Ruby - Feature #10769

Negative counterpart to Enumerable#slice_when

01/22/2015 12:35 AM - sawa (Tsuyoshi Sawada)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	

Description

It seems to me that most useful cases of Enumerable#slice_when involve a negative condition inside the block. That observation seems to be confirmed by the official examples in http://docs.ruby-lang.org/ja/2.2.0/method/Enumerable/i/slice_when.html. In these examples, the conditions inside the block are negations of what would is intended (which is expressed in the comment above each code).

```
# 1づつ増加する部分配列に分ける
[1,2,4,9,10,11,12,15,16,19,20,21]
.slice_when{|i, j| i + 1 != j}.to_a    # => [[1, 2], [4], [9, 10, 11, 12], [15, 16], [19, 20, 21]]

#  ( 6 )
[3, 11, 14, 25, 28, 29, 29, 41, 55, 57]
.slice_when{|i, j| 6 < j - i}.to_a    # => [[3], [11, 14], [25, 28, 29, 29], [41], [55, 57]]

# 
[0, 9, 2, 2, 3, 2, 7, 5, 9, 5]
.slice_when{|i, j| i > j}.to_a    # => [[0, 9], [2, 2, 3], [2, 7], [5, 9], [5]]

# 
[7, 5, 9, 2, 0, 7, 9, 4, 2, 0]
.slice_when{|i, j| i.even? != j.even?}.to_a    # => [[7, 5, 9], [2, 0], [7, 9], [4, 2, 0]]
```

I propose that there should be a method on Enumerable that works like slice_when except that it works with the block negatively as compared to slice_when. Let us call this method Enumerable#chunk_while. Then, the examples above would be written more naturally without having a negative notion in the blocks.

```
# 1づつ増加する部分配列に分ける
[1,2,4,9,10,11,12,15,16,19,20,21]
.chunk_while{|i, j| i + 1 == j}.to_a

#  ( 6 )
[3, 11, 14, 25, 28, 29, 29, 41, 55, 57]
.chunk_while{|i, j| j - i <= 6}.to_a

# 
[0, 9, 2, 2, 3, 2, 7, 5, 9, 5]
.chunk_while{|i, j| i <= j}.to_a

# 
[7, 5, 9, 2, 0, 7, 9, 4, 2, 0]
.chunk_while{|i, j| i.even? == j.even?}.to_a
```

I am not sure about the method name. There can be a better name.

Associated revisions

Revision 11aa7c9604055b252c5d84c6b8cb5e95be7bfafa - 06/14/2015 08:54 AM - akr (Akira Tanaka)

- enum.c (enum_chunk_while): New method Enumerable#chunk_while.
  [ruby-core:67738] [Feature #10769] proposed by Tsuyoshi Sawada.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50889 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 11aa7c96 - 06/14/2015 08:54 AM - akr (Akira Tanaka)

- enum.c (enum_chunk_while): New method Enumerable#chunk_while.
  [ruby-core:67738] [Feature #10769] proposed by Tsuyoshi Sawada.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50889 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 01/25/2015 09:24 PM - austin (Austin Ziegler)

What about .slice_unless?

#2 - 01/25/2015 10:22 PM - sawa (Tsuyoshi Sawada)

Austin Ziegler wrote:

> What about .slice_unless?

The point of my proposal is that this negation is a negation if you look from the point of view of slice_when, but it is in fact more natural than slice_when, so that it should not be considered as negation of the latter (Rather, slice_when should be considered the negation of this method.). But "unless" carries the meaning of negation.

#3 - 06/12/2015 08:22 AM - matz (Yukihiro Matsumoto)

#chunk_while looks good to me. Go ahead.

Matz.

#4 - 06/14/2015 08:54 AM - akr (Akira Tanaka)

- Status changed from Open to Closed

Applied in changeset r50889.

- enum.c (enum_chunk_while): New method Enumerable#chunk_while.
  [ruby-core:67738] [Feature #10769] proposed by Tsuyoshi Sawada.