

Ruby - Bug #21337

Using `not` on the RHS of a logical operator becomes valid syntax with Prism

05/14/2025 04:57 AM - koic (Koichi ITO)

Status:	Assigned	
Priority:	Normal	
Assignee:	prism	
Target version:		
ruby -v:	ruby 3.4.3 (2025-04-14 revision d0b7e5b6a0) +PRISM [x86_64-darwin24]	Backport: 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

The following syntax behavior differs between Ruby 3.3 (parse.y by default) and Ruby 3.4 (Prism by default).

```
if true && not true; end
```

Expected

In Ruby 3.3 and earlier, the following code results in a syntax error. The default parser is parse.y.

```
$ ruby -vce 'if true && not true; end'
ruby 3.3.8 (2025-04-09 revision b200bad6cd) [x86_64-darwin24]
-e:1: syntax error, unexpected `true', expecting '('
if true && not true; end
ruby: compile error (SyntaxError)
```

The key point is that not is used on the right-hand side of a logical operator.

Actual

When the default parser is switched to Prism, it is accepted as valid syntax with Ruby 3.4+ (Prism by default):

```
$ ruby -vce 'if true && not true; end'
ruby 3.4.3 (2025-04-14 revision d0b7e5b6a0) +PRISM [x86_64-darwin24]
Syntax OK

$ ruby -vce 'if true && not true; end'
ruby 3.5.0dev (2025-05-13T02:05:19Z master 9b8c846bdf) +PRISM [x86_64-darwin24]
Syntax OK
```

Specifying parse.y causes a syntax error, as before 3.3 (parse.y by default):

```
$ ruby -vce 'if true && not true; end'
ruby 3.3.8 (2025-04-09 revision b200bad6cd) [x86_64-darwin24]
-e:1: syntax error, unexpected `true', expecting '('
if true && not true; end
ruby: compile error (SyntaxError)

$ ruby --parser=parse.y -vce 'if true && not true; end'
ruby 3.5.0dev (2025-05-13T02:05:19Z master 9b8c846bdf) [x86_64-darwin24]
-e:1: syntax error, unexpected 'true', expecting '('
if true && not true; end
ruby: compile error (SyntaxError)
```

The behavior seen in the parse.y producing a syntax error might be probably the expected one. Any case, it is likely that users would not expect such a discrepancy between parsers.

This issue was noticed in the context of the following RuboCop issue:
<https://github.com/rubocop/rubocop/issues/14177>

History

#1 - 05/14/2025 07:03 AM - byroot (Jean Boussier)

- Assignee set to prism

#2 - 05/22/2025 03:39 AM - Dan0042 (Daniel DeLorme)

I prefer the behavior of Prism here. Intuitively, true && not true seems like it should be valid just like true and not true

#3 - 05/28/2025 05:07 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

#4 - 06/05/2025 08:53 AM - matz (Yukihiro Matsumoto)

I am against this prism behavior. It introduces ambiguity for not a && b (might be not (a && b) or (not a) && b). I don't think we can have formal definition of this not behavior.

Matz.

#5 - 06/05/2025 10:52 AM - mame (Yusuke Endoh)

p(not 1) is also allowed in Ruby 3.4, but this should also be prohibited, [@matz \(Yukihiro Matsumoto\)](#) said.

[@kddnewton \(Kevin Newton\)](#) [@tenderlovmaking \(Aaron Patterson\)](#) [@eileencodes \(Eileen Uchitelle\)](#) Can you fix this issue soon? We need to backport to 3.4 to stop Ruby 3.4 users writing such code.