`naive-ebnf`: LaTeX 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:

⟨λ-Expr⟩ → ⟨Var⟩
      | "λ" ⟨Var⟩ "." ⟨Expr⟩
      | "(" ⟨Expr⟩ ⟨Expr⟩ ")"

```
1  \documentclass{minimal}
2  \usepackage{naive-ebnf}
3  \usepackage{mathtools}
4  \begin{document}
5  \begin{ebnf}
6  <$\lambda$-Expr> := <Var> \\
7    || "$\lambda$" <Var> "." <Expr> \\
8    || "$\lparen$" <Expr> <Expr> "$\rparen$"
9  \end{ebnf}
10 \end{document}
```

`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 LaTeX 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:
> $\langle \mathsf{VeryLongVariable} \rangle \rightarrow \langle \mathsf{X} \rangle \mid \langle \mathsf{Y} \rangle$
> $\langle \mathsf{X} \rangle \rightarrow \text{"X"}\ \textsc{EOL}$
> $\langle \mathsf{Y} \rangle \rightarrow \text{"Y"}$

```
4  This EBNF has a larger width of \\
5  the left hand side than usual: \par
6  \begin{ebnf}[1.5in]
7  <VeryLongVariable> := <X> | <Y> \\
8  <X> := "X" 'EOL' \\
9  <Y> := "Y"
10 \end{ebnf}
```

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

> The non-terminal $\langle \mathsf{Var} \rangle$ in λ-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{)}.
```

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

> If "(" $f_1 \langle \lambda\text{-}\mathsf{Var} \rangle$ ")" is always true, then f_1 is a tautology.

```
6  If $\terminal{(} f_1
7  \nonterminal{$\lambda$-Var}
8  \terminal{)}$ is always true, then
9  $f_1$ is a tautology.
```

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 LaTeX3 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}{O{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

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.