Ruby master - Bug #18883
parse.y: trailing comma cannot coexist with star

06/28/2022 03:44 AM - qnighy (Masaki Hara)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>ruby -v:</td>
<td>ruby 3.1.2p20 (2022-04-12 revision 4491bb740a) [x86_64-linux]</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN</td>
</tr>
</tbody>
</table>

Description

The following code is a syntax error:

```
*x, y, = 1, 2  # syntax error, unexpected '='
```

although the following:

```
x, y = 1, 2  # OK
x, y, = 1, 2  # OK
*x, y = 1, 2  # OK
# *x, y, = 1, 2
```

In my understanding, the trailing comma does nothing unless the lhs becomes otherwise a single assignment.

Therefore it is natural to allow the trailing comma whether * exists or not.

History

#1 - 06/28/2022 07:25 AM - zverok (Victor Shepelev)

As far as I understand, trailing comma is an equivalent of *_. (unpack the rest and drop it):

```
x, = [1, 2, 3]
# is the same as
x, "_" = [1, 2, 3]
# put 1 in x, ignore the rest, it is the shortest way to do it
```

So the fact that it is prohibited syntactically is in line with the fact that *x, y, *z = ... are also prohibited: Ruby can't handle two unpacks in one assignment.

If this would be allowed, what would be the semantics of it?...

```
*x, y, = 1, 2, 3
*x, y, *z = 1, 2, 3
```

What it should put in x and y?...

#2 - 07/14/2022 08:00 PM - jeremyevans0 (Jeremy Evans)

zverok (Victor Shepelev) wrote in #note-1:

As far as I understand, trailing comma is an equivalent of *_. (unpack the rest and drop it):

```
x, = [1, 2, 3]
# is the same as
x, "_" = [1, 2, 3]
# put 1 in x, ignore the rest, it is the shortest way to do it
```

Close. *_. will set a local variable, and allocate an extra array. x, = [1, 2, 3] is actually the same as x, * = [1, 2, 3] (same VM instructions). However, this doesn't fully explain the issue, because x = (single assignment) and x, = (multiple assignment) are significantly different. It would be better to compare with:

```
x, a = [1, 2, 3]
x, a, = [1, 2, 3]
x, a, * = [1, 2, 3]
```
All of these result in the same VM instructions, indicating that a trailing comma shouldn't make a difference.

That being said, I don't think this is a bug, but I wouldn't be opposed to modifying the parser to support this as a feature, assuming it is doable.