# Ruby - Bug #9776

## Ruby double-splat operator unexpectedly modifies hash

04/25/2014 08:12 PM - jessesielaff (Jesse Sielaff)

| | | | |
|---|---|---|---|
| **Status:** | Closed | | |
| **Priority:** | Normal | | |
| **Assignee:** | matz (Yukihiro Matsumoto) | | |
| **Target version:** | 2.2.0 | | |
| **ruby -v:** | ruby 2.1.1p76 (2014-02-24 revision 45161) [x86_64-linux] | **Backport:** | 2.0.0: DONTNEED, 2.1: DONE |

**Description**

I noticed what I find to be a very surprising behavior with the double-splat (**) operator in Ruby 2.1.1.

When key-value pairs are used before a **hash, the hash remains unmodified. However, when key-value pairs are only used after the **hash, the hash is permanently modified.

```
h = { b: 2 }

{ a: 1, **h }        # => { a: 1, b: 2 }
h                    # => { b: 2 }

{ a: 1, **h, c: 3 }  # => { a: 1, b: 2, c: 3 }
h                    # => { b: 2 }

{ **h, c: 3 }        # => { b: 2, c: 3 }
h                    # => { b: 2, c: 3 }
```

For comparison, consider the behavior of the splat (*) operator on arrays:

```
a = [2]

[1, *a]     # => [1, 2]
a           # => [2]

[1, *a, 3]  # => [1, 2, 3]
a           # => [2]

[*a, 3]     # => [2, 3]
a           # => [2]
```

The array remains unchanged throughout.

---

Tsuyoshi Sawada has also highlighted that the expression's result is the self-same object as the original hash:

```
h.object_id == { **h, c: 3 }.object_id # => true
```

---

I investigated parse.y to try to determine the error there, but I couldn't narrow it down any further than the list_concat or rb_ary_push function calls in the assocs : block of the grammar.

Without exhaustively examining the C source, I think the best clue to the mechanism behind the erroneous behavior might be the following:

```
h = { a: 1 }
{ **h, a: 99, **h } # => {:a=>99}
```

That we don't see {:a=>1} illustrates that h[:a] is already overwritten by the time the second **h is evaluated.

---

Here is the use case that led me to this discovery:

```
def foo (arg) arg end

h = { a: 1 }

foo(**h, b: 2)

h # => { a: 1, b: 2 }
```

In the above example, I don't want { b: 2 } permanently added to my existing hash. I'm currently solving it like this:

```
h = { a: 1 }

foo(**h.dup, b: 2)

h # => { a: 1 }
```

The call to #dup feels unnecessary, and is inconsistent with the analogous behavior when using the single * operator. If this bug is fixed, I'll be able to eliminate that call.

## Associated revisions

**Revision 37e432b5bf4d873a987738891d86f588e97a53a6 - 04/26/2014 01:55 AM - nobu (Nobuyoshi Nakada)**

compile.c: non-destructive keyword splat

- compile.c (compile_array_): make copy a first hash not to modify
  the argument itself.  keyword splat should be non-destructive.
  [ruby-core:62161] [Bug #9776]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45724 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 37e432b5 - 04/26/2014 01:55 AM - nobu (Nobuyoshi Nakada)**

compile.c: non-destructive keyword splat

- compile.c (compile_array_): make copy a first hash not to modify
  the argument itself.  keyword splat should be non-destructive.
  [ruby-core:62161] [Bug #9776]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45724 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 5a2f6a4af8a4b1082207979e2913c681ad6ea1bb - 06/16/2014 03:59 PM - nagachika (Tomoyuki Chikanaga)**

merge revision(s) r45724: [Backport #9776]

```
    * compile.c (compile_array_): make copy a first hash not to modify
      the argument itself.  keyword splat should be non-destructive.
      [ruby-core:62161] [Bug #9776]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@46451 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 5a2f6a4a - 06/16/2014 03:59 PM - nagachika (Tomoyuki Chikanaga)**

merge revision(s) r45724: [Backport #9776]

```
    * compile.c (compile_array_): make copy a first hash not to modify
      the argument itself.  keyword splat should be non-destructive.
      [ruby-core:62161] [Bug #9776]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@46451 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## History

**#1 - 04/25/2014 08:30 PM - jessesielaff (Jesse Sielaff)**

Origin:

http://stackoverflow.com/questions/23282342/ruby-double-splat-operator-destructive-vs-nondestructive

**#2 - 04/26/2014 01:54 AM - nobu (Nobuyoshi Nakada)**

*- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.0.0: DONTNEED, 2.1: REQUIRED*

**#3 - 04/26/2014 01:55 AM - nobu (Nobuyoshi Nakada)**

*- Status changed from Open to Closed*

*- % Done changed from 0 to 100*

Applied in changeset r45724.

---

compile.c: non-destructive keyword splat

- compile.c (compile_array_): make copy a first hash not to modify
  the argument itself.  keyword splat should be non-destructive.
  [ruby-core:62161] [Bug #9776]

**#4 - 06/16/2014 04:00 PM - nagachika (Tomoyuki Chikanaga)**

*- Backport changed from 2.0.0: DONTNEED, 2.1: REQUIRED to 2.0.0: DONTNEED, 2.1: DONE*

Backported into ruby_2_1 branch at r46451.