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:

```
\begin{bnfgrammar}
  a \in \textit{Vars}
  ;;
  expr ::=
    expr + term : sum
  | term        : term
  ;;
  term ::=
    term * a : product
  | a        :
  variable
\end{bnfgrammar}
```

a ∈ *Vars*

expr	::=	expr + term	*sum*
	|	term	*term*
term	::=	term * a	*product*
	|	a	*variable*

*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.

```
\begin{bnfgrammar}
  a : Variables \in
  \textit{Vars}
  ;;
  expr : Expressions
  ::=
    expr + term
  | term
  ;;
  term ::=
    term * a
  | a
\end{bnfgrammar}
```

Variables a ∈ *Vars*
Expressions expr ::= expr + term
| term
term ::= term * a
| a

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

Variables a ∈ *Vars*

expr ::=	expr + term	*sum*
|	term	*term*
term ::=	term * a	*product*
|	a	*variable*

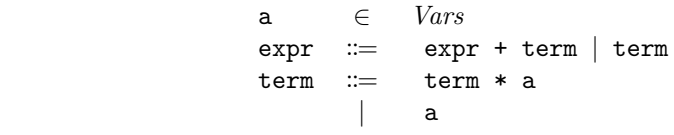
```
\begin{bnfgrammar}[lr@{\hspace{4pt}}c@{\hspace{2pt}}ll]
  a : Variables \in \textit{Vars}
  ;;
  expr ::=
    expr + term : sum
  | term        : term
  ;;
  term ::=
    term * a : product
  | a        : variable
\end{bnfgrammar}
```

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

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

a ∈ *Vars*
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 seperators:




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