

Ruby - Feature #18242

Parser makes multiple assignment sad in confusing way

10/06/2021 04:39 AM - danh337 (Dan H)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		
Description		
Example:		
<pre>a, b = 2, 1 if 1 < 2 # Works a, b = [2, 1] if 1 < 2 # Works (a, b) = 2, 1 if 1 < 2 # Works (a, b) = [2, 1] if 1 < 2 # Works (a, b = [2, 1]) if 1 < 2 # Works a, b = 2, 1 unless 2 < 1 # Works a, b = [2, 1] unless 2 < 1 # Works (a, b) = 2, 1 unless 2 < 1 # Works (a, b) = [2, 1] unless 2 < 1 # Works (a, b = [2, 1]) unless 2 < 1 # Works 1 < 2 and a, b = 2, 1 # SyntaxError 1 < 2 and a, b = [2, 1] # SyntaxError 1 < 2 and (a, b) = 2, 1 # SyntaxError 1 < 2 and (a, b) = [2, 1] # SyntaxError (1 < 2) and a, b = 2, 1 # SyntaxError (1 < 2) and a, b = [2, 1] # SyntaxError (1 < 2) and (a, b) = 2, 1 # SyntaxError (1 < 2) and (a, b) = [2, 1] # SyntaxError 1 < 2 and (a, b = 2, 1) # Works 1 < 2 and (a, b = [2, 1]) # Works 2 < 1 or a, b = 2, 1 # SyntaxError 2 < 1 or a, b = [2, 1] # SyntaxError 2 < 1 or (a, b) = 2, 1 # SyntaxError 2 < 1 or (a, b) = [2, 1] # SyntaxError (2 < 1) or a, b = 2, 1 # SyntaxError (2 < 1) or a, b = [2, 1] # SyntaxError (2 < 1) or (a, b) = 2, 1 # SyntaxError (2 < 1) or (a, b) = [2, 1] # SyntaxError 2 < 1 or (a, b = 2, 1) # Works 2 < 1 or (a, b = [2, 1]) # Works</pre>		
Based on the precedence rules I've been able to find, all of these should work.		
Believe it or not, there are cases where using and or or in a stanza of lines is much more readable.		
Should the parser allow all of these? See attached driver script to reproduce this output.		

History

- #1 - 10/06/2021 04:28 PM - jeremyevans0 (Jeremy Evans)
- File and-or-masgn-18242.diff added
 - Tracker changed from Bug to Feature
 - ruby -v deleted (ruby 3.0.2p107 (2021-07-07 revision 0db68f0233) [x86_64-linux])
 - Backport deleted (2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN)

The parser is currently not setup to handle this, so I don't think it is a bug. You would have to add new parser rules to allow it, so I would consider this a feature request and not a bug. Attached is a patch that implements the necessary parser rules, without causing any conflicts. However, the patch doesn't include the parts needed to make ripper work, as I couldn't figure that out. @nobu (Nobuyoshi Nakada), any chance you could add the necessary code to make ripper work?

#2 - 10/06/2021 08:56 PM - danh337 (Dan H)

jeremyevans0 (Jeremy Evans) wrote in [#note-1](#):

[...] Attached is a patch that implements the necessary parser rules, without causing any conflicts. However, the patch doesn't include the parts needed to make ripper work, as I couldn't figure that out. [@nobu \(Nobuyoshi Nakada\)](#), any chance you could add the necessary code to make ripper work?

[@jeremyevans0 \(Jeremy Evans\)](#) you rock. Thanks sir.

#3 - 10/07/2021 03:44 PM - nobu (Nobuyoshi Nakada)

That patch seems not including `1 < 2` and `a = 1, 2`.

<https://github.com/ruby/ruby/pull/4945>

#4 - 10/08/2021 07:40 PM - danh337 (Dan H)

nobu (Nobuyoshi Nakada) wrote in [#note-3](#):

That patch seems not including `1 < 2` and `a = 1, 2`.

<https://github.com/ruby/ruby/pull/4945>

Thank you [@nobu \(Nobuyoshi Nakada\)](#) !

#5 - 10/09/2021 02:36 AM - danh337 (Dan H)

[@nobu \(Nobuyoshi Nakada\)](#) I built your PR branch and confirmed that this new parsing works great with the driver.rb attached.

Will this patch have to wait for the 3.1.0 release? Sorry I'm not sure how the whole patching and backporting processes work. (But I'm pretty sure it can get complicated.) If these were syntax errors in the past, should the patch be safe for stable versions too?

#6 - 10/09/2021 02:40 AM - danh337 (Dan H)

- File driver.rb added

Attached is an improved driver. Again, every test case works with nobu's branch.

#7 - 10/09/2021 07:58 AM - nobu (Nobuyoshi Nakada)

I agree with [@jeremyevans0 \(Jeremy Evans\)](#) on that this is not a bug. Feature requests need the approval by Matz to merge.

#8 - 10/15/2024 06:02 PM - danh337 (Dan H)

These and & or operators are good things. They are much more readable for one-liners in some cases.

[@matz \(Yukihiro Matsumoto\)](#) [@nobu \(Nobuyoshi Nakada\)](#) I know you are super busy, but is there any way to move this forward?

Files

driver.rb	1.17 KB	10/06/2021	danh337 (Dan H)
and-or-masgn-18242.diff	963 Bytes	10/06/2021	jeremyevans0 (Jeremy Evans)
driver.rb	2.1 KB	10/09/2021	danh337 (Dan H)