです。

引数詳細	
引数	詳細
#1	引数として <code>mod</code> を指定すると、保険金即時払い (payable at the moment of the death) を表すためのバーを出力します。また、 <code>*</code> を指定すると、営業保険料を表す記号を出力します。 例: <code>\lisyb[mod]{P}{x}{n}</code> $\rightarrow \bar{P}$ 、 <code>\lisyb[*]{P}{x}{n}</code> $\rightarrow P^*$ これらのオプションをカンマで区切って同時に指定することもできます。 例: <code>\lisyb[mod,*]{P}{x}{n}</code> $\rightarrow \bar{P}^*$
#2	引数を左下添え字として出力します。例: <code>\lisyb(t){V}{x}{n}</code> $\rightarrow {}_tV$
#3	引数を左上添え字として出力します。例: <code>\lisyb<m>{V}{x}{n}</code> $\rightarrow {}^mV$
#4	保険種類などを表すための文字を指定します。例: <code>\lisyb{A}{x}{n}</code> $\rightarrow A$
#5	引数を第 1 右上添え字として出力し、ブラケット[] で囲みます。 例: <code>\lisyb{V}[A]{x}{n}</code> $\rightarrow V^{[A]}$
#6	引数を第 2 右上添え字として出力し、パーレーン() で囲みます。 例: <code>\lisyb{P}(k){x}{n}</code> $\rightarrow P^{(k)}$
#7	引数を第 1 右下添え字として出力します。通常、保険加入時の年齢を表します。 null 値でも構いません。 例: <code>\lisyb{A}{x}{n}</code> $\rightarrow A_x$
#8	引数を第 2 右下添え字として出力します。その際、保険期間を表す記号 $\overline{}$ も出力されます。 null 値でも構いません。その場合、記号も含め引数#8 は出力されません。 例: <code>\lisyb{a}{x}{n}</code> $\rightarrow a_{\overline{n}}$

出力例			
出力	入力	出力	入力
$a_{x:\overline{n}}$	<code>\lisyb{a}{x}{n}</code>	$a_{\overline{n}}$	<code>\lisyb{a}{}{n}</code>
a_x	<code>\lisyb{a}{x}{}{}</code>	$A_{x:\overline{n}}^{(k)}$	<code>\lisyb{A}(k){x}{n}</code>
$s_{\overline{n+f}}^{(k)}$	<code>\lisyb{s}(k){x}{n+f}</code>	$P_{x:\overline{n}}^{(\infty)}$	<code>\lisyb{P}(\infty){x}{n}</code>
$\bar{P}_{x:\overline{n}}$	<code>\lisyb[mod]{P}{x}{n}</code>	$\bar{P}_{x:\overline{n}}^*$	<code>\lisyb[mod,*]{P}{x}{n}</code>
${}_mP_{x:\overline{n}}$	<code>\lisyb(m){P}{x}{n}</code>	${}_t^mV_{x:\overline{n}}$	<code>\lisyb(t)<m>{V}{x}{n}</code>
${}_tV_{x:\overline{n}}^{[hz]}$	<code>\lisyb(t){V}[hz]{x}{n}</code>	${}_tV_{x:\overline{n}}^{[hz](k)}$	<code>\lisyb(t){V}[hz](k){x}{n}</code>
${}_t^m\bar{V}_{x:\overline{n}}^{[hz](k)*}$	<code>\lisyb[mod,*](t)<m>{V}[hz](k){x}{n}</code>		

`\defer` `\defer{#1}{#2}`

据え置き (deferred) を表す記号を出力します。例えば、`\defer{f}{\lissymb{A}{x}{n}}` と書けば、 ${}_f|A_{x:\overline{n}}$ と出力されます。また、第 2 引数を用いれば、`\defer{f}{n}q_x` と書くことで、 ${}_f|_nq_x$ と出力されます。

`\annuity` `\annuity[#1](#2){#3}{#4}{#5}{#6}`

このコマンドは確定年金 (annuity certain) または生命年金 (life annuity) の現価と終価を表す記号を出力します。この使い方は `\lissymb` とほとんど同じです。なお、期末払い (ordinary) を表したい場合は [#1] オプションに `o` を指定します。デフォルトであるオプション `d` を指定することで、期始払い (due) を陽に示すこともできます。連続払いを表すためには、[#1] オプションに `mod` を、または (#4) オプションに `inf` (infty, infinite) を指定してください。

