

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

Yegor Bugayenko  
yegor256@gmail.com

2023-01-29, 0.0.2

## 1 Introduction

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

$\begin{array}{l} \langle \text{Expr} \rangle \rightarrow \langle \text{Var} \rangle \\   \text{"}\lambda\text{" } \langle \text{Var} \rangle \text{"}. \text{" } \langle \text{Expr} \rangle \\   \text{"}(\text{" } \langle \text{Expr} \rangle \langle \text{Expr} \rangle \text{"})\text{"} \end{array}$	<pre>1 \documentclass{minimal} 2 \usepackage{naive-ebnf} 3 \usepackage{mathtools} 4 \begin{document} 5 \begin{ebnf} 6 &lt;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>
--	---

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 series of options to choose from;
- `[...]` denotes an optional substitution;
- `{...}` denotes a zero or more times repetition;

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

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

---

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

- `\lbrack` and `\rbrack` instead of “[” and ”” (also `mathtools`);
- `\vert` instead of “|”.

`\terminal` Inside the text, terminals and non-terminals may be formatted using two supplementary commands:

The non-terminal `\Var` in  $\lambda$ -calculus may be equal to  $v_1, v_2, \dots$ . Application starts with “(” and ends with “)”.

```

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{)}.

```

## 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 grammar is black-and-white:

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

## 3 Implementation

First, we process package options:

```

1 \RequirePackage{pgfopts}
2 \pgfkeys{
3   /ebnf/.cd,
4   bw/.store in=\ebnf@bw,
5 }
6 \ProcessPgfPackageOptions{/ebnf}

```

Then, we include a few packages, mostly to deal with  $\text{\LaTeX}$  expressions:

```

7 \RequirePackage{filecontentsdef}
8 \RequirePackage{expl3}

```

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

```

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

```

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

```

15 \makeatletter
16 \newcommand\terminal[1]{%
17   \relax\ifmmode\else\ttfamily\fi%
18   \ebnf@color{gray}{"}#1\ebnf@color{gray}{"}}
19 \makeatother

```

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

```

20 \makeatletter
21 \newcommand\nonterminal[1]
22   {\(\ebnf@color{gray}\langle\textsf{#1}\ebnf@color{gray}\rangle\)}
23 \makeatother

```

Then, we define supplementary commands:

```

24 \makeatletter
25 \newcommand\ebnf@optional[1]
26   {\ebnf@color{gray}\{[]\#1\ebnf@color{gray}\}}
27 \newcommand\ebnf@repetition[1]
28   {\ebnf@color{gray}\{\}\#1\ebnf@color{gray}\{\}\}}
29 \newcommand\ebnf@grouping[1]
30   {\ebnf@color{gray}\{\}\#1\ebnf@color{gray}\{\}\}}
31 \ExplSyntaxOn
32 \newcommand\ebnf@terminal[1]{
33   \tl_set:Nn \l_ebnf_tl { }
34   \tl_set_rescan:Nno \l_ebnf_tl { } { #1 }
35   \terminal{\l_ebnf_tl}
36 }
37 \ExplSyntaxOff
38 \newcommand\ebnf@to
39   {\ebnf@color{gray}\(\to\)}
40 \newcommand\ebnf@alternation
41   {\ebnf@color{gray}\(\vert\)}
42 \newcommand\ebnf@eol{\}
43 \makeatother

```

ebnf Then, we define the ebnf environment:

```

44 \ExplSyntaxOn
45 \cs_generate_variant:Nn \tl_replace_all:Nnn {Nx}
46 \NewDocumentEnvironment{ebnf}{}{\filecontentsdefmacro\l__ebnf_tmp_tl}{
47   \endfilecontentsdefmacro
48   \str_set:NV \l__ebnf_tmp_tl \l__ebnf_tmp_tl
49   \str_set:Nx \l__ebnf_tmp_tl {\str_range:Nnn \l__ebnf_tmp_tl {1} {-2}}
50   \regex_replace_all:nnN { \{(.+)\} }
51     {\c{ebnf@repetition}\1} \l__ebnf_tmp_tl
52   \regex_replace_all:nnN { \((.+?)\}
53     {\c{ebnf@grouping}\1} \l__ebnf_tmp_tl
54   \regex_replace_all:nnN { \[([^\]]+?)\] }
55     {\c{ebnf@optional}\1} \l__ebnf_tmp_tl
56   \regex_replace_all:nnN { <([A-Za-z][a-z-]+)> }
57     {\c{nonterminal}\1} \l__ebnf_tmp_tl
58   \regex_replace_all:nnN { "[^"]+" }
59     {\c{ebnf@terminal}\1} \l__ebnf_tmp_tl
60   \regex_replace_all:nnN { \^M\s*\| }
61     {\^M :=} \l__ebnf_tmp_tl
62   \regex_replace_all:nnN { \| }
63     {\c{ebnf@alternation}\} \l__ebnf_tmp_tl
64   \regex_replace_all:nnN { \^M\s*:= }
65     {\^M \c{-}\c{hspace}{3em}\c{ebnf@alternation}\} \l__ebnf_tmp_tl
66   \regex_replace_all:nnN { := }
67     {\c{ebnf@to}\} \l__ebnf_tmp_tl
68   \regex_replace_all:nnN { \^M }

