Changelog:

Represent lex_state by bit field instead of serial integer enum so that single or multiple values can be checked together using unifying macro IS_lex_state(). States remain mutually exclusive.

Example:

Each use of current macro IS_BEG()..
#define IS_BEG() (lex_state == EXPR_BEG || lex_state == EXPR_MID || lex_state == EXPR_VALUE || lex_state == EXPR_CLASS)
.. results in up to 4 tests which can be reduced to 1.

An extreme use case..

If (IS_BEG() || lex_state == EXPR_DOT || IS_ARG()) {

.. requires up to 7 tests on lex_state.

To my knowledge, compilers can't optimize this. As a bit field, there's no need to:

if (IS_lex_state(EXPR_BEG_ANY | EXPR_ARG_ANY | EXPR_DOT))

.. reduces to: if (lex_state & (test_bits)) // TRUE/FALSE

All changes in this patch were scripted to eliminate the possibility of typos. Where some replaced sections looked similar or the same, they were verified with MD5sum before applying! Multiple state tests were merged using simple, strict logic and any "surprise" code would have been left unchanged.

Therefore, it should not be necessary to check every line of the patch; I hope that a check of a random sample will give confidence in the rest.

There were several repeated switches which could have been replaced with:

lex_state = (IS_lex_state(EXPR_FNAME | EXPR_DOT)) ? EXPR_ARG : EXPR_BEG

but I assumed you would prefer if-else for flexibility and legibility:

if (IS_lex_state(EXPR_FNAME | EXPR_DOT)) {
  lex_state = EXPR_ARG;
} else {
  lex_state = EXPR_BEG;
}

Say what you don't like about anything - it might be easy to change. :)

Wouldn't it be nice to use some spare bits to combine other state?
if ((lex_state == EXPR_BEG && !cmd_state) || ...
if (IS_ARG() && space_seen && ...

- I haven't explored those, yet.

My motivation was not to reduce processor heat. :) In the future, if we need to break the dependence on Yacc/Bison, these state transitions might need to become token information and the change makes them far easier to store. (e.g. Any of ‘THESE states’ => ‘THIS state’)

daz
=end

Associated revisions

Revision 197e7b81 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37338 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37338 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]

Revision 37338 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]

Revision 37338 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]

Revision 37338 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]

Revision 37338 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]

Revision 37338 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]

Revision 37338 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]

Revision 37338 - 10/27/2012 01:07 AM - nobu (Nobuyoshi Nakada)
parse.y: bit field lex_state
  - parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].  
    [Feature #1493]

History

#1 - 09/18/2009 04:57 AM - marcandre (Marc-Andre Lafortune)
- Category set to core
- Assignee set to matz (Yukihiro Matsumoto)
- Priority changed from Normal to 3

#2 - 04/04/2010 01:16 AM - znz (Kazuhiro NISHIYAMA)
- Category set to core
- Status changed from Open to Assigned
- Target version set to 2.0.0

#3 - 02/14/2012 12:26 AM - mame (Yusuke Endoh)
- Assignee changed from matz (Yukihiro Matsumoto) to nobu (Nobuyoshi Nakada)

Nobu, aren't you interested in this? Could you please review and import the patch if it looks good to you?

This seems to be just a patch for refactoring. Though, the purpose is not so clear to me. (Who is planning to break the dependence on bison?) But, the patch seems benign (I read only the ticket, not the patch itself).

--
Yusuke Endoh mame@tsg.ne.jp

#4 - 10/27/2012 10:08 AM - nobu (Nobuyoshi Nakada)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r37338.
Dave, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

**parse.y: bit field lex_state**

- parse.y (enum lex_state_e): [EXPERIMENTAL] lex_state as bit field / IS_lex_state() macro. based on the patch by Dave B in [ruby-core:23503].
  [Feature #1493]

**Files**

<table>
<thead>
<tr>
<th>File</th>
<th>13.2 KB</th>
<th>05/20/2009</th>
<th>daz (Dave B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS_LEX.patch</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>