Relating to keyword argument (#5474), there are some requests for caller-side **. [ruby-core:40518]

```ruby
def foo(k1: 1, k2: 2)
p [k1, k2]
end
h = {k2: "bar"}
foo(k1: "foo", **h)              # <== here
#=> ["foo", "bar"]
```

Marc-Andre explained the use case [ruby-core:41772], and matz agreed with this feature. [ruby-core:41818]

However, it conflicts with power expression when parens are omitted:

```ruby
foo h # foo(h)? or foo.send("**", h)?
```

Which should it be interpreted?

Anyway, I have no idea to avoid yacc conflict. Nobu, could you please try to implement it?

--

Yusuke Endoh mame@tsg.ne.jp

Associated revisions

Revision 33809741 - 04/28/2012 09:12 PM - nobu (Nobuyoshi Nakada)

- parse.y (assoc, parser_yylex): add syntax to splat keyword hash. [ruby-core:44591][Feature #6353]
- compile.c (compile_array_): generate keyword splat insns.
- vm.c (m_core_hash_merge_kwd): merge keyword hash into intermediate hash. leftward argument is prior currently.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35489 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 35489 - 04/28/2012 09:12 PM - nobu (Nobuyoshi Nakada)

- parse.y (assoc, parser_yylex): add syntax to splat keyword hash. [ruby-core:44591][Feature #6353]
- compile.c (compile_array_): generate keyword splat insns.
- vm.c (m_core_hash_merge_kwd): merge keyword hash into intermediate hash. leftward argument is prior currently.

Revision 35489 - 04/28/2012 09:12 PM - nobu (Nobuyoshi Nakada)

- parse.y (assoc, parser_yylex): add syntax to splat keyword hash. [ruby-core:44591][Feature #6353]
- compile.c (compile_array_): generate keyword splat insns.
- vm.c (m_core_hash_merge_kwd): merge keyword hash into intermediate hash. leftward argument is prior currently.

Revision 35489 - 04/28/2012 09:12 PM - nobu (Nobuyoshi Nakada)

- parse.y (assoc, parser_yylex): add syntax to splat keyword hash. [ruby-core:44591][Feature #6353]
- compile.c (compile_array_): generate keyword splat insns.
- vm.c (m_core_hash_merge_kwd): merge keyword hash into intermediate hash. leftward argument is prior currently.
Hi,

In message "Re: [ruby-core:44591] [ruby-trunk - Feature #6353] [Assigned] caller-side **" on Tue, 24 Apr 2012 22:04:03 +0900, "mame (Yusuke Endoh)" mame@tsg.ne.jp writes:

|However, it conflicts with power expression when parens are omitted:|
| | foo h # foo(h)? or foo.send("***", h)? |
| | Which should it be interpreted? |

It should be interpreted as *, i.e.

foo a # foo("a")
foo * a # foo(a)
foo*a # foo:*(a)

thus

foo h # foo(h)
foo ** h # foo.(h)
foo h # foo.**(h)

matz.

=begin

Almost implemented, and another question.

What should happen in this case?

def foo(k1: 1)
  p k1
end
h = [k1: "bar"]
foo(k1: "foo", **h) # == conflict

{({k1:1})} will be {({"foo")}, or {({"bar")}? Or an exception should be raised?
=end

This issue was solved with changeset r35489. Yusuke, thank you for reporting this issue. Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- parse.y (assoc, parser_yylex): add syntax to splat keyword hash. [ruby-core:44591][Feature #6353]
- compile.c (compile_array[]): generate keyword splat insns.
- vm.c (m_core_hash_merge_kwd): merge keyword hash into intermediate hash. leftward argument is prior currently.