

Ruby - Bug #21031

Incompatibility with prism and parse.y when eval'ing unnamed forwarding variables

01/13/2025 10:54 AM - ksss (Yuki Kurihara)

Status:	Closed	
Priority:	Normal	
Assignee:	prism	
Target version:		
ruby -v:	ruby 3.5.0dev (2025-01-13T03:12:28Z master 384e6945ac) +PRISM [arm64-darwin23]	Backport: 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: DONE

Description

Case 1

```
# t.rb
def foo(*)
  eval("p(*)")
end

foo(1)

$ ruby --parser=prism t.rb
t.rb:4:in 'Kernel#eval': (eval at t.rb:4):1: syntax error found (SyntaxError)
> 1 | p(*)
    |   ^ unexpected `*`; no anonymous rest parameter

from t.rb:4:in 'Object#foo'
from t.rb:7:in '<main>'

$ ruby --parser=parse.y t.rb
1
```

Case 2

```
# t.rb
def foo(**)
  eval("p(**)")
end

foo(a: 1)

$ ruby --parser=prism t.rb
t.rb:4:in 'Kernel#eval': (eval at t.rb:4):1: syntax error found (SyntaxError)
> 1 | p(**)
    |   ^~ unexpected `**`; no anonymous keyword rest parameter

from t.rb:4:in 'Object#foo'
from t.rb:7:in '<main>'

$ ruby --parser=parse.y t.rb
{a: 1}
```

Case 3

```
# t.rb
def foo(&)
  eval("p(&)")
end

foo{}
```

```
$ ruby --parser=prism t.rb
t.rb:4:in 'Kernel#eval': (eval at t.rb:4):1: syntax error found (SyntaxError)
> 1 | p(&)
    |   ^ unexpected `&`; no anonymous block parameter

from t.rb:4:in 'Object#foo'
from t.rb:7:in '<main>'

$ ruby --parser=parse.y t.rb
```

Case 4

```
# t.rb
def foo(...)
  eval("p(...)")
end

foo(1, 2, 3)

$ ruby --parser=prism t.rb
t.rb:4:in 'Kernel#eval': (eval at t.rb:4):1: syntax error found (SyntaxError)
> 1 | p(...)
    |   ^~~ unexpected ... when the parent method is not forwarding

from t.rb:4:in 'Object#foo'
from t.rb:7:in '<main>'

$ ruby --parser=parse.y t.rb
1
2
3
```

I expect to the parse.y behavior.

Associated revisions

Revision [cb419e3912f0514b8151469b0a4a4b83cbbcce78](#) - 01/14/2025 11:41 PM - kddnewton (Kevin Newton)

[PRISM] Handle forwarding inside eval

Fixes [Bug #21031]

Revision [cb419e3912f0514b8151469b0a4a4b83cbbcce78](#) - 01/14/2025 11:41 PM - kddnewton (Kevin Newton)

[PRISM] Handle forwarding inside eval

Fixes [Bug #21031]

Revision [cb419e39](#) - 01/14/2025 11:41 PM - kddnewton (Kevin Newton)

[PRISM] Handle forwarding inside eval

Fixes [Bug #21031]

Revision [7adf89d7ad30552d7e57709d24eec266f601d38b](#) - 02/14/2025 05:06 AM - k0kubun (Takashi Kokubun)

merge revision(s) [cb419e3912f0514b8151469b0a4a4b83cbbcce78](#): [Backport #21031]

[PRISM] Handle forwarding inside eval

Fixes [Bug #21031]

Revision [7adf89d7ad30552d7e57709d24eec266f601d38b](#) - 02/14/2025 05:06 AM - k0kubun (Takashi Kokubun)

merge revision(s) [cb419e3912f0514b8151469b0a4a4b83cbbcce78](#): [Backport #21031]

[PRISM] Handle forwarding inside eval

Fixes [Bug #21031]

Revision 7adf89d7 - 02/14/2025 05:06 AM - k0kubun (Takashi Kokubun)

merge revision(s) cb419e3912f0514b8151469b0a4a4b83cbbcce78: [Backport #21031]

[PRISM] Handle forwarding inside eval

Fixes [Bug #21031]

Revision de7a196bd4b22f6073e6cf6c2d7d6fd8171b6baa - 02/14/2025 05:20 AM - k0kubun (Takashi Kokubun)

Revert "merge revision(s) cb419e3912f0514b8151469b0a4a4b83cbbcce78: [Backport #21031]"

This reverts commit 7adf89d7ad30552d7e57709d24eec266f601d38b.

Revision de7a196bd4b22f6073e6cf6c2d7d6fd8171b6baa - 02/14/2025 05:20 AM - k0kubun (Takashi Kokubun)

Revert "merge revision(s) cb419e3912f0514b8151469b0a4a4b83cbbcce78: [Backport #21031]"

This reverts commit 7adf89d7ad30552d7e57709d24eec266f601d38b.

