Bison Quick Reference

Starting Bison

To use Bison, type: `bison filename`
Options can be used as: `bison options filename`

Command Line Options

- Display usage information: `-h`
- Display version information: `-V`
- Generate token and YYSTYPE definitions: `-d`
- Prepend a prefix to external symbols: `-p prefix`
- Don't put #line directives in the parser: `-l`
- Specify the output file: `-o filename`
- Debug or trace mode: `-t`
- Verbose description of the parser: `-v`
- Emulate yacc (generate y.tab.* files): `-y`

Note: The token and YYSTYPE definitions are generated to a file called `y.tab.h` if the `-y` option is used, otherwise it will have the format `name.tab.h`, where `name` is the leading part of the parser definition filename.

Definitions

- Declare a terminal symbol: `%token <t> n`
- Declare a terminal symbol, and define its association: `association <t> n`
- Generate a reentrant (pure) parser: `%pure_parser`
- Define the union of all data types used in the parser: `%union{field list}`
- Tell bison where to start parsing: `%start m`
- Tell bison the data type of symbols: `%type <t> s1...sn`

In the above, `t` is a type defined in the `%union` definition, `n` is a terminal symbol name, `m` is a non-terminal symbol name, and `association` can be one of `%left`, `%right`, or `%nonassoc`.

The `<t>` after `%token`, `%left`, `%right` and `%nonassoc` is optional. Additionally, precedence may be overridden in the rules with `%prec` commands.

Parser Definition Files

The general form for a parser definition is:

```c
{%
    /* Initial C code. */
%
}

%{
    Token and type definitions
%
    Rule definition 1
    
    Rule definition n
%
    /* Other C code. */
```
Rule definitions
Rules take the form:

\[
\text{non-terminal} : \quad \text{statement } 1 \\
| \text{statement } 2 \\
| \vdots \\
| \text{statement } n \\
\]

Where \textit{statements} can be either empty, or contain a mixture of C code (enclosed in {...}), and the symbols that make up the non-terminal. For example:

\[
\text{expression} : \quad \text{number } '+' \text{ number} \{ \mathit{$$} = \mathit{$$} 1 + \mathit{$$} 3 \} \\
| \text{number } '-' \text{ number} \{ \mathit{$$} = \mathit{$$} 1 - \mathit{$$} 3 \} \\
| \text{number } '/' \text{ number} \{ \mathit{$$} = \mathit{$$} 1 / \mathit{$$} 3 \} \\
| \text{number } '*' \text{ number} \{ \mathit{$$} = \mathit{$$} 1 * \mathit{$$} 3 \} \\
\]

For altering the precedence of a symbol use:

\%prec name

For example:

\[
\text{foo} : \quad \text{gnu bar gnu} \quad \%\text{prec bar} \\
\]

Grammar Variables and Symbols

Recognize an error & continue parsing \quad \text{name}\quad \text{error}
Access data associated with a symbol. \quad \mathit{$$$}, \mathit{$_0.._n$}
Access a field of the data union. \quad \mathit{$$$}.t, \mathit{$_0.t.._n.t$}
Access symbol's location. \quad \mathit{0$$}, \mathit{00...0n}
Access data's line location. \quad \mathit{x.line\_spec}
Access data's column location. \quad \mathit{x.column\_spec}

Where \(t\) is a type defined in the \%union, \(n\) is a number, \(x\) is a number or $\$, \mathit{line\_spec} one of first\_line and last\_line, and \mathit{column\_spec} is specified as either first\_column or last\_column.

Variables and Types

Current look ahead token. \quad \text{uchar}
Debug mode flag. \quad \text{yydebug}
Data associated with the current token. \quad \text{yyval}
Source position of current token. \quad \text{yyloc}
Number of errors encountered. \quad \text{yyerrors}
Position information type. \quad \text{YYLTYPE}
Data type associated with symbols. \quad \text{YYSTYPE}

Functions

User defined error handler. \quad \text{int yyyerror(char *)}
User defined lexical analyzer. \quad \text{int yylex()}
The grammar parser. \quad \text{int yyparse()}

Macros

Quit parsing immediately. Return 1. \quad \text{YYABORT}
Quit parsing immediately. Return 0. \quad \text{YYACCEPT}
Pretend a syntax error occurred. \quad \text{YYERROR}
Value in \texttt{uchar} if no look-ahead token. \quad \text{YYEMPTY}
Clear previous look ahead token. \quad \texttt{yyclearin}
Recover normally from an error. \quad \text{yyerrok}

Copyright © 2001 Free Software Foundation, Inc.
November 2001, Bison Refcard Version 0.5
designed by Brendan Kehoe and Gavin Nicol
for Bison 1.31
Permission is granted to make and distribute copies of this card
provided the copyright notice and this permission notice are
preserved on all copies.
For information, write to:
Free Software Foundation, Inc.
59 Temple Place - Suite 330
Boston, MA 02111-1307 USA