出力例

出力	入力	出力	入力
$\ddot{s}_{\overline{n} }$	<code>\annuity{s}{x}{n}</code>	\ddot{a}_x	<code>\annuity{a}{x}{x}</code>
$a_{x:\overline{n} }$	<code>\annuity{o}{a}{x}{n}</code>	$\ddot{a}_{x:\overline{n} }$	<code>\annuity{d}{a}{x}{n}</code>
$a_{x:\overline{n} }^{(k)}$	<code>\annuity{o}{a}{(k)}{x}{n}</code>	${}_f \ddot{s}_{x:\overline{n} }$	<code>\annuity(f){s}{x}{n}</code>
$\ddot{a}_{x:\overline{n} }^{(\infty)}$	<code>\annuity{a}{\infty}{x}{n}</code>	$\bar{a}_{x:\overline{n} }$	<code>\annuity{a}{inf}{x}{n}</code>
${}_f \bar{a}_{x:\overline{n} }$	<code>\annuity[mod](f){a}{inf}{x}{n}</code>	$\bar{a}_{x:\overline{n} }$	<code>\annuity[mod]{a}{x}{n}</code>
$\ddot{a}_{n-\frac{1}{k}}^{(k)}$	<code>\annuity{a}{(k)}{n-\frac{1}{k}}</code>		
$a_{n-\frac{1}{k}}^{(k)}$	<code>a_{\step{n-\frac{1}{k}}}{\frac{1}{k}}^{\frac{1}{k}}\frac{1}{k}</code>		

`\pureendow` `\pureendow{#1}(#2){#3}{#4}`

純粋生存保険 (pure endowment) を表す記号を出力します。通常、第 3、4 引数が `null` の純粋生存保険は考えられませんが、このコマンドではそのような場合でも出力されます。(例参照) また、分割払いの場合、例のように $\frac{1}{n}$ よりも後ろにそれを表す記号が出力されてしまいます。この点は好き嫌いがあると思いますので、今後改良していきたいと考えています。

出力例

出力	入力	出力	入力
$A_{x:\overline{n} }^{\frac{1}{n}}$	<code>\pureendow{A}{x}{n}</code>	$P_{x:\overline{n} }^{\frac{1}{n}(k)}$	<code>\pureendow{P}{(k)}{x}{n}</code>
$A_{\overline{n} }^{\frac{1}{n}}$	<code>\pureendow{A}{x}{n}</code>	$P_x^{\frac{1}{n}}$	<code>\pureendow{P}{x}{x}</code>

`\termins` `\termins[#1]{#2}(#3)[#4]{#5}{#6}`

定期保険 (term insurance) を表す記号を出力します。引数 #6 を省略した場合、終身保険の記号を出力します。

出力例

出力	入力	出力	入力
$P_{x:\overline{n}}^1$	<code>\termins{P}{x}{n}</code>	$P_{x:\overline{n}}^{1(k)}$	<code>\termins{P}(k){x}{n}</code>
$\bar{P}_{x:\overline{n}}^1$	<code>\termins[mod]{P}{x}{n}</code>	$P_{x:\overline{n}}^{1*}$	<code>\termins{P}[*]{x}{n}</code>
$\bar{P}_{x:\overline{n}}^{1(\infty)}$	<code>\termins[mod]{P}(\infty){x}{n}</code>	$P_{x:\overline{n}}^{1(k)*}$	<code>\termins{P}(k)[*]{x}{n}</code>
$P_{\overline{n}}^1$	<code>\termins{P}{}{n}</code>	P_x	<code>\termins{P}{x}{}{}</code>
$P_x^{(k)}$	<code>\termins{P}(k){x}{}{}</code>	P_x^*	<code>\termins{P}[*]{x}{}{}</code>

`\endow` `\endow[#1](#2){#3}(#4)[#5]{#6}{#7}`

養老保険 (endowment) を表す記号を出力します。引数#6 を省略した場合、元金償還保険の記号を出力します。引数#7 を省略した場合、終身保険の記号を出力します。

