Ruby master - Bug #7343

Enumerator::Lazy seems to nest herself infinitely

11/13/2012 04:42 PM - udzura (Uchio KONDO)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Rejected</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>2.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>ruby 2.0.0dev (2012-11-13)</td>
</tr>
</tbody>
</table>

| Backport:          |                             |

**Description**

Hello, Ruby dev team.

I tried Ruby 2.0 trunk, and found Enumerator::Lazy a bit weird.

```
> [1, 2, 3].cycle.lazy.take 5
# => #<Enumerator::Lazy: #<Enumerator::Lazy: #<Enumerator: [1, 2, 3]:cycle>>:take(5)>

> [1, 2, 3].cycle.lazy.take_while { |i| i < 3 }.lazy.take(2)
# => #<Enumerator::Lazy: #<Enumerator::Lazy: #<Enumerator::Lazy: #<Enumerator: [1, 2, 3]:cycle>>:take_while>:take(2)>

> [1, 2, 3].cycle.lazy.take_while { |i| i < 3 }.lazy.take(2).lazy.take(3).lazy
# => #<Enumerator::Lazy: #<Enumerator::Lazy: #<Enumerator::Lazy: #<Enumerator::Lazy: #<Enumerator: [1, 2, 3]:cycle>>:take_while>:take(2)>:take(3)>
```

Is this type of infinite nesting desirable?

Thank you,
Uchio KONDO udzura@udzura.jp

**History**

#1 - 11/13/2012 05:59 PM - udzura (Uchio KONDO)

I understand Enumerator::Lazy's behavior via

- [https://twitter.com/mashiro/status/268271989964292096](https://twitter.com/mashiro/status/268271989964292096)
- [https://gist.github.com/1939610](https://gist.github.com/1939610)

Sorry for making a noise...

#2 - 11/14/2012 12:44 AM - marcandre (Marc-Andre Lafortune)

- Status changed from Open to Rejected