

# naive-ebnf: L<sup>A</sup>T<sub>E</sub>X Package for EBNF in Plain Text\*

Yegor Bugayenko  
yegor256@gmail.com

2023-05-27, 0.0.6

## 1 Introduction

This package helps render an [Extended Backus-Naur Form](#) using plain text notation:

$\langle \lambda\text{-Expr} \rangle \rightarrow \langle \text{Var} \rangle$ $\quad   \text{"}\lambda\text{" } \langle \text{Var} \rangle \text{"}. \text{" } \langle \text{Expr} \rangle$ $\quad   \text{"}\langle \text{Expr} \rangle \langle \text{Expr} \rangle \text{"}$	<pre>1 \documentclass{minimal} 2 \usepackage{naive-ebnf} 3 \usepackage{mathtools} 4 \begin{document} 5 \begin{ebnf} 6 &lt;\$\lambda\$-Expr&gt; := &lt;Var&gt; \\ 7      "\$\lambda\$" &lt;Var&gt; "." &lt;Expr&gt; \\ 8      "\$\lparen\$" &lt;Expr&gt; &lt;Expr&gt; "\$\rparen\$" 9 \end{ebnf} 10 \end{document}</pre>
---	---

`ebnf` The `ebnf` environment *doesn't* add any formatting to the paragraph, but only replaces the plain text symbols, such as “:=” and “<Var>” with proper L<sup>A</sup>T<sub>E</sub>X commands. The following syntax is understood inside the `ebnf` environment:

- := separates the left-hand side from the right-hand side of the production rule;
- < . . . > denotes a non-terminal (variable);
- " . . . " denotes a terminal symbol;
- ' . . . ' denotes a special non-printable terminal symbol, like 'EOL';
- ( . . . | . . . ) denotes a series of options to choose from;
- [ . . . ] denotes an optional substitution;
- { . . . } denotes a zero or more times repetition;
- || denotes an indented vertical bar at the beginning of the string.

**Attention:** The usage of some symbols is prohibited inside terminals. Instead, the following substitutions are recommended:

---

\*The sources are in GitHub at [yegor256/naive-ebnf](#)

- `\lparen` and `\rparen` instead of “(” and “)” (from the [mathtools](#) package);
- `\langle` and `\rangle` instead of “<” and “>”;
- `\lbrace` and `\rbrace` instead of “{” and “}” (also [mathtools](#));
- `\lbrack` and `\rbrack` instead of “[” and “]” (also [mathtools](#));
- `\vert` instead of “|”.

`width` There is an optional argument of `ebnf` environment, which sets the width of the left-hand side of each rule (the default width is `6em`):

This EBNF has a larger width of the left hand side than usual: $(\text{VeryLongVariable}) \rightarrow (X) \mid (Y)$ $(X) \rightarrow "X" \text{ EOL}$ $(Y) \rightarrow "Y"$	<pre> 4 This EBNF has a larger width of \\  5 the left hand side than usual: \par  6 \begin{ebnf}[1.5in]  7 &lt;VeryLongVariable&gt; := &lt;X&gt;   &lt;Y&gt; \\  8 &lt;X&gt; := "X" 'EOL' \\  9 &lt;Y&gt; := "Y"  10 \end{ebnf} </pre>
--	---

`\terminal` Inside the text, terminals, non-terminals, and special terminals may be formatted  
`\nonterminal` using three supplementary commands:  
`\sterminal`

The non-terminal $(\text{Var})$ in $\lambda$ -calculus may be equal to $v_1, v_2, \dots$ . Application starts with “(” and ends with “)”.	<pre> 6 The non-terminal \nonterminal{Var}  7 in \$\lambda\$-calculus may be equal  8 to \$v_1, v_2, \dots\$. Application  9 starts with \terminal{()} and ends  10 with \terminal{)}. </pre>
---	---

It’s possible to use them in math-mode too, for example:

If “ $(f_1(\lambda\text{-Var}))$ ” is always true, then $f_1$ is a tautology.	<pre> 6 If \$\terminal{()} f_1  7 \nonterminal{\lambda-Var}  8 \terminal{)}\$ is always true, then  9 \$f_1\$ is a tautology. </pre>
---	--