出力例

出力	入力	出力	入力
$P_{x:\overline{n}}$	<code>\endow{P}{x}{n}</code>	$P_{x:\overline{n}}^{(k)}$	<code>\endow{P}(k){x}{n}</code>
$\bar{P}_{x:\overline{n}}$	<code>\endow[mod]{P}{x}{n}</code>	$P_{x:\overline{n}}^*$	<code>\endow{P}[*]{x}{n}</code>
$\bar{V}_{x:\overline{n}}^{(k)}$	<code>\endow[mod]{V}(k){x}{n}</code>	$\bar{P}_{x:\overline{n}}^{(k)*}$	<code>\endow[mod]{P}(k)[*]{x}{n}</code>
P_x	<code>\endow{P}{x}{}{}</code>	$P_{\overline{n}}$	<code>\endow{P}{}{n}</code>

`\joint` `\joint[#1]{#2}(#3)[#4]`

連合生命 (joint lives) において、加入者集合を表す記号を出力します。[#1] オプションへの - 指定は単生命などを表すオーバーラインを出力し、^を指定した場合は共存を意味する下向き亀甲括弧 (`\overtortoise{#1}`) を出力します。

このコマンドによる出力は、デフォルトでは高さや深さが0になっています。この設定を解除するためには、[#1] オプションで `ns` (not smash) を指定してください。例えば、 $q_{\overline{xyz}}^A$ (`q_{\joint[-]{xyz}[r]}^A`) は、`ns` 指定することで $q_{\overline{xyz}}^A$ (`q_{\joint[ns,-]{xyz}[r]}^A`) になります。

出力例

出力	入力	出力	入力
p_{xy}	<code>p_{\joint{xy}}</code>	$p_{xyz\cdots(m)}$	<code>p_{\joint{xyz}(m)}</code>
$p_{\overline{xy\cdots(m)}}$	<code>p_{\joint[-]{xy}(m)}</code>	$p_{\overline{xy\cdots(m)}}^r$	<code>p_{\joint[-]{xy}(m)[r]}</code>
$A_{\overline{xy}}$	<code>A_{\joint[^]{xy}}</code>	$A_{\overline{xy} z:\overline{n}}^1$	<code>\endow{A}{\joint[^]{xy}[1] z}{n}</code>
$p_{xy\cdots(m)}^{[r]}$	<code>p_{\joint{xy}(m)[[r]]}</code>	$p_{\overline{xy\cdots(m)}}^r$	<code>p_{\joint[^]{xy}(m)[r]}</code>
$p_{\overline{xy\cdots(m)}}^r$	<code>p_{\joint[ns,-]{xy}(m)[r]}</code>	$p_{\overline{xy\cdots(m)}}^r$	<code>p_{\joint[ns,^]{xy}(m)[r]}</code>

`\aalign` `\aalign[#1]{#2}{#3}`

このコマンドは、引数#2の上 (above) に引数#3を出力します。

このコマンドを使用すると、 q_x ($q_{\{x\}}$) と比べ添え字の高さが大きくなるため、 x などの文字は下に付き過ぎな印象を受けます。これを避けたい場合は、[#1] オプションで `s` を指定してください。`\smash` が実行され、高さや深さが 0 になります。

例: $q_x \rightarrow q_x^1$ ($q_{\{x\}} \rightarrow q_{\{x\}}^1$)

ついでに、既に定義されているコマンドと比較してみます。どれも同じような出力

出力例		
<code>\mathop{#1}</code>	$\frac{1}{x}$	<code>\mathop{x}{\limits^1}</code>
<code>\stackrel{#1}{#2}</code>	$\frac{1}{x}$	<code>\stackrel{1}{x}</code>
<code>\atop</code>	$\frac{x}{1}$	<code>\atop 1</code>
<code>\oalign{#1}\crlr{#2}</code>	$\frac{1}{x}$	<code>\oalign{#1}\crlr{x}</code>
<code>\aalign{#1}{#2}</code>	$\frac{1}{x}$	<code>\aalign{x}{1}</code>

