Ruby - Bug #7877

E::Lazy#with_index should be lazy

02/18/2013 08:27 PM - shyouhei (Shyouhei Urabe)

Status:	Closed		
Priority:	Normal		
Assignee:	jeremyevans0 (Jeremy Evans)		
Target version:			
ruby -v:	2.1.0-dev	**Backport:**	2.0.0: UNKNOWN

Description

So I wanted some real benefit of being lazy. I wrote a Leibniz formula:

```
def leibniz(n)
  (0..Float::INFINITY).lazy.with_index {|i, j| (-1 ** j) / (2*i+1).to_f }.take(n).reduce(:+)
end
```

But it doesn't work (well, it does, indeed. It just doesn't stop working). I got frustrated.
How about it? Don't you feel it nifty?

Of course I can wait for the release next to 2.0.0.

Associated revisions

Revision 83498854eb5a824f1f83c31fac18c9279f9ee10d - 09/02/2019 05:20 AM - jeremyevans (Jeremy Evans)

Make Enumerator::Lazy#with_index be lazy

Previously, Enumerator::Lazy#with_index was not defined, so it
picked up the default implementation from Enumerator, which was
not lazy.

Based on earlier patch from nobu.

Fixes [Bug #7877]

Revision 83498854eb5a824f1f83c31fac18c9279f9ee10d - 09/02/2019 05:20 AM - jeremyevans (Jeremy Evans)

Make Enumerator::Lazy#with_index be lazy

Previously, Enumerator::Lazy#with_index was not defined, so it
picked up the default implementation from Enumerator, which was
not lazy.

Based on earlier patch from nobu.

Fixes [Bug #7877]

Revision 83498854 - 09/02/2019 05:20 AM - jeremyevans (Jeremy Evans)

Make Enumerator::Lazy#with_index be lazy

Previously, Enumerator::Lazy#with_index was not defined, so it
picked up the default implementation from Enumerator, which was
not lazy.

Based on earlier patch from nobu.

Fixes [Bug #7877]

Revision e94ac03eb0d07af3cbff20194acf479bf8851c39 - 09/03/2019 06:30 PM - jeremyevans (Jeremy Evans)

Make Enumerator::Lazy#with_index be lazy

Previously, Enumerator::Lazy#with_index was not defined, so it
picked up the default implementation from Enumerator, which was
not lazy.

Based on earlier patch from nobu.

Fixes [Bug #7877]

Revision e94ac03eb0d07af3cbff20194acf479bf8851c39 - 09/03/2019 06:30 PM - jeremyevans (Jeremy Evans)

Make Enumerator::Lazy#with_index be lazy

Previously, Enumerator::Lazy#with_index was not defined, so it
picked up the default implementation from Enumerator, which was
not lazy.

Based on earlier patch from nobu.

Fixes [Bug #7877]

Revision e94ac03e - 09/03/2019 06:30 PM - jeremyevans (Jeremy Evans)

Make Enumerator::Lazy#with_index be lazy

Previously, Enumerator::Lazy#with_index was not defined, so it
picked up the default implementation from Enumerator, which was
not lazy.

Based on earlier patch from nobu.

Fixes [Bug #7877]

History

#1 - 02/19/2013 03:25 PM - nobu (Nobuyoshi Nakada)

- File 0001-enumerator.c-Enumerator-Lazy-with_index.patch added

- Description updated

#2 - 02/19/2013 04:16 PM - marcandre (Marc-Andre Lafortune)

See #7696 on how to handle state...

#3 - 02/19/2013 08:43 PM - shyouhei (Shyouhei Urabe)

- Description updated

OK, so @marcandre (Marc-Andre Lafortune) is interested in. I re-wrote the description in English.

#4 - 02/20/2013 01:30 AM - marcandre (Marc-Andre Lafortune)

Note that (thanks to #7715), you can use with_index without a block and follow it with map:

```
def leibniz(n)
  (0..Float::INFINITY).lazy.with_index.map {|i, j| (-1 ** j) / (2*i+1).to_f }.take(n).reduce(:+)
end
```

I'm neutral about this feature request. The problem I see is that it's too late for 2.0.0 and it would introduce potential incompatibility in next minor.

#5 - 02/20/2013 03:00 PM - shyouhei (Shyouhei Urabe)

@marcandre (Marc-Andre Lafortune) oh, thank you! You saved my day.

Still I want this though.

#6 - 04/05/2013 11:32 AM - zzak (zzak _)

Propose to move this to next major?

#7 - 04/05/2013 01:21 PM - duerst (Martin Dürst)

zzak (Zachary Scott) wrote:

> Propose to move this to next major?

Do you mean because of "potential incompatibility" (https://bugs.ruby-lang.org/issues/7877#note-4)? What exactly would this incompatibility be? To me, it seems that lazy.with_index would just work, so we should just fix it. Next major seems way too long to wait.

#8 - 06/02/2013 02:50 PM - zzak (zzak _)

@duerst (Martin Dürst) You're probably right, since this feature was introduced in 2.0.0

If yhara-san wants to implement #with_index with a block then I see no problem with introducing this in 2.1.0

#9 - 06/02/2013 04:31 PM - zzak (zzak _)

- Tracker changed from Feature to Bug

- Subject changed from E::Lazy#with_index needed to E::Lazy#with_index should be lazy

- Target version changed from 2.6 to 2.1.0

- ruby -v set to 2.1.0-dev

- Backport set to 2.0.0: UNKNOWN

#10 - 01/30/2014 06:16 AM - hsbt (Hiroshi SHIBATA)

- Target version changed from 2.1.0 to 2.2.0

#11 - 07/16/2014 02:03 PM - Nakilon (Victor Maslov)

Is it somehow related? http://stackoverflow.com/q/24782712/322020
... .lazy ... .take_while.with_index{ ...
ArgumentError - tried to call lazy take_while without a block:`

Nevermind, swaping the chain in this way .with_index.take_while saved me.

#12 - 08/08/2016 01:59 AM - shyouhei (Shyouhei Urabe)

Is there reason this is not loved? I'd like to have this.

#13 - 10/11/2016 07:18 AM - shyouhei (Shyouhei Urabe)

- Assignee changed from yhara (Yutaka HARA) to nobu (Nobuyoshi Nakada)

#14 - 08/08/2019 08:45 PM - jeremyevans0 (Jeremy Evans)

- File lazy-with-index-block-7877.patch added

This bug is still present in the master branch. I've updated nobu's patch to apply to the master branch, which required a significant rewrite. The updated patch is attached.

#15 - 08/09/2019 08:07 AM - nobu (Nobuyoshi Nakada)

- Description updated

- Assignee changed from nobu (Nobuyoshi Nakada) to jeremyevans0 (Jeremy Evans)

Thank you, I've missed it.

#16 - 09/04/2019 04:08 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

Files

0001-enumerator.c-Enumerator-Lazy-with_index.patch	3.33 KB	02/19/2013	nobu (Nobuyoshi Nakada)
lazy-with-index-block-7877.patch	4.08 KB	08/08/2019	jeremyevans0 (Jeremy Evans)