Ruby trunk - Feature #12217

Introducing Enumerable#sum for precision compensated summation and revert r54237

03/25/2016 11:41 AM - mrkn (Kenta Murata)

Status:	Closed	
Priority:	Normal	
Assignee:	mrkn (Kenta Murata)	
Target version:		

Description

In this issue I propose to introduce Enumerable#sum for precision compensated summation of an array of floating point numbers.

In r54237, I've changed Enumerable#inject to support precision compensated summation for summing up floating point numbers.
But I noticed that this commit broke the equality of ary.inject(:+) == ary.inject {|a, x| a + x }.
I think this equality is important property of inject method, so I don't want to break it.

And because precision compensated algorithms are complicated, I think they are provided in the standard library, especially simple summation.

Related issues:

Related to Ruby trunk - Feature #10298: Array#float_sum (like math.fsum of Py...	**Rejected**
Related to Ruby trunk - Feature #12222: Introducing basic statistics methods ...	**Closed**

Associated revisions

Revision 30d7fb37 - 04/13/2016 02:40 PM - akr (Akira Tanaka)

add a space in [ruby-core:74569] [Feature #12217]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@54566 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 54566 - 04/13/2016 02:40 PM - akr (Akira Tanaka)

add a space in [ruby-core:74569] [Feature #12217]

Revision 54566 - 04/13/2016 02:40 PM - akr (Akira Tanaka)

add a space in [ruby-core:74569] [Feature #12217]

Revision 54566 - 04/13/2016 02:40 PM - akr (Akira Tanaka)

add a space in [ruby-core:74569] [Feature #12217]

Revision 54566 - 04/13/2016 02:40 PM - akr (Akira Tanaka)

add a space in [ruby-core:74569] [Feature #12217]

History

#1 - 03/25/2016 01:21 PM - akr (Akira Tanaka)

- File inject-plus.txt added

I think sum of elements of an enumerable is used in many situations.

I searched '.inject[({]' and '+' in .rb files in gems:
inject-plus.txt

It seems .inject(:+) is an idiom for summation.

```
% fgrep '.inject(:+)' inject-plus.txt|wc -l
1040
% fgrep '.inject(&:+)' inject-plus.txt|wc -l
446
% fgrep '.inject(0, :+)' inject-plus.txt|wc -l
138
% fgrep '.inject(0, &:+)' inject-plus.txt|wc -l
90
% fgrep '.inject(0.0, :+)' inject-plus.txt|wc -l
```

```
13
% fgrep '.inject(0.0, &:+)' inject-plus.txt|wc -l
23
```

It seems .inject {|s,e| s + e} is also used.

```
% grep '.inject(0) *{|\([a-z0-9]\+\), *\([a-z0-9]\+\)| *\1 *+ *\2 *}' inject-plus.txt|wc -l
311
% grep '.inject *{|\([a-z0-9]\+\), *\([a-z0-9]\+\)| *\1 *+ *\2 *}' inject-plus.txt|wc -l
213
% grep '.inject(0.0) *{|\([a-z0-9]\+\), *\([a-z0-9]\+\)| *\1 *+ *\2 *}' inject-plus.txt|wc -l
27
```

I think Enumerable#sum is useful:

- Enumerable#sum is shorter than Enumerable#inject(:+) ant it makes code succinct.
- it can return 0 for empty enumerable. It makes us to avoid bugs on empty enumerable.
- more accurate for sum of Floats by Kahan's compensated summation algorithm (as muraken said)
- faster than Enumerable#inject(:+) (Enumerable#inject(:+) will also be faster since Ruby 2.4, though)

However it has a problem:

- Enumerable#sum is not well defined for non-numeric elements, especially objects without + method.

#2 - 03/25/2016 01:21 PM - akr (Akira Tanaka)

- Related to Feature #10298: Array#float_sum (like math.fsum of Python) added

#3 - 03/25/2016 03:05 PM - akr (Akira Tanaka)

Akira Tanaka wrote:

> - Enumerable#sum is not well defined for non-numeric elements, especially objects without + method.

An idea to avoid this problem is an argument to specify the initial object.

```
module Enumerable
  def sum(init=0)
    inject(init, :+) # or better algorithm.
  end
end
```

Programmers have responsibility that init has + method.

However init has the default value 0 because #sum is mostly used for numeric.

#4 - 03/27/2016 01:07 PM - mrkn (Kenta Murata)

- Related to Feature #12222: Introducing basic statistics methods for Enumerable (and optimized implementation for Array) added

#5 - 04/13/2016 07:03 AM - akr (Akira Tanaka)

Array#sum is accepted by matz at
https://bugs.ruby-lang.org/issues/12222#note-6

#6 - 04/13/2016 01:41 PM - mrkn (Kenta Murata)

- Assignee changed from matz (Yukihiro Matsumoto) to mrkn (Kenta Murata)

#7 - 04/13/2016 02:40 PM - akr (Akira Tanaka)

- Status changed from Assigned to Closed

Applied in changeset r54566.

---

add a space in [ruby-core:74569] [Feature #12217]

#8 - 05/17/2016 10:02 AM - akr (Akira Tanaka)

matz accepted Enumerable#sum (and Range#sum).

Examples:
Range#sum for mathematical sigma: (0..n).sum {|i| ... }

Hash#sum for expected value such as { value1 => probability1, ... }.sum {|v, prob| v * prob }

#9 - 07/07/2016 07:59 PM - schneems (Richard Schneeman)

Would be nice if we could match behavior with Rails Enumerable#sum
https://github.com/rails/rails/blob/3d716b9e66e334c113c98fb3fc4bcf8a945b93a1/activesupport/lib/active_support/core_ext/enumerable.rb#L2-L27

#10 - 07/08/2016 01:25 AM - akr (Akira Tanaka)

Richard Schneeman wrote:

> Would be nice if we could match behavior with Rails Enumerable#sum
> https://github.com/rails/rails/blob/3d716b9e66e334c113c98fb3fc4bcf8a945b93a1/activesupport/lib/active_support/core_ext/enumerable.rb#L2-L27

I don't like it because the behavior is too complex.

For example, [false].sum returns 0.

It is difficult to document (and understand) the behavior.

#11 - 07/08/2016 01:36 AM - mrkn (Kenta Murata)

> For example, [false].sum returns 0.

I reported this undocumented behavior as a bug several months ago.
https://github.com/rails/rails/issues/24796

Files

inject-plus.txt	1.21 MB	03/25/2016	akr (Akira Tanaka)