ですが、`\mathop` は (`\sum` などの) 演算子を表すためのものであり、連生を表すには正しい使い方ではありません。同様に、数学での関係記号を出力する `\stackrel` も連生においては正しいとは言えません。`\atop` は線の無い分数のような出力をします。したがって、ベースライン上に乗らないことが欠点と言えます。`\oalign` は上付き数字を出力するには正確な出力を得ません。このように、デフォルトコマンドでは正しい使い方でも連生を上手に表現することができないため、`\aalign` を定義しました。

添字の位置を調整するには、パラメータ `\alignsep` の値を再定義することにより行えます。デフォルトでは 1.3 に設定され、大きい値に再定義するとより離れて出力されます:

p_x^1 `\renewcommand{\alignsep}{2}` p_x^1

`\balign` `\balign[#1]{#2}{#3}`

このコマンドは、#2の下 (below) に#3を出力します。

このコマンドを使用すると、 q_x ($q_{\{x\}}$) と比べ添え字の高さが大きくなるため、全体として下に付き過ぎな印象を受けます。これを避けたい場合は、[#1] オプションで `s` を指定してください。`\smash` が実行され、高さや深さが 0 になります。例: q_x

$\rightarrow q_x$ (`\fbox{$q_{\{x\}}$}` \rightarrow `\fbox{$q_{\{x\}}$}`)

なお、連生の条件 (保険事故の順序) を表す記号を出力するこれらのコマンドは、連続して入力すると冗長となる欠点があります。そこで、これらの入力を簡略化したコマンド `\order` を v3 にて公開しました。したがって、v3 公開までの一時的な代替策として用意した省略コマンド `\aa`, `\ba` の必要性がなくなりましたので、廃止しました。

`\order` `\order{x_1, x_2, \dots}{n_1, n_2, \dots}`

このコマンドは、被保険者列 $(x_i)_{i=1,2,\dots}$ のそれぞれの元に対し、保険事故の順番を表す列 $(n_i)_{i=1,2,\dots}$ を対応付けて出力するコマンドです。期間内の保険事故の場

出力例

出力	入力
$q_{x_2 y}$	<code>q_{\aalign{x}{1}\balign{y}{2}}</code>
$q_{x_1^{2:3} y z}$	<code>q_{\balign{x}{1}\aalign{y}{2:3}z}</code>
$q_{x_1^{2:3} y z}$	<code>q_{\balign{x}{1}\aalign[s]{y}{2:3}z}</code>
$a_{\overline{xy} z_1}$	<code>a_{\aalign[s]{\overline{xy}}{2} \balign{z}{1}}</code>
$a_{\overline{xy} z_1}$	<code>a_{\aalign[s]{\joint[-]{xy}}{2} \balign{z}{1}}</code>
$a_{\overline{xy} z_1}$	<code>a_{\{\joint[-]{xy}[2] \balign{z}{1}}</code>
$a_{\overline{xy} z_1}$	<code>a_{\aalign[s]{\overbracket{xy}}{2} \balign{z}{1}}</code>
$a_{\overline{xy} z_1}$	<code>a_{\aalign[s]{\joint[^]{xy}}{2} \balign{z}{1}}</code>
$a_{\overline{xy} z_1}$	<code>a_{\joint[^]{xy}[2] \balign{z}{1}}</code>
$q_{xy z \overline{w}_1}$	<code>q_{xy\balign{\joint[-]{zw}}{1}}</code>
$A_{x y z : \overline{n}}$	<code>\lisyb{A}{\aalign{x}{1}y z}{n}</code>
$M_{x+n, y+n}$	<code>M_{\aalign{x+n}{1}, y+n}</code>
$M_{x+n, y+n}$	<code>M_{\aalign[s]{x+n}{1}, y+n}</code>
$M_{x+n, y+n}$	<code>M_{\balign{x+n}{1}, y+n}</code>
$M_{x+n, y+n}$	<code>M_{\balign[s]{x+n}{1}, y+n}</code>

合、その番号 n を上付きで表現するため、引数には $\wedge n$ と入力します。期間外の場合は、下付きで表現するため $_n$ と入力します。列の元が複数ある場合は、それらをカンマで区切ります。例えば、 $\overset{2}{x}_1y$ と出力したい場合、 $\backslash\text{order}\{x,y\}\{\wedge 2,_1\}$ と入力します。それぞれの列の元の個数が等しくない場合は、少ない個数の方に制限され、それ以降の元は無視されます。

