The `backnaur` package

Adrian P. Robson

Version 3.1
18 June 2019

1 Introduction

The `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:

\[ \langle \text{list} \rangle \mid \langle \text{listitems} \rangle \mid \lambda \]

\[ \langle \text{listitems} \rangle \mid \langle \text{item} \rangle \mid (\langle \text{item} \rangle ; \langle \text{listitems} \rangle) \]

\[ \langle \text{item} \rangle \mid \text{description of item} \]

Here, \mid signifies produces, \mid is an or operator, \langle ... \rangle are production names, and \lambda represents the empty string. However, some BNF users prefer alternative terminologies, where \mid stands for is defined as, \langle ... \rangle is a category name or nonterminal, and \lambda is referred to as null or empty.

The above definition was created with the following code:

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

*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 §3.4 below. Terminal (\texttt{bnfts{;}}) and nonterminal (\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

\begin{itemize}
  \item \texttt{perp} The empty string symbol is \( \perp \)
  \item \texttt{epsilon} The empty string symbol is \( \epsilon \)
  \item \texttt{tsrm} Terminal string typeface is roman
  \item \texttt{altpo} Production operator is ::=
\end{itemize}

The defaults are: the empty string symbol is \( \lambda \), the production operator is |\(=\), and the terminal string typeface is typewriter.

3.2 Environments

\begin{itemize}
  \item \texttt{bnf} BNF productions are defined in a \texttt{bnf} or \texttt{bnf*} environment, which respectively give numbered or unnumbered lists of productions.
  \begin{verbatim}
  \begin{bnf}
  <list of productions>
  \end{bnf}
  \begin{bnf*}
  <list of productions>
  \end{bnf*}
  \end{verbatim}
\end{itemize}

3.3 Productions

\begin{itemize}
  \item \texttt{bnfprod} A production is defined by \texttt{bnfprod} or \texttt{bnfprod*}, which respectively give a numbered or unnumbered line in the \texttt{bnf} environment. They have identical unnumbered behaviour in the \texttt{bnf*} environment. They take two arguments:
  \begin{verbatim}
  \bnfprod{<production name>}{<production definition>}
  \bnfprod*[<production name>]{<production definition>}
  \end{verbatim}
  \item \texttt{bnfmore} A production can be continued on addition lines by \texttt{bnfmore} or \texttt{bnfmore*}, which respectively give a numbered or unnumbered line in the \texttt{bnf} environment. They are treated the same in the \texttt{bnf*} environment. They take one arguments:
  \begin{verbatim}
  \bnfmore{<production definition>}
  \bnfmore*[<production definition>]
  \end{verbatim}
\end{itemize}
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:

\begin{verbatim}
\texttt{bnfpn\{list item\} \langle list item\rangle}
\end{verbatim}

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 $\texttt{bnftm}$ command, which takes a single argument. By default literal terminal strings are printed in typewriter font, but this can be changed as a package option (see §3.1). 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 to $\epsilon$ or $\perp$ as a package option (see §3.1).

\begin{verbatim}
\texttt{bnfts\{terminal\}} \texttt{terminal} \\
\texttt{bnftd\{description\}} \texttt{description} \\
\texttt{bnfes} \texttt{\lambda}
\end{verbatim}

$\texttt{bnfsk}$

Some literal terminal strings can be abbreviated with the ‘skip’ token, which is generated by the $\texttt{bnfsk}$ command. This substitutes for a sequence of terminal characters. It is used like this:

\begin{verbatim}
\texttt{bnfts\{A\} \bnfsk \bnfts\{Z\}} \texttt{A...Z}
\end{verbatim}

$\texttt{bnfor}$

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

\begin{verbatim}
\texttt{bnfpn\{abc\} \bnfor \bnfts\{xzy\}} \langle abc \rangle | xzy \\
\texttt{bnfpn\{abc\} \bnfsp \bnfts\{xzy\}} \langle abc \rangle xzy
\end{verbatim}

3.5 Inline expressions

The $\texttt{bnfprod}$ and $\texttt{bnfmore}$ macros cannot be used inline, so the $\texttt{bnfpn}$ and $\texttt{bnfpo}$ macros are provided to support typesetting productions inline using maths mode. The production’s name can be typeset with $\texttt{bnfpn\{name\}}$ and the production operator with $\texttt{bnfpo}$. By default the production operator is $\mid$, but it can be changed to $\ ::= $ with a package option (see §3.1). The right side of the production can be defined with the usual macros (see §3.4). So $\$\texttt{bnfpn\{name\} \bnfpo \bnftd\{description\}}\$ \ gives \ $\langle name \rangle \ ::= \ description.$

