Ruby master - Bug #10388

Operator precedence problem in multiple assignment (massign)

10/15/2014 03:35 AM - knu (Akinori MUSHA)

Status:	Open		
Priority:	Normal		
Assignee:	matz (Yukihiro Matsumoto)		
Target version:			
ruby -v:	ruby 2.2.0dev (2014-10-13 trunk 47904) [x86_64-freebsd10]	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN

Description

I understand it wouldn't be easy to fix this, but since I happened to find it here it goes.

- a, b = c = 1, 2 is currently taken as a, b = (c = 1), 2; I'd expect it to be taken as a, b = (c = 1, 2).
- a, b = *c = 1, 2 is currently taken as a, b = *(c = 1), 2; I'd expect it to be taken as a, b = *(c = 1, 2) or even a, b = (*c = 1, 2).
- a, b = c, d = 1, 2 is currently taken as a, b = (c), (d = 1), 2; I'd expect it to be taken as a, b = (c, d = 1, 2).

Should they be fixed/changed or not, issuing a warning would be greatly helpful.

History

#1 - 08/20/2019 08:33 PM - jeremyevans0 (Jeremy Evans)

I tried working on this a couple weeks ago and I don't believe the current LALR(1) parser can support it. Attempting to modify the parser to support the behavior you desire leads to many shift/reduce or reduce/reduce conflicts. It is possible that switching from the default LALR(1) parser to a GLR parser (which bison also supports) may allow for the behavior your desire, but I'm not sure what the ramifications of that are. It's also possible there is a way to introduce this behavior with the existing LALR(1) parser, and I am just not aware of it, as I do not have much experience in this area.