また、保険者を表す文字の高さや深さが均一でない場合、その順序を表す文字の位置も不均一に出力されます。そのため、高さや深さを均一にするコマンド $\backslash\text{flattenalign}$ を用意しました。デフォルトではこの設定が有効になっていますが、これを解除したい場合は $\backslash\text{breakflat}$ を実行してください。

出力例			
出力	入力	出力	入力
$p_{xyz}^{23}_1$	$p_{\backslash\text{order}\{x,y,z\}\{_1,\wedge 2,\wedge 3\}}$	p_{abcd}^{1234}	$p_{\backslash\text{order}\{a,b,c,d\}\{\wedge 1,\wedge 2,\wedge 3,\wedge 4\}}$
p_{xyz}_{123}	$p_{\backslash\text{order}\{x,y,z\}\{_1,_2,_3\}}$	$p_{xy}^1_{2}$	$p_{\backslash\text{order}\{x,y\}\{\wedge 1,_2,_3\}}$
$p_{xy}^1_{2}$	$p_{\backslash\text{order}\{x,y,z\}\{\wedge 1,_2\}}$		

$\backslash\text{breakflat}$ 後の出力例			
出力	入力	出力	入力
$p_{xyz}^{23}_1$	$p_{\backslash\text{order}\{x,y,z\}\{_1,\wedge 2,\wedge 3\}}$	p_{abcd}^{234}	$p_{\backslash\text{order}\{a,b,c,d\}\{\wedge 1,\wedge 2,\wedge 3,\wedge 4\}}$
p_{xyz}_{123}	$p_{\backslash\text{order}\{x,y,z\}\{_1,_2,_3\}}$	$p_{xy}^1_{2}$	$p_{\backslash\text{order}\{x,y\}\{\wedge 1,_2,_3\}}$
$p_{xy}^1_{2}$	$p_{\backslash\text{order}\{x,y,z\}\{\wedge 1,_2\}}$		

2 コマンドの応用

前節までで定義したコマンドを組み合わせることで、より多くのアクチュアリー記号を出力することができます。

応用例		
対象	出力	入力
累加（逓増）年金	$(Is)_{\overline{n}}$	<code>(Is)_{\step{n}}</code>
	$(Ia)_{\overline{n}}$	<code>\lisymp{(Ia)}_{\{x\}{n}}</code>
	$(I\ddot{a})_{\overline{n}}$	<code>\lisymp{(I\ddot{a})}_{\{x\}{n}}</code>
	$(I\ddot{a})_{\overline{n}}$	<code>(I\ddot{a})_{\step{n}}</code>
累加（逓増）生命年金	$(I_{\overline{m}}a)_{\overline{n}}$	<code>\lisymp{(I_{\step{m}}a)}_{\{x\}{n}}</code>
	$(I\ddot{a})_{x:\overline{n}}$	<code>(I\ddot{a})_{x:\step{n}}</code>
	$(Ia)_{x:\overline{n}}$	<code>\lisymp{(Ia)}_{\{x\}{n}}</code>
	$(I_{\overline{m}}a)_x$	<code>\lisymp{(I_{\step{m}}a)}_{\{x\}{}}</code>
累加（逓増）定期保険	$(IA)^1_{x:\overline{n}}$	<code>\termins{(IA)}_{\{x\}{n}}</code>
	$(I_{\overline{m}}\bar{A})^1_{x:\overline{n}}$	<code>\termins{(I_{\step{m}}\bar{A})}_{\{x\}{n}}</code>
	$(\bar{I}\bar{A})_x$	<code>\termins{(\bar{I}\bar{A})}_{\{x\}{}}</code>
完全年金	$\ddot{a}_{x:\overline{n}}$	<code>\annuity[o]{\mathring{a}}_{\{x\}{n}}</code>
	$\ddot{a}_{x:\overline{n}}$	<code>\lisymp{\mathring{a}}_{\{x\}{n}}</code>
利率の明示	$a_{x:\overline{n}}^{[1.5\%]}$	<code>\lisymp{a}[1.5\%]_{\{x\}{n}}</code>
短期チルメル式責準	${}_tV_{x:\overline{n}}^{[hz]}$	<code>\lisymp{(t){V}[hz]_{\{x\}{n}}</code>
調整純保険料式責準	${}_tV_{x:\overline{n}}^{[I]}$	<code>\lisymp{(t){V}[I]_{\{x\}{n}}</code>
分割払連生年金	$\ddot{a}_{xy:\overline{n}}^{(k)}$	<code>\annuity{a}(k)_{\{xy\}{n}}</code>
最終生存者連生年金	$a_{\overline{xy},\overline{zw}:\overline{n}}$	<code>\annuity[o]{a}{\joint[-]{xy},\joint[-]{zw}}_{\{n}}</code>
分割払最終生存者	$A_{\overline{xy}:\overline{n}}^{(k)}$	<code>\termins{A}(k){\joint[-]{xy}}_{\{n}}</code>
連生定期保険料	$A_{\overline{xy}:\overline{n}}^{(k)}$	<code>\lisymp{A}(k){\joint[ns,-]{xy}[1]}_{\{n}}</code>
	$A_{\overline{xy}:\overline{n}}^{(k)}$	<code>\lisymp{A}{\joint[-]{xy}[1]}_{\{n\}}^{\frac{\text{pay}\{k\}}</code>