Revision de7a196b - 02/14/2025 05:20 AM - k0kubun (Takashi Kokubun)

Revert "merge revision(s) cb419e3912f0514b8151469b0a4a4b83cbbcce78: [Backport #21031]"

This reverts commit 7adf89d7ad30552d7e57709d24eec266f601d38b.

Revision 45fe3c137b6cc0b2546493e37d6334d8f39e076d - 02/14/2025 06:19 PM - kddnewton (Kevin Newton)

Backport Bug #21031

Revision 45fe3c137b6cc0b2546493e37d6334d8f39e076d - 02/14/2025 06:19 PM - kddnewton (Kevin Newton)

Backport Bug #21031

Revision 45fe3c13 - 02/14/2025 06:19 PM - kddnewton (Kevin Newton)

Backport Bug #21031

History

#1 - 01/13/2025 11:29 AM - Eregon (Benoit Daloze)

[@jeremyevans0 \(Jeremy Evans\)](#) Isn't this parse.y behavior breaking the optimization to forward `*/**/&` efficiently (and maybe ... too)? It means static analysis can't reason about where the `*/**/&/...` are used.

IMO all these should be `SyntaxError`.
I'm not sure the parse.y behavior is intentional.

#2 - 01/13/2025 11:38 AM - Eregon (Benoit Daloze)

As a note, I checked and the `*` and `**` examples (Case 1 & 2) work in 3.2 and 3.3.

Case 3 / `&` seems incorrect though, as it should print the block but doesn't.
EDIT: ah it's because `p` ignores the block and it's passed as a block and not a positional argument, nevermind.

#3 - 01/13/2025 04:37 PM - jeremyevans0 (Jeremy Evans)

Eregon (Benoit Daloze) wrote in [#note-1](#):

[@jeremyevans0 \(Jeremy Evans\)](#) Isn't this parse.y behavior breaking the optimization to forward `*/**/&` efficiently (and maybe ... too)? It means static analysis can't reason about where the `*/**/&/...` are used.

For `*` and `**`, it shouldn't break the allocationless anonymous splat forwarding optimization. The optimization just makes the parameter not allocate an array/hash, regardless of the content of the method, since you can only splat the arguments, not access them directly.

For `&`, there is no optimization for anonymous block arguments. The block argument optimization to avoid Proc allocation was `getblockparamproxy` in Ruby 2.5, and applies to both named and anonymous blocks, and I believe works fine in eval (maybe [@ko1 \(Koichi Sasada\)](#) can confirm).

For `...`, that never used allocationless anonymous splat forwarding in a shipped version of Ruby, only temporarily during the 3.4 development cycle. [@tenderlovmaking \(Aaron Patterson\)](#) applied a separate generic argument forwarding optimization ([cdf33ed5f37f9649c482c3ba1d245f0d80ac01ce](#)). I assume it works inside eval, but [@tenderlovmaking \(Aaron Patterson\)](#) would know better than I.

#4 - 01/13/2025 05:09 PM - Eregon (Benoit Daloze)

- Assignee set to prism

Thanks for clarifying, I thought the allocationless anonymous splat forwarding optimization relied on some specific usage of `/**` but it seems not, all good.

Then given this all works on 3.2 and 3.3 as well this seems a clear bug of the Prism compiler (and the lack of tests/specs for this).

#5 - 01/15/2025 01:59 AM - kddnewton (Kevin Newton)

- Status changed from Open to Closed

Applied in changeset [git|cb419e3912f0514b8151469b0a4a4b83cbbcce78](#).

[PRISM] Handle forwarding inside eval

Fixes [Bug [#21031](#)]

#6 - 01/16/2025 02:50 AM - kddnewton (Kevin Newton)

- Backport changed from 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN to 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: REQUIRED

#7 - 02/14/2025 05:07 AM - k0kubun (Takashi Kokubun)

- Backport changed from 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: REQUIRED to 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: DONE

ruby_3_4 [7adf89d7ad30552d7e57709d24eec266f601d38b](#) merged revision(s) [cb419e3912f0514b8151469b0a4a4b83cbbcce78](#).

#8 - 02/14/2025 05:23 AM - k0kubun (Takashi Kokubun)

- Backport changed from 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: DONE to 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: REQUIRED

Reverted the backport [7adf89d7ad30552d7e57709d24eec266f601d38b](#) for now to fix undeclared identifier/function errors.

#9 - 02/14/2025 06:28 PM - k0kubun (Takashi Kokubun)

- Backport changed from 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: REQUIRED to 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: DONE

ruby_3_4 [45fe3c137b6cc0b2546493e37d6334d8f39e076d](#).