

# simplebnf – A simple package to format Backus-Naur form<sup>\*</sup>

Jay Lee<sup>†</sup>

2019/12/23

```
\bnfexpr \bnfannot
```

The `\bnfexpr` and the `\bnfannot` commands are simply wrappers around `\textttt` and `\textit`, respectively.

```
\begin{bnfgrammar}{\langle term \rangle} text \end{bnfgrammar}
```

The *term* argument of the `bnfgrammar` environment is the term to define a grammar. The text inside the environment should be a comma-separated list of keypairs. Each keypair represents an alternative syntactic form of the *term* and its annotation, delimited with a colon(:).

A sample code and the result is shown below:

```
\begin{bnfgrammar}
\bnfexpr{v} ::= 
  n:           integer
  | $lambda$x.e: abstraction
\end{bnfgrammar}
```

```
\begin{bnfgrammar}
\bnfexpr{C} ::= 
  \bnfexpr{[]}:   hole
  | \bnfexpr{C\,e}: application 1
  | \bnfexpr{v\,C}: application 2
  | \bnfexpr{C\,+\\,e}: addition 1
  | \bnfexpr{v\,+\\,C}: addition 2
\end{bnfgrammar}
```

$v ::= n \quad \text{integer}$ $  \lambda x. e \quad \text{abstraction}$	$v ::= n \quad \text{integer}$ $  \lambda x. e \quad \text{abstraction}$
$C ::= [] \quad \text{hole}$ $  C e \quad \text{application 1}$ $  v C \quad \text{application 2}$ $  C + e \quad \text{addition 1}$ $  v + C \quad \text{addition 2}$	$C ::= [] \quad \text{hole}$ $  C e \quad \text{application 1}$ $  v C \quad \text{application 2}$ $  C + e \quad \text{addition 1}$ $  v + C \quad \text{addition 2}$

---

<sup>\*</sup>This file describes v0.1.0.

<sup>†</sup>E-mail: jaeho.lee@snu.ac.kr