Ruby master - Feature #14808

Last token of endless range should have EXPR_END

06/02/2018 01:23 AM - aycabta (aycabta .)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	

Description

In 2.5.1:

```
irb(main):001:0> require 'ripper'
irb(main):002:0> pp Ripper.lex("case 5\nwhen 3..\n  puts(true)\nend\n")
[[[1, 0], :on_kw, "case", EXPR_BEG],
 [[1, 4], :on_sp, " ", EXPR_BEG],
 [[1, 5], :on_int, "5", EXPR_END|EXPR_ENDARG],
 [[1, 6], :on_nl, "\n", EXPR_BEG],
 [[2, 0], :on_kw, "when", EXPR_BEG],
 [[2, 4], :on_sp, " ", EXPR_BEG],
 [[2, 5], :on_int, "3", EXPR_END|EXPR_ENDARG],
 [[2, 6], :on_op, "..", EXPR_BEG],
 [[2, 8], :on_ignored_nl, "\n", EXPR_BEG],
 [[3, 0], :on_sp, "  ", EXPR_BEG],
 [[3, 2], :on_ident, "puts", EXPR_ARG],
 [[3, 6], :on_lparen, "(", EXPR_BEG|EXPR_LABEL],
 [[3, 7], :on_kw, "true", EXPR_END],
 [[3, 11], :on_rparen, ")", EXPR_ENDFN],
 [[3, 12], :on_nl, "\n", EXPR_BEG],
 [[4, 0], :on_kw, "end", EXPR_END],
 [[4, 3], :on_nl, "\n", EXPR_BEG]]
```

This is invalid code in 2.5.1, so I understand this result.

In 63451:

```
irb(main):001:0> require 'ripper'
irb(main):002:0> pp Ripper.lex("case 5\nwhen 3..\n  puts(true)\nend\n")
[[[1, 0], :on_kw, "case", EXPR_BEG],
 [[1, 4], :on_sp, " ", EXPR_BEG],
 [[1, 5], :on_int, "5", EXPR_END],
 [[1, 6], :on_nl, "\n", EXPR_BEG],
 [[2, 0], :on_kw, "when", EXPR_BEG],
 [[2, 4], :on_sp, " ", EXPR_BEG],
 [[2, 5], :on_int, "3", EXPR_END],
 [[2, 6], :on_op, "..", EXPR_BEG],
 [[2, 8], :on_ignored_nl, "\n", EXPR_BEG],
 [[3, 0], :on_sp, "  ", EXPR_BEG],
 [[3, 2], :on_ident, "puts", EXPR_ARG],
 [[3, 6], :on_lparen, "(", EXPR_BEG|EXPR_LABEL],
 [[3, 7], :on_kw, "true", EXPR_END],
 [[3, 11], :on_rparen, ")", EXPR_ENDFN],
 [[3, 12], :on_nl, "\n", EXPR_BEG],
 [[4, 0], :on_kw, "end", EXPR_END],
 [[4, 3], :on_nl, "\n", EXPR_BEG]]
```

This is correct code in this revision.

I think that lex_state of the last token of endless range, [[2, 6], :on_op, "..", EXPR_BEG], it should be EXPR_END. Because it's the end of an argument. It's important for REPL, RDoc, and so on.

But lex_state is parser matter in parse.y. How about this for the parser of Ruby compiler?

Related issues:

History

#1 - 06/02/2018 04:33 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

That code is valid syntax in both versions, and it is *not* an endless range in the trunk as well as 2.5.
As .. is an infix operator and requires the RHS, newlines following it are just ignored.
So it equals to:

```
case 5
when (3..puts(true))
end
```

It results in an error at runtime in 2.5, not a syntax error.

```
$ ruby2.5 -rripper -e 'case 5' -e 'when 3..' -e  'puts(true)' -e end
true
Traceback (most recent call last):
-e:3:in `<main>': bad value for range (ArgumentError)
```

To put an endless range there, use parentheses.

#2 - 06/05/2018 06:26 AM - aycabta (aycabta .)

- Related to Bug #14824: Endless Range Support in irb added