## 2 Package Options

It’s possible to configure the behavior of the package with the help of a few package options:

`bw` By default, some colors are used in the rendered grammar. However, the `bw` package option disables any colors and makes sure the gammar is black-and-white:

```
\usepackage[bw]{naive-ebnf}
```

`trail` The `ebnf` environment is doing pre-processing of the  $\TeX$  commands provided and then let  $\LaTeX$  render them. It may be useful to see the output generated by the pre-processing. The `trail` option (with a file name) asks the package to save the content of the environment after the pre-processing into the file:

```
\usepackage[trail=log.tex]{naive-ebnf}
```

### 3 Implementation

First, we process package options:

```
1 \RequirePackage{pgfopts}
2 \pgfkeys{
3   /ebnf/.cd,
4   bw/.store in=\ebnf@bw,
5   trail/.store in=\ebnf@trail,
6   trail/.default=naive-ebnf.tmp.tex,
7   trail
8 }
9 \ProcessPgfPackageOptions{/ebnf}
```

Then, we include a few packages, mostly to deal with L<sup>A</sup>T<sub>E</sub>X3 expressions:

```
10 \RequirePackage{expl3}
```

`\ebnf@color` Then, we include `xcolor` to colorize the output a bit:

```
11 \makeatletter\ifdefined\ebnf@bw\else
12   \RequirePackage{xcolor}
13 \fi
14 \newcommand\ebnf@color[2]
15   {\ifdefined\ebnf@bw#2\else\textcolor{#1}{#2}\fi}
16 \makeatother
```

`\terminal` Then, we a command to render a single terminal:

```
17 \makeatletter
18 \newcommand\terminal[1]{%
19   \relax\ifmmode\else\ttfamily\fi%
20   \ebnf@color{gray}{\relax\ifmmode\textsf{''}\else{\sffamily''}\fi}%
21   #1%
22   \ebnf@color{gray}{\relax\ifmmode\textsf{''}\else{\sffamily''}\fi}}
23 \makeatother
```

`\nonterminal` Then, we a command to render a single non-terminal:

```
24 \makeatletter
25 \newcommand\nonterminal[1]{%
26   \ebnf@color{gray}{\relax\ifmmode\langle\else{\langle}\fi}%
27   \relax\ifmmode\textsf{#1}\else{\sffamily#1}\fi%
28   \ebnf@color{gray}{\relax\ifmmode\rangle\else{\rangle}\fi}}
29 \makeatother
```

`\sterminal` Then, we a command to render a single non-terminal:

```
30 \makeatletter
31 \newcommand\sterminal[1]{\relax\ifmmode\else\ttfamily\fi#1}%
32 \makeatother
```

Then, we define supplementary commands:

```
33 \makeatletter
34 \newcommand\ebnf@optional[1]
35   {\ebnf@color{gray}{[ ]#1\ebnf@color{gray}{}}}
36 \newcommand\ebnf@repetition[1]
37   {\ebnf@color{gray}{\{ }#1\ebnf@color{gray}{\}}}
38 \newcommand\ebnf@grouping[1]
```

```

39  {\ebnf@color{gray}{\(#1\ebnf@color{gray}{})}}
40 \ExplSyntaxOn
41 \newcommand\ebnf@terminal[1]{
42   \tl_set:Nn \l_ebnf_tl { }
43   \tl_set_rescan:Nno \l_ebnf_tl { } { #1 }
44   \terminal{\l_ebnf_tl}
45 }
46 \newcommand\ebnf@sterminal[1]{
47   \tl_set:Nn \l_ebnf_tl { }
48   \tl_set_rescan:Nno \l_ebnf_tl { } { #1 }
49   \sterminal{\l_ebnf_tl}
50 }
51 \newcommand\ebnf@nonterminal[1]{
52   \tl_set:Nn \l_ebnf_tl { }
53   \tl_set_rescan:Nno \l_ebnf_tl { } { #1 }
54   \nonterminal{\l_ebnf_tl}
55 }
56 \ExplSyntaxOff
57 \newcommand\ebnf@to
58  {\ebnf@color{gray}{\(\to)}}
59 \newcommand\ebnf@alternation
60  {\ebnf@color{gray}{\(\vert)}}
61 \makeatother

