### Ruby master - Feature #15918

#### Pattern matching for Set

06/12/2019 01:44 PM - marcandre (Marc-Andre Lafortune)

<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>ktsj (Kazuki Tsujimoto)</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

#### Description

Currently, Set does not respond to deconstruct. Shouldn't we implement it using to_a?

```ruby
require 'set'
case Set[1, 2, 3]
in [1, 2, 3]
  p "match"
else
  p "no match"
end
# => "no match", should be "match"
```

#### Related issues:

- Related to Ruby master - Feature #14912: Introduce pattern matching syntax - Closed

#### History

**#1 - 06/12/2019 02:02 PM - Eregon (Benoit Daloze)**

Did you mean in Set[1, 2, 3] ?

**#2 - 06/12/2019 02:20 PM - marcandre (Marc-Andre Lafortune)**

Eregon (Benoit Daloze) wrote:

> Did you mean in Set[1, 2, 3] ?

I didn't, but it should match too; it's the same as my example but with the added constraint that the object should be a descendant a Set. Note that in Set[1, 2, 3] does not call Set.[](1, 2, 3)...

**#3 - 06/12/2019 02:58 PM - jeremyevans0 (Jeremy Evans)**

Sets are supposed to be unordered (any ordering is an implementation detail). If Set[1, 2, 3] matches in your example, so should Set[3, 2, 1], since Set[1, 2, 3] == Set[3, 2, 1]. We could attempt to sort the elements of the set before pattern matching, but some sets contain unsortable elements (e.g. elements of different types). If pattern matching can work correctly when using in Set[...], then maybe this would be desirable, but I'm not sure if that is possible.

To answer your question, in my opinion, yes, we shouldn't implement deconstruct using to_a.

**#4 - 06/12/2019 11:38 PM - jeremyevans0 (Jeremy Evans)**

- Backport deleted (2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN)
- Tracker changed from Bug to Feature

**#5 - 06/14/2019 11:40 PM - ktsj (Kazuki Tsujimoto)**

- Related to Feature #14912: Introduce pattern matching syntax added

**#6 - 07/29/2019 08:12 AM - ko1 (Koichi Sasada)**

- Assignee set to ktsj (Kazuki Tsujimoto)