次のページに続きます。

		応用例
対象	出力	入力
遺族年金	$a_{xy z:\overline{n}}$	<code>\annuity[o]{a}{xy z}{n}</code>
	$a_{xy z:\overline{n}}$	<code>\annuity[o]{a}{xy \vert z}{n}</code>
即時開始復帰年金	$\hat{a}_{x y:\overline{n}}$	<code>\annuity[o]{\hat{a}}{x y}{n}</code>
	$\widehat{a}_{x y:\overline{n}}$	<code>\annuity[o]{\widehat{a}}{x y}{n}</code>
即時開始完全年金	$\hat{\ddot{a}}_{x y}^{(k)}$	<code>\annuity[o]{\hat{\mathring{a}}}{(k){x y}{}</code>
条件付き連生	$\ddot{a}_{xy}^{3,4:65:7}_{1,2:w:v:\overline{n}}$	<code>\annuity{a}{\order{x,y,z,w,v}{^3,_1,^4:6,^5:7,_2}}{n}</code>
就業年金	$\ddot{a}_{x:\overline{n}}^{aa}$	<code>\annuity{a}{x}{n}^{\{aa\}}</code>
	$\ddot{a}_{x:\overline{n}}^i$	<code>\annuity{a}{x}{n}^{\{i\}}</code>
	$\ddot{a}_{x:\overline{n}}^{a(i:\overline{m})}$	<code>\annuity{a}{x}{n}^{\{a(i:\step{m})\}}</code>
	$\ddot{a}_{x:\overline{n}}^{(k)aa}$	<code>\annuity{a}{(k){x}{n}^{\{aa\}}</code>
	$\ddot{a}_{x:\overline{n}}^{..aa(k)}$	<code>\annuity{a}{x}{n}^{\{aa \fp{k}\}}</code>
保険料払込免除特約	$\ddot{a}_{x+t:n-1-t}^{a(i:\overline{y-x-t})}$	<code>\annuity{a}{x+t}{n-1-t}^{\{a(i:\step{y-x-t})\}}</code>
複雑な遺族年金	$\hat{a}_{x y:\overline{n}}^{(a+I)W}$	<code>\annuity[o]{\hat{a}}{x \vert y}{n}^{\{(a+I)W\}}</code>

3 最後に

このマクロを作成するにあたって、生保数理や記号の参考に文献 [1]、[2] を、パッケージ作成に文献 [3]、[4]、[5] をそれぞれ参照しました。

References

- [1] 二見隆、生命保険数学（上・下）、生命保険文化研究所、1999。
- [2] 山内恒人、生命保険数学の基礎—アクチュアリー数学入門—、東京大学出版会、2009。
- [3] 奥村晴彦、 \LaTeX 2 ε 美文書作成入門、技術評論社、2007。
- [4] 藤田眞作、 \LaTeX 2 ε コマンドブック、ソフトバンクパブリッシング、2004。
- [5] F. Mittelbach, M. Goossens, The \LaTeX companion 2nd edi, Addison-Wesley, 2004.