```

ebnf Then, we define the ebnf environment:

```

62 \ExplSyntaxOn
63 \cs_generate_variant:Nn \tl_replace_all:Nnn {Nx}
64 \makeatletter
65 \NewDocumentEnvironment{ebnf}{0{4em}+b}
66  {\tl_set:Nn \l__ebnf_tmp_tl{#2}}
67  {%
68   \regex_replace_all:nnN { \{(.\+?)\} }%
69   {\c{ebnf@repetition}{\1} \l__ebnf_tmp_tl%
70   \regex_replace_all:nnN { \((.\+?)\} }%
71   {\c{ebnf@grouping}{\1} \l__ebnf_tmp_tl%
72   \regex_replace_all:nnN { \[(.\+?)\} }%
73   {\c{ebnf@optional}{\1} \l__ebnf_tmp_tl%
74   \regex_replace_all:nnN { (<[^\>]+\>\ :=) }%
75   {\c{makebox}[#1][r]{\1} \l__ebnf_tmp_tl%
76   \regex_replace_all:nnN { <(.+?)> }%
77   {\c{ebnf@nonterminal}{\1} \l__ebnf_tmp_tl%
78   \regex_replace_all:nnN { "(.+?)" }%
79   {\c{ebnf@terminal}{\1} \l__ebnf_tmp_tl%
80   \regex_replace_all:nnN { '(.+?)' }%
81   {\c{ebnf@sterminal}{\1} \l__ebnf_tmp_tl%
82   \regex_replace_all:nnN { \(\|) }%
83   {\c{makebox}[#1][r]{ \1 } \l__ebnf_tmp_tl%
84   \regex_replace_all:nnN { \| }%
85   {\c{ebnf@alternation}{}} \l__ebnf_tmp_tl%
86   \regex_replace_all:nnN { := }%
87   {\c{ebnf@to}{}} \l__ebnf_tmp_tl%
88   \tl_put_left:Nn \l__ebnf_tmp_tl {\noindent}
89   \tl_put_right:Nn \l__ebnf_tmp_tl {}
90   \ifdefined\ebnf@trail%

```

```
91 \newwrite\ebnf@write%
92 \immediate\openout\ebnf@write\ebnf@trail\relax%
93 \immediate\write\ebnf@write{\unexpanded\expandafter{\l__ebnf_tmp_tl}}%
94 \immediate\closeout\ebnf@write%
95 \message{naive-ebnf:\space pre-processed\space TeX
96 \space saved\space to\space "\ebnf@trail"^^J}%
97 \fi%
98 \l__ebnf_tmp_tl}
99 \makeatother
100 \ExplSyntaxOff

101 \endinput
```

## Change History

0.0.1	General: First draft. . . . .	3	0.0.4	<code>ebnf</code> : Any symbols are allowed inside <code>\nonterminal</code> commands and inside the <code>ebnf</code> environment, where non-terminals are mentioned. . . . .	4
0.0.2	General: Proper parsing of grouping. . . . .	3			
	Substitutions suggested for special symbols. . . . .	3			
	<code>\nonterminal</code> : New command <code>\nonterminal</code> added, to enable rendering non-terminal symbols outside of the <code>ebnf</code> environment. . . . .	3	0.0.5	General: New package option <code>trail</code> added, to enable saving of the generated $\TeX$ content to a file, for debugging purposes. . . . .	3
	<code>\terminal</code> : New command <code>\terminal</code> added, to enable rendering terminal symbols outside of the <code>ebnf</code> environment. . . . .	3	0.0.6	<code>\sterminal</code> : New command <code>\sterminal</code> added, to enable rendering of special non-printable terminal symbols outside of the <code>ebnf</code> environment. . . . .	3
0.0.3	<code>\terminal</code> : Quotes fixed in both text and math modes. . . . .	3			

