[*x*] (array consisting only of a splat) does not necessarily return a new array

02/07/2009 08:48 AM - mernen (Daniel Luz)

Status: Closed
Priority: Normal
Assignee: nobu (Nobuyoshi Nakada)
Target version: 2.0.0

Description

=end

For [*x*], these are basically the possible outcomes:

1. if x is an Array, returns it unmodified.
2. elsif x responds to to_ary (to_a on 1.9.1), invokes that method and returns its result unmodified.
3. else, returns a new array with x as its only element.

Cases #1 and #2 IMO violate the POLS, as I expected that an array literal would always return a new array. The practical consequence here is that I expected I'd be free to modify it without side effects. (For comparison, "#{x}" always returns a new string)

Simple test case:

```ruby
x = [1, 2, 3]
[*x*] << 4
p x # => [1, 2, 3, 4]
```

Thus, I propose ensuring these two cases always return new arrays.

One possible solution would be dup'ing the resulting array (I guess that'd have a rather low cost; the third case would result in an unnecessary dup, but at least it's just a single-item array). Another one would be to dumb down the interpreter, making it create a zero-length array and then concat the result of the splat to it.

=end

Related issues:

Has duplicate Backport193 - Backport #5124: foo = [*bar*] implies foo.equal?(bar) Closed 07/31/2011

Associated revisions

Revision fde3c421 - 02/15/2012 08:41 PM - nobu (Nobuyoshi Nakada)

- insns.def (splatarray): make new array if flag is set.
- compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34633 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34633 - 02/15/2012 08:41 PM - nobu (Nobuyoshi Nakada)

- insns.def (splatarray): make new array if flag is set.
- compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]

Revision 34633 - 02/15/2012 08:41 PM - nobu (Nobuyoshi Nakada)

- insns.def (splatarray): make new array if flag is set.
- compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]

Revision 34633 - 02/15/2012 08:41 PM - nobu (Nobuyoshi Nakada)

- insns.def (splatarray): make new array if flag is set.
- compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]

Revision 34633 - 02/15/2012 08:41 PM - nobu (Nobuyoshi Nakada)

- insns.def (splatarray): make new array if flag is set.
- compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]
Revision 34633 - 02/15/2012 08:41 PM - nobu (Nobuyoshi Nakada)

- insns.def (splatarray): make new array if flag is set.
- compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]

Revision 34633 - 02/15/2012 08:41 PM - nobu (Nobuyoshi Nakada)

- insns.def (splatarray): make new array if flag is set.
- compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]

Revision f52f0339 - 06/26/2012 11:32 PM - naruse (Yui NARUSE)
merge revision(s) 34633: [Backport #5124]

* insns.def (splatarray): make new array if flag is set.
* compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@36228 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 09/26/2009 05:37 AM - marcandre (Marc-Andre Lafortune)
- Category set to core
- Status changed from Open to Assigned
- Assignee set to matz (Yukihiro Matsumoto)

=begin
=end

#2 - 05/30/2010 06:35 PM - mame (Yusuke Endoh)

=begin
Hi,

I expected that an array literal would always return a new array.

Agreed, but "bug" is too strong a word for this.
I move this ticket to 1.9.x feature.

--
Yusuke Endoh mame@tsg.ne.jp
=end

#3 - 05/30/2010 06:36 PM - mame (Yusuke Endoh)

- Target version set to 2.0.0

=begin
=end

#4 - 02/13/2012 11:27 PM - mame (Yusuke Endoh)
- Assignee changed from matz (Yukihiro Matsumoto) to mame (Yusuke Endoh)

Hello,

I'll commit the following patch unless there is objection:

diff --git a/compile.c b/compile.c
index 32bda52..9ea2a3c 100644
--- a/compile.c
+++ b/compile.c
@@ -4644,6 +4644,8 @@ iseq_compile_each(rb_iseq_t *iseq, LINK_ANCHOR *ret, NODE * node, int poped)
case NODE_SPLAT:
    COMPILE(ret, "splat", node->nd_head);
I'll commit the following patch unless there is objection:

Isn't the flag of splatarray for this purpose?

```diff
diff --git i/compile.c w/compile.c
index 32bda52..c8aced3 100644
--- i/compile.c
+++ w/compile.c
@@ -4643,7 +4643,7 @@ iseq_compile_each(rb_iseq_t *iseq, LINK_ANCHOR *ret, NODE * node, int poped)
                 }
 case NODE_SPLAT:
  COMPILER(ret, "splat", node->nd_head);
-     ADD_INSN1(ret, nd_line(node), splatarray, Qfalse);
+     ADD_INSN1(ret, nd_line(node), splatarray, Qtrue);

 if (poped) {  ADD_INSN(ret, nd_line(node), pop);
```
Hello,

2012/2/14 Nobuyoshi Nakada nobu@ruby-lang.org:

Isn't the flag of splatarray for this purpose?

I don't know. Please go ahead. You have control!

--

Yusuke Endoh mame@tsg.ne.jp

#7 - 02/16/2012 05:41 AM - nobu (Nobuyoshi Nakada)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r34633.
Daniel, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- insns.def (splatarray): make new array if flag is set.
- compile.c (iseq_compile_each): make new array with splat. [ruby-core:21901][Feature #1125]