Ruby trunk - Bug #5747

Proc#curry doesn't always detect too many arguments

12/12/2011 09:30 AM - marcandre (Marc-Andre Lafortune)

Status:	Closed		
Priority:	Normal		
Assignee:	marcandre (Marc-Andre Lafortune)		
Target version:	2.0.0		
ruby -v:	-	**Backport:**	

Description

Currently, Proc#curry checks the validity of the arity argument and will raise an error if:

1) arity is less than the number of required arguments or
2) arity is more than the maximum number of arguments
Note that simple Procs always accept any number of arguments (even though they may be ignored), so (2) doesn't applies to them, only to lambda's and procs made from methods.

#2 isn't done as well as it could in case of limited optional parameters:

```
def Object.foo(a, b=42); end
def Object.bar(a, b);    end

Object.method(:foo).to_proc.curry(3) # => curried proc which will raise an error only when passed it's 3rd parameter
Object.method(:bar).to_proc.curry(3) # => ArgumentError: wrong number of arguments (3 for 2)

Same thing with lambdas
```

My proposed fix passes SST:
a) usefulness: "fail-early" principle [ruby-core:24130]
b) consistency: both methods can only accept a maximum of 2 parameters and are thus treated consistently when attempting to curry with more than 2 arguments
c) intuitiveness and d) performance: similar

It is straightforward (but not obvious), as it passes NIT (but not ODT).

Related issues:

Blocked by Ruby trunk - Bug #6085: Treatment of Wrong Number of Arguments	**Closed**	**02/25/2012**

Associated revisions

Revision f810d180 - 02/01/2013 10:46 PM - marcandre (Marc-Andre Lafortune)

- proc.c (proc_curry): Fix arity check [Bug #5747]
- test/ruby/test_proc.rb: Test for above

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39008 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 39008 - 02/01/2013 10:46 PM - marcandre (Marc-Andre Lafortune)

- proc.c (proc_curry): Fix arity check [Bug #5747]
- test/ruby/test_proc.rb: Test for above

Revision 39008 - 02/01/2013 10:46 PM - marcandre (Marc-Andre Lafortune)

- proc.c (proc_curry): Fix arity check [Bug #5747]
- test/ruby/test_proc.rb: Test for above

Revision 39008 - 02/01/2013 10:46 PM - marcandre (Marc-Andre Lafortune)

- proc.c (proc_curry): Fix arity check [Bug #5747]
- test/ruby/test_proc.rb: Test for above

Revision 39008 - 02/01/2013 10:46 PM - marcandre (Marc-Andre Lafortune)

- proc.c (proc_curry): Fix arity check [Bug #5747]
- test/ruby/test_proc.rb: Test for above

Revision 39008 - 02/01/2013 10:46 PM - marcandre (Marc-Andre Lafortune)

- proc.c (proc_curry): Fix arity check [Bug #5747]
- test/ruby/test_proc.rb: Test for above

Revision 39008 - 02/01/2013 10:46 PM - marcandre (Marc-Andre Lafortune)

- proc.c (proc_curry): Fix arity check [Bug #5747]
- test/ruby/test_proc.rb: Test for above

History

#1 - 02/25/2012 01:45 PM - ko1 (Koichi Sasada)

- Target version set to 2.0.0

Any progress on it?

#2 - 02/25/2012 04:23 PM - marcandre (Marc-Andre Lafortune)

Hi,

Koichi Sasada wrote:

> Any progress on it?

I'm having some problems running all tests, but here is what I've done so far.

The first patches defines various min_max_arity functions to get min and max number of (non-ignored) arguments of Proc/Lambdas/Methods. It would make the addition of Ruby methods arity_max or arity_range easy.

Note that to know the maximum arity for most builtin methods would require an extensive refactoring.

The second patch fixes Proc#curry (again, except for most builtin methods). This uses both the first patch and the patches of issue #6089.

https://github.com/marcandre/ruby/commits/proc_curry

#3 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#4 - 02/02/2013 07:46 AM - marcandre (Marc-Andre Lafortune)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r39008.
Marc-Andre, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- proc.c (proc_curry): Fix arity check [Bug #5747]
- test/ruby/test_proc.rb: Test for above