

simplebnf — A simple package to format Backus-Naur form*

Jay Lee[†]

2022/05/07

This package provides a simple way to typeset grammars written in Backus-Naur form (BNF).

```
\bnfexpr \bnfannot
```

These commands are wrappers around `\texttt` and `\textit` respectively.

```
\begin{bnfgrammar} text\end{bnfgrammar}
```

can be used to typeset BNF grammars. The *text* inside the environment should be formatted as:

```
term1 ::= rhs1
;;
term2 ::= rhs2
;;
...
termk ::= rhsk
```

where each of the *rhs* represents alternative syntactic forms of the *term*. An annotation may accompany each alternative in which case the alternative must be separated from its annotation with a colon (:). If you don't need annotations, simply omit the colons and annotations altogether. The alternatives themselves are separated using the pipe symbol (`|`).

A sample code and the result is shown below:

<pre>\begin{bnfgrammar} a \in \textit{Vars} ;; expr ::= expr + term : sum term : term ;; term ::= term * a : product a : variable \end{bnfgrammar}</pre>	<pre> a ∈ Vars expr ::= expr + term sum term term term ::= term * a product a variable</pre>
--	---

*This file describes v0.3.0.

[†]E-mail: jaeho.lee@snu.ac.kr

Annotations can also be provided on left-hand sides, to label the nonterminal instead of a specific production.

<pre>\begin{bnfgrammar} a : Variables \in \textit{Vars} ;; expr : Expressions ::= expr + term term ;; term ::= term * a a \end{bnfgrammar}</pre>	<table style="border: none; margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;"><i>Variables</i></td> <td style="padding-right: 10px;">a</td> <td style="padding-right: 10px;">∈</td> <td><i>Vars</i></td> </tr> <tr> <td style="padding-right: 10px;"><i>Expressions</i></td> <td style="padding-right: 10px;">expr</td> <td style="padding-right: 10px;">::=</td> <td>expr + term term</td> </tr> <tr> <td></td> <td style="padding-right: 10px;">term</td> <td style="padding-right: 10px;">::=</td> <td>term * a a</td> </tr> </table>	<i>Variables</i>	a	∈	<i>Vars</i>	<i>Expressions</i>	expr	::=	expr + term term		term	::=	term * a a
<i>Variables</i>	a	∈	<i>Vars</i>										
<i>Expressions</i>	expr	::=	expr + term term										
	term	::=	term * a a										

You can also provide an optional specification to the grammar environment, to redefine alignment or spacing.

<i>Variables</i>	a	∈	<i>Vars</i>
	expr	::=	expr + term <i>sum</i> term <i>term</i>
	term	::=	term * a <i>product</i> a <i>variable</i>

```
\begin{bnfgrammar}[lr@{\hspace{4pt}}c@{\hspace{2pt}}ll]
a : Variables \in \textit{Vars}
;;
expr ::=
  expr + term : sum
| term        : term
;;
term ::=
  term * a : product
| a        : variable
\end{bnfgrammar}
```

If you want to typeset multiple productions on a single line, you can use double vertical bars by default.

<pre>\begin{bnfgrammar} a \in \textit{Vars} ;; expr ::= expr + term term ;; term ::= term * a a \end{bnfgrammar}</pre>	<table style="border: none; margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;">a</td> <td style="padding-right: 10px;">∈</td> <td><i>Vars</i></td> </tr> <tr> <td style="padding-right: 10px;">expr</td> <td style="padding-right: 10px;">::=</td> <td>expr + term term</td> </tr> <tr> <td style="padding-right: 10px;">term</td> <td style="padding-right: 10px;">::=</td> <td>term * a a</td> </tr> </table>	a	∈	<i>Vars</i>	expr	::=	expr + term term	term	::=	term * a a
a	∈	<i>Vars</i>								
expr	::=	expr + term term								
term	::=	term * a a								

The second and third optional arguments specify regular expressions for the line-breaking and non-breaking RHS separators:

$$\begin{aligned} a &\in Vars \\ \text{expr} &::= \text{expr} + \text{term} \mid \text{term} \\ \text{term} &::= \text{term} * a \\ &\mid a \end{aligned}$$

```
\begin{bnfgrammar}[llc11][\|\|][\|]
a \in \textit{Vars}
;;
expr ::= expr + term | term
;;
term ::= term * a
|| a
\end{bnfgrammar}
```