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

	<b>Symbols</b>	<code>\ExplSyntaxOn</code> . . . . . 40, 62	<b>P</b>
<code>\(</code> . . . . .	26, 28, 58, 60, 70		<code>\pgfkeys</code> . . . . . 2
<code>\)</code> . . . . .	26, 28, 58, 60, 70	<b>I</b>	<code>\ProcessPgfPackageOptions</code> . . . . . 9
<code>\[</code> . . . . .	72	<code>\ifdefined</code> . . . . . 11, 15, 90	
<code>\{</code> . . . . .	37, 68	<code>\ifmode</code> . . . . . 19, 20, 22, 26, 27, 28, 31	<b>R</b>
<code>\}</code> . . . . .	37, 68	<code>\immediate</code> . . . . . 92, 93, 94	<code>\rangle</code> . . . . . 28
<code>\_</code> . . . . .	74		<code>\regex</code> . . . . . 68, 70, 72, 74, 76, 78, 80, 82, 84, 86
<code>\]</code> . . . . .	72	<b>L</b>	<code>\relax</code> . . . . . 19, 20, 22, 26, 27, 28, 31, 92
<code>\ </code> . . . . .	82, 84	<code>\l</code> . . . . . 42, 43, 44, 47, 48, 49, 52, 53, 54, 66, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 88, 89, 93, 98	<code>\RequirePackage</code> 1, 10, 12
	<b>C</b>	<code>\langle</code> . . . . . 26	<b>S</b>
<code>\c</code> . . . . .	69, 71, 73, 75, 77, 79, 81, 83, 85, 87		<code>\sffamily</code> . . . . . 20, 22, 27
<code>\closeout</code> . . . . .	94	<b>M</b>	<code>\space</code> . . . . . 95, 96
<code>\cs</code> . . . . .	63	<code>\makeatletter</code> . . . . . . . . . . 11, 17, 24, 30, 33, 64	<code>\sterminal</code> . . . . . <u>30</u> , 49
	<b>E</b>	<code>\makeatother</code> . . . . . . . . . . 16, 23, 29, 32, 61, 99	<b>T</b>
<code>\ebnf</code> . . . . .	62	<code>\message</code> . . . . . 95	<code>\terminal</code> . . . . . <u>17</u> , 44
<code>\ebnf@alternation</code> . . . . .	59	<b>N</b>	<code>\textcolor</code> . . . . . 15
<code>\ebnf@bw</code> . . . . .	4, 11, 15	<code>\newcommand</code> . . . . . 14, 18, 25, 31, 34, 36, 38, 41, 46, 51, 57, 59	<code>\textsf</code> . . . . . 20, 22, 27
<code>\ebnf@color</code> . . . . .	. . . . . <u>11</u> , 20, 22, 26, 28, 35, 37, 39, 58, 60	<code>\NewDocumentEnvironment</code> . . . . . 65	<code>\tl</code> . . . . . 42, 43, 47, 48, 52, 53, 63, 66, 88, 89
<code>\ebnf@grouping</code> . . . . .	38	<code>\newwrite</code> . . . . . 91	<code>\to</code> . . . . . 58
<code>\ebnf@nonterminal</code> . . . . .	51	<code>\noindent</code> . . . . . 88	<code>\ttfamily</code> . . . . . 19, 31
<code>\ebnf@optional</code> . . . . .	34	<code>\nonterminal</code> . . . . . <u>24</u> , 54	<b>U</b>
<code>\ebnf@repetition</code> . . . . .	36		<code>\unexpanded</code> . . . . . 93
<code>\ebnf@sterminal</code> . . . . .	46	<b>O</b>	<b>V</b>
<code>\ebnf@terminal</code> . . . . .	41	<code>\openout</code> . . . . . 92	<code>\vert</code> . . . . . 60
<code>\ebnf@to</code> . . . . .	57		<b>W</b>
<code>\ebnf@trail</code> . . . . .	5, 90, 92, 96		<code>\write</code> . . . . . 93
<code>\ebnf@write</code> . . . . .	91, 92, 93, 94		
<code>\endinput</code> . . . . .	101		
<code>\expandafter</code> . . . . .	93		
<code>\ExplSyntaxOff</code> . . . . .	56, 100		