3.6 Command summary

The commands that can be used to define a BNF production in a $\texttt{bnf}$ or $\texttt{bnf*}$ environment are as follows:
\begin{bnf*}
  \bnfprod{sentence} \bnfprod{start} \bnfprod{rest} \bnfts{.} \\
  \bnfprod{start} \bnfor \bnfes \\
  \bnfprod{rest} \bnfor \bnfes \\
  \bnfprod{word} \bnfpn{space} \bnfpn{space} \bnfpn{space} \bnfes \\
  \bnfprod{word} \bnfor \bnfes \\
  \bnfprod{wchar} \bnfpn{space} \bnfpn{space} \bnfpn{space} \bnfpn{space} \bnfes \\
  \bnfprod{wchar} \bnfor \bnfes \\
  \bnfprod{cchar} \bnfes \\
  \bnfprod{ichar} \bnfes \\
  \bnfprod{char} \bnfes \\
  \bnfmore{\bnfts{<\hspace{1em}>} \bnfes} \bnfes \\
  \bnfes \\
  \bnfes \\
\end{bnf*}

This creates the following BNF definition:

\begin{align}
  \langle \text{sentence} \rangle & \mid \langle \text{start} \rangle \cdot \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
\end{align}

4 Example

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

\begin{align}
  \langle \text{sentence} \rangle & \mid \langle \text{start} \rangle \cdot \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
\end{align}
Notice the kludge in production 9. We use \texttt{`\hspace{1em}'} to typeset a representation for a space character. This is needed because we do not want to print in typewriter font, which would imply the quotes were part of an actual terminal string. The \texttt{\texttt{\texttt{\texttt{\texttt{}}}{}{}}} is needed because are in maths mode.

5 Terminal string characters

The characters used with \texttt{\texttt{\texttt{\texttt{}}}{}{}} (terminal string) are just standard LaTeX that is typeset in either a roman or typewriter font. This means we might have to use some escape pairs and a few special characters. Apostrophes and speech marks can be confusing. There are some of the possibilities:

\begin{verbatim}
alpha  \texttt{\texttt{\texttt{\texttt{}}}{}{}}{abcdABCD}  abcdABCD  abcdABCD
numeric \texttt{\texttt{\texttt{\texttt{}}}{}{}}{01234}  01234  01234
simple \texttt{\texttt{\texttt{\texttt{}}}{}{}}{<>[]()+-=}  <>[]()+-=  <>[]()+-=
simple \texttt{\texttt{\texttt{\texttt{}}}{}{}}{@!?/,;:}  @!?/,;:  @!?/,;:
escaped \texttt{\texttt{\texttt{\texttt{}}}{}{}}{\{\}$_0$\%$_0$\&$_0$\_$_0$\#}  \{}\$\%\&\_\#  \{}\$\%\&\_\#
quotes \texttt{\texttt{\texttt{\texttt{}}}{}{}}{`\'\''\''}  `\'\''\''  `\'\''\''
quotes \texttt{\texttt{\texttt{\texttt{}}}{}{}}{\textquotesingle}  /quotesingle.ts1  /quotesingle.ts1
pound \texttt{\texttt{\texttt{\texttt{}}}{}{}}{\texttt{\texttt{\texttt{\texttt{}}}{}{}}}{}{pounds}  £  £
hat \texttt{\texttt{\texttt{\texttt{}}}{}{}}{\texttt{\texttt{\texttt{\texttt{}}}{}{}}}{}{textasciicircum}  ^  ^
backslash \texttt{\texttt{\texttt{\texttt{}}}{}{}}{\texttt{\texttt{\texttt{\texttt{}}}{}{}}}{}{\texttt{\texttt{\texttt{\texttt{}}}{}{}}}{}{textasciitilde}  \  \
tilde \texttt{\texttt{\texttt{\texttt{}}}{}{}}{\texttt{\texttt{\texttt{\texttt{}}}{}{}}}{}{\texttt{\texttt{\texttt{\texttt{}}}{}{}}}{}{textasciitilde}  ~  ~
\end{verbatim}

The \texttt{\texttt{\texttt{\texttt{}}}{}{}} symbol needs the textcomp package, which provides lots of other interesting symbols. Consult the excellent \texttt{The Comprehensive LATEX Symbol List} by Scott Pakin for more information.