Ruby master - Bug #10994
Inconsistent behavior when mixing optional argument and keyword splat

03/21/2015 05:28 PM - ruipserra (Rui Serra)

Status: Closed
Priority: Normal
Assignee: 
Target version: ruby -v: ruby 2.2.1p85 (2015-02-26 revision 49769) [x86_64-darwin13] Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description
The following code produces inconsistent behavior:

```ruby
def foo(x=nil, **kw)
    x
end

h = { "a" => 1 }

foo(h)               # => {"a"=>1}
foo(h).equal?(h)     # => false
foo(h, y: 1).equal?(h) # => true
```

It seems that in this very specific case, `foo` is returning a copy of `h` instead of returning `h` itself. I guess this is an edge case in the keyword extraction process.

Thank you for your time.

Related issues:
Related to Ruby master - Feature #14183: "Real" keyword argument

History
#1 - 07/07/2019 04:23 PM - jeremyevans0 (Jeremy Evans)
- Related to Feature #14183: "Real" keyword argument added

#2 - 09/02/2019 04:31 AM - jeremyevans0 (Jeremy Evans)
- Status changed from Open to Closed

With the acceptance of #14183, you now get:

```ruby
foo(h)
# => {"a"=>1}

foo(h).equal?(h)
# => true

foo(h, y: 1).equal?(h)
# => true
```