The \texttt{backnaur} package∗

Adrian P. Robson†

12 December 2012

1 Introduction

The \texttt{backnaur} package typesets Backus-Naur Form (BNF) definitions. It creates aligned lists of productions, with numbers if required. It can also print in line BNF expressions using math mode.

Backus-Naur Form is a notation for defining context free grammars. It is used to describe such things as programming languages, communication protocols and command syntaxes, but it can be useful whenever a rigorous definition of language is needed.

2 BNF Definitions

The following is a BNF definition of a semicolon separated list:

\begin{bnf*}
\bnfprod{list}{\bnfpn{listitems} \bnfor \bnfn{\ldots}}
\bnfprod{listitems}{\bnfpn{item} \bnfor \bnfpn{item} \bnfes \bnfts{;} \bnfes \bnfpn{listitems}}
\bnfprod{item}{\bnftd{description of item}}
\end{bnf*}

Here, \bnfpn{...} signifies produces, \bnfor is an or operator, \bnfn{...} are production names, and \bnfn{\ldots} represents the empty string. However, some BNF users prefer alternative terminologies, where \bnfpn{...} stands for is defined as, \bnfn{...} is a category name or nonterminal, and \bnfn{\ldots} is refered to as null or empty.

The above definition was created with the following code:

\begin{verbatim}
\usepackage{backnaur}
...
\begin{bnf*}
  \bnfprod{list}{\bnfpn{listitems} \bnfor \bnfn{\ldots}}
  \bnfprod{listitems}{\bnfpn{item} \bnfor \bnfpn{item} \bnfes \bnfts{;} \bnfes \bnfpn{listitems}}
  \bnfprod{item}{\bnftd{description of item}}
\end{bnf*}
\end{verbatim}

∗This work replaces \textit{Writing BNF Notation in \LaTeX}, which described a non-package method of BNF typesetting. This document corresponds to \texttt{backnaur} 1.1, dated 2012/12/12.
†adrian.robson@nepsweb.co.uk
Each BNF production is defined by a \texttt{bnfprod} command, which has two arguments giving its left and right sides. The right hand side of each production is specified with the commands described in §3. Terminal (\texttt{bnfts{;}}) and non-terminal (\texttt{bnfpn{item}}), elements are separated by spaces (\texttt{bnfsp}) and OR symbols (\texttt{bnfor}). The \texttt{bnfes} command gives the symbol for the empty string.

3 Package Commands

3.1 Loading and options

The package is loaded with

\texttt{\usepackage{backnaur}}

or

\texttt{\usepackage[<options>]{backnaur}}

Possible options are

\texttt{perp} The empty string symbol is \(\perp\)

\texttt{epsilon} The empty string symbol is \(\epsilon\)

\texttt{tstt} Terminal string typeface is typewriter

The defaults are: the empty string symbol is \(\lambda\), and the terminal string typeface is normal (roman).

3.2 Environments

\texttt{bnf} BNF productions are defined in a \texttt{bnf} or \texttt{bnf*} environment, which respectively give numbered and unnumbered lists of productions.

\begin{verbatim}
\begin{bnf}
<list of productions>
\end{bnf}
\end{verbatim}

\begin{verbatim}
\begin{bnf*}
<list of productions>
\end{bnf*}
\end{verbatim}

3.3 Productions

\texttt{bnfprod} A production is defined by \texttt{bnfprod}, which takes two arguments:

\texttt{bnfprod{<production name>{<production definition>}}}

3.4 Production definitions

The following commands are used to compose the right hand side of a production. They are deployed in the second argument of the \texttt{bnfprod} command.

\texttt{bnfpn} The \texttt{bnfpn} command generates a production name. It takes a single argument that is the name. It is used as follows:

\texttt{bnfpn{list item} \{list item\}}

\texttt{bnftd} There are three types of terminal item: a literal string, a descriptive phrase and an empty string. A literal terminal string is specified by the \texttt{bnftd} command, which takes a single argument. The \texttt{bnftd} command generates a descriptive phrase, as an alternative to a literal string. The \texttt{bnfes} command generates a token that represents the empty string. This is normally \(\lambda\), but it can be changed as a package option (see §3.1).
Some literal terminal strings can be abbreviated with the ‘skip’ token, which is generated by the `\bnfsk` command. This substitutes for a sequence of terminal characters. It is used like this:

```
\bnfts{A} \bnfsk \bnfts{Z}
```

All items are separated by an OR or a space. The `\bnfor` command generates the OR symbol, and the `\bnfsp` command introduces a space. A space can be considered equivalent to an AND operator.

```
\bnfpn{abc} \bnfor \bnfts{xzy}
\bnfpn{abc} \bnfsp \bnfts{xzy}
```

### 3.5 Inline expressions

The package’s definition commands can be typeset inline using maths mode, so the expression `$\bnfpn{name}$` will give `⟨name⟩`.

