The following script:

```ruby
def mutate
  re = /foo/
  state = re.instance_variable_get(:@state)
  re.instance_variable_set(:@state, state.to_i + 1)
  state
end

3.times do
  p mutate
end
```

Output this:

```
nil
1
2
```

IMHO, you shouldn't be able to mutate an unduplicated literal.

GitHub pull request: [https://github.com/ruby/ruby/pull/2705](https://github.com/ruby/ruby/pull/2705)

**Related issues:**

Is duplicate of Ruby master - Feature #8948: Frozen regex

---

**Revision 98ef38ad - 01/15/2020 01:38 AM - byroot (Jean Boussier)**

Freeze Regexp literals

[Feature #8948] [Feature #16377]

Since Regexp literals always reference the same instance, allowing to mutate them can lead to state leak.

**History**

#1 - 11/28/2019 06:37 AM - byroot (Jean Boussier)

Apparently it's a duplicate of [https://bugs.ruby-lang.org/issues/8948](https://bugs.ruby-lang.org/issues/8948)

#2 - 11/28/2019 06:38 AM - shyouhei (Shyouhei Urabe)

- Is duplicate of Feature #8948: Frozen regex added

#3 - 12/11/2019 02:54 PM - Dan0042 (Daniel DeLorme)

I really hope this does not go through.

Regexp literals have been "unduplicated" like this since at least 1.8 and we've never had problems. And now we should freeze them and introduce an incompatibility just for the sake of Communicating the Holy Gospel of Immutability? I don't find that a valid reason. Additionally, Regexp literals are not deduplicated in the same sense as frozen string literals; one /abc/ is independant from another /abc/ so we're not actually leaking "global" state. In the example above the state is local to the method but shared between invocations. Not sure how that should be called, but certainly not "global".
And what if the mutate behavior shown above is actually wanted? Sure it's a hack, but it's a bit like function-static variables in PHP.

Or, more realistically, what about something like this?

```ruby
class Regexp
  def analyze
    @analyze ||= RegexpAnalyzer.analyze_performance_issues(self)
  end
end
```

Freezing objects closes off possibilities when it's done by default, and should only be done when absolutely necessary.

**#4 - 12/12/2019 10:09 AM - byroot (Jean Boussier)**

```
just for the sake of Communicating the Holy Gospel of Immutability?
```

Please don't put ideological thoughts on me, that's not a good basis for debate.

Regexp literals are not deduplicated in the same sense as frozen string literals; one /abc/ is independent from another /abc/

And if they were frozen, they could be.

so we're not actually leaking "global" state

On a particular callsite it does.

```
Or, more realistically, what about something like this?
```

```ruby
class Regexp
  @@analyses = {}.compare_by_identity
  def analyze
    @@analyses[self] ||= RegexpAnalyzer.analyze_performance_issues(self)
  end
end
```

**#5 - 12/13/2019 02:43 AM - Dan0042 (Daniel DeLorme)**

```
Please don't put ideological thoughts on me, that's not a good basis for debate.
```

Oh no, I'm not. Sorry for the misunderstanding. I was mostly replying to comments in #8948, but I ended up posting here because this is the ticket that is linked in #16393 DevelopersMeeting20191220Japan.

Yes, of course when one way is blocked there's always an alternate way of doing things. But the point is that if someone is currently relying on Regexp literals not being frozen, this change will break their code. Given that the benefit is close to zero, I don't think it's responsible to force anyone to bear that cost. I'm very conservative about backward compatibility.

**#6 - 12/25/2019 12:15 PM - mame (Yusuke Endoh)**

```
- Target version set to 36
```

**#7 - 12/26/2019 02:43 AM - mame (Yusuke Endoh)**

```
- Assignee set to mame (Yusuke Endoh)
```

At the previous dev meeting, matz said that let's give it a try :-)

For the record: Regexp.new should continue to return unfrozen Regexp instance.

I'll review the pull request.

**#8 - 01/15/2020 01:38 AM - ko1 (Koichi Sasada)**

Could you add a NEWS entry?
Freeze Regexp literals

[Feature #8948] [Feature #16377]

Since Regexp literals always reference the same instance, allowing to mutate them can lead to state leak.

#10 - 09/29/2020 03:37 AM - hsbt (Hiroshi SHIBATA)
- Target version changed from 36 to 3.0