```

```

69      {\c{ebnf@eol}{}} \l__ebnf_tmp_tl
70      \tl_put_left:Nn \l__ebnf_tmp_tl {}
71      \tl_put_right:Nn \l__ebnf_tmp_tl {}
72      \l__ebnf_tmp_tl
73 }
74 \ExplSyntaxOff

75 \endinput

```

# Change History

0.0.1		
General: First draft. . . . .	2	<code>\nonterminal</code> added, to enable rendering non-terminal symbols outside of the <code>ebnf</code> environment. . 3
0.0.2		
General: Proper parsing of grouping. .	2	<code>\terminal</code> : New command
Substitutions suggested for special symbols. . . . .	2	<code>\terminal</code> added, to enable rendering terminal symbols outside of the <code>ebnf</code> environment. . 2
<code>\nonterminal</code> : New command		

# 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>\endinput</code> ..... 75	<code>\nonterminal</code> ..... <u>20</u>
<code>\(</code> ..... 22, 39, 41, 52		<code>\ExplSyntaxOff</code> ... 37, 74	
<code>\)</code> ..... 22, 39, 41, 52		<code>\ExplSyntaxOn</code> ... 31, 44	<b>P</b>
<code>\[</code> ..... 54			<code>\pgfkeys</code> ..... 2
<code>\{</code> ..... 28, 50	<b>F</b>		<code>\ProcessPgfPackageOptions</code>
<code>\}</code> ..... 28, 50	<code>\filecontentsdefmacro</code>		..... 6
<code>\]</code> ..... 54	..... 46		
<code>\^</code> ..... 60, 61, 64, 65, 68	<b>I</b>	<b>R</b>	
<code>\ </code> ..... 60, 62	<code>\ifdefined</code> ..... 9, 13	<code>\rangle</code> ..... 22	
	<code>\ifmmode</code> ..... 17	<code>\regex</code> .. 50, 52, 54, 56,	
<b>C</b>		58, 60, 62, 64, 66, 68	
<code>\c</code> ..... 51, 53, 55,	<b>L</b>	<code>\relax</code> ..... 17	
57, 59, 63, 65, 67, 69	<code>\l</code> ..... 33, 34, 35,	<code>\RequirePackage</code> 1, 7, 8, 10	
<code>\cs</code> ..... 45	46, 48, 49, 51, 53,		
	55, 57, 59, 61, 63,	<b>S</b>	
<b>E</b>	65, 67, 69, 70, 71, 72	<code>\str</code> ..... 48, 49	
<code>\ebnf</code> ..... <u>44</u>	<code>\langle</code> ..... 22		
<code>\ebnf@alternation</code> .. 40	<b>M</b>	<b>T</b>	
<code>\ebnf@bw</code> ..... 4, 9, 13	<code>\makeatletter</code> 9, 15, 20, 24	<code>\terminal</code> ..... <u>15</u> , 35	
<code>\ebnf@color</code> .... <u>9</u> , 18,	<code>\makeatother</code> 14, 19, 23, 43	<code>\textcolor</code> ..... 13	
22, 26, 28, 30, 39, 41		<code>\textsf</code> ..... 22	
<code>\ebnf@eol</code> ..... 42	<b>N</b>	<code>\tl</code> ..... 33, 34, 45, 70, 71	
<code>\ebnf@grouping</code> ..... 29	<code>\newcommand</code> .....	<code>\to</code> ..... 39	
<code>\ebnf@optional</code> ..... 25	... 12, 16, 21, 25,	<code>\ttfamily</code> ..... 17	
<code>\ebnf@repetition</code> ... 27	27, 29, 32, 38, 40, 42		
<code>\ebnf@terminal</code> ..... 32	<code>\NewDocumentEnvironment</code>	<b>V</b>	
<code>\ebnf@to</code> ..... 38	..... 46	<code>\vert</code> ..... 41	
<code>\endfilecontentsdefmacro</code>			
..... 47			