The `\bnfpo` command is provided so that the production operator `|=` can be printed independently from the `bnf` environment if required. The `\bnfprod` command cannot be used inline.

The `\bnfpo` command is provided so that the production operator `|=` can be printed independently from the `bnf` environment if required. The `\bnfprod` command cannot be used inline.

The `\bnfprod` command cannot be used inline. So the `\bnfpo` command is provided so that the production operator `|=` can be printed independently from the `bnf` environment if required.

### 3.6 Command summary

The commands that can be used to define a BNF production in a `bnf` or `bnf*` environment are as follows:

<table>
<thead>
<tr>
<th>Command</th>
<th>Operator</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\bnfpn{}</code></td>
<td>production name</td>
<td>⟨name⟩</td>
</tr>
<tr>
<td><code>\bnfor</code></td>
<td>OR operator</td>
<td></td>
</tr>
<tr>
<td><code>\bnfsk</code></td>
<td>skip</td>
<td>...</td>
</tr>
<tr>
<td><code>\bnfsp</code></td>
<td>space/AND operator</td>
<td></td>
</tr>
<tr>
<td><code>\bnfes</code></td>
<td>empty string</td>
<td>λ</td>
</tr>
<tr>
<td><code>\bnfts{}</code></td>
<td>terminal string</td>
<td>terminal</td>
</tr>
<tr>
<td><code>\bnftd{}</code></td>
<td>terminal description</td>
<td>description</td>
</tr>
<tr>
<td><code>\bnfpo</code></td>
<td>production operator</td>
<td></td>
</tr>
</tbody>
</table>
4 Example

A more significant example is the following definition of a \(\langle\text{sentence}\rangle\), where \(\langle\text{cchar}\rangle\) are countable characters, and \(\langle\text{ichar}\rangle\) are characters that should be ignored:

\[
\begin{align*}
\text{Sentence production:} \quad & \langle\text{sentence}\rangle \mid \langle\text{start}\rangle \langle\text{rest}\rangle. \\
\text{Start production:} \quad & \langle\text{start}\rangle \mid \langle\text{space}\rangle \mid \lambda \\
\text{Rest production:} \quad & \langle\text{rest}\rangle \mid \langle\text{word}\rangle \langle\text{space}\rangle \langle\text{rest}\rangle \mid \langle\text{word}\rangle \mid \lambda \\
\text{Word production:} \quad & \langle\text{word}\rangle \mid \langle\text{wchar}\rangle \langle\text{word}\rangle \mid \langle\text{wchar}\rangle \\
\text{Space production:} \quad & \langle\text{space}\rangle \mid \langle\text{schar}\rangle \langle\text{space}\rangle \mid \langle\text{schar}\rangle \\
\text{wchar production:} \quad & \langle\text{wchar}\rangle \mid \langle\text{cchar}\rangle \mid \langle\text{ichar}\rangle \\
\text{Schar production:} \quad & \langle\text{schar}\rangle \mid ' ' \mid '!' \mid '*' \mid '(' \mid ')' \mid '{' \mid '}' \mid ':' \mid ';' \mid '?' \mid ,
\end{align*}
\]

This creates the following BNF definition:

\[
\langle\text{sentence}\rangle \mid= \langle\text{start}\rangle \langle\text{rest}\rangle.
\]

\[
\langle\text{start}\rangle \mid= \langle\text{space}\rangle \mid \lambda
\]

\[
\langle\text{rest}\rangle \mid= \langle\text{word}\rangle \langle\text{space}\rangle \langle\text{rest}\rangle \mid \langle\text{word}\rangle \mid \lambda
\]

\[
\langle\text{word}\rangle \mid= \langle\text{wchar}\rangle \langle\text{word}\rangle \mid \langle\text{wchar}\rangle
\]

\[
\langle\text{space}\rangle \mid= \langle\text{schar}\rangle \langle\text{space}\rangle \mid \langle\text{schar}\rangle
\]

\[
\langle\text{wchar}\rangle \mid= \langle\text{cchar}\rangle \mid \langle\text{ichar}\rangle
\]

\[
\langle\text{schar}\rangle \mid= ' ' \mid '!' \mid '*' \mid '(' \mid ')' \mid '{' \mid '}' \mid ':' \mid ';' \mid '?' \mid ,
\]

4