Ruby - Bug #21382

Syntax for arguments in || is more strict than arguments in ()

05/29/2025 07:29 AM - herwin (Herwin W)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 3.4.4 (2025-05-14 revision a38531fd3f) +PRISM [x86_64-linux] (seen in older versions too)	**Backport:**	3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

```
p ->(x, y = x + 1) { x + y }.call(1)
```

This works fine, this gives y the value of x + 1 and prints 3. The same thing can be used in method definitions.

```
p lambda { |x, y = x + 1| x + y }.call(1)
```

This is the same thing, but results in a parse error (both prism and parse.y):

```
# Prism:
p lambda { |x, y = x + 1| x + y }.call(1)
                    ^ expected the block parameters to end with `|`
                     ^ unexpected '+', ignoring it

# Parse.y:
syntax error, unexpected '+', expecting '|'
```

It works if the default is a single statement (wrapping it in parentheses), so it looks to be purely a grammatical issue

```
p lambda { |x, y = (x + 1)| x + y }.call(1)
```

History

#1 - 06/05/2025 08:40 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Closed

Unfortunately, | is a binary-or operator too. So we have to be more strict to avoid ambiguity and syntax conflict.

Matz.