

Ruby 1.8 - Bug #3938

Ruby incorrectly compares the length of Array elements of Enumerable objects to the arity of Methods given as blocks to Enumerable methods

10/13/2010 03:19 AM - Ethan (Ethan -)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	
ruby -v:	ruby 1.8.6 (2009-06-08 patchlevel 369) [universal-darwin9.0] (and others, see description)

Description

=begin

Given an Enumerable object that has an element which is an Array, when an Enumerable method which takes a block (#map, #detect, etc) is called on that object and given a Method as its block, Ruby will incorrectly compare the length of the Array to the arity of the Method.

```
def Object.onearg(arg)
  arg
end
amethod = Object.method(:onearg)
aproc = proc{|arg| arg}
```

amethod and aproc both have arity of 1. they should generally behave the same. but, they behave differently when given as a block to Enumerable methods, with amethod behaving incorrectly when the Enumerable in question contains an array:

```
1, 2.detect(&amethod)
ArgumentError: wrong number of arguments (2 for 1)
```

this seems to incorrectly compare the length of the element [1, 2] to the arity of amethod, even though it's passing one argument (the array [1, 2]) to amethod.

the Proc behaves correctly:

```
1, 2.detect(&aproc)
=> [1, 2]
```

this does not compare the element's length to the arity of aproc, and so works fine.

Giving the Method as a block works fine when the arity happens to be the same as the length of the element which is an array:

```
1.detect(&amethod)
=> [1]
```

Even though it is passed the whole array (seen in the return value), and not the element of the array.

File is attached to minimally reproduce, and its output is:

```
[1, 2]
[1]
methodarity.rb:10:in onearg': wrong number of arguments (2 for 1) (ArgumentError)
from methodarity.rb:10:in into_proc'
from methodarity.rb:7:in detect'
from methodarity.rb:10:in each'
from methodarity.rb:10:in `detect'
from methodarity.rb:10
```

Tested on:

ruby 1.8.6 (2009-06-08 patchlevel 369) [universal-darwin9.0]
ruby 1.8.6 (2010-02-05 patchlevel 399) [i686-darwin9.8.0]
ruby 1.8.7 (2009-06-12 patchlevel 174) [i686-darwin9.8.0]
ruby 1.8.7 (2010-06-23 patchlevel 299) [i686-darwin9.8.0]
ruby 1.8.6 (2007-09-24 patchlevel 111) [i386-mswin32]
ruby 1.8.6 (2010-02-04 patchlevel 398) [i386-mingw32]
ruby 1.8.7 (2010-01-10 patchlevel 249) [i386-mingw32]

Behaves correctly in ruby 1.9.*.

=end

History

#1 - 10/13/2010 10:58 AM - shyouhei (Shyouhei Urabe)

- *Status changed from Open to Feedback*

=begin

Designed behaviour. This is how an assignment works for 1.8.

But yes, it turned out to be odd. So we changed it to be more natural in 1.9. So your correct way is to switch to 1.9. If you need this backported to 1.8, tell us why. Beware that we are not for that because of backward compatibilities.

=end

Files

methodarity.rb

347 Bytes

10/13/2010

Ethan (Ethan -)