Ruby master - Bug #15449

Range#=== is not using cover in Ruby 2.6

12/21/2018 01:13 PM - ana06 (Ana Maria Martinez Gomez)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	2.6.0dev (2018-12-20 trunk 66466)	**Backport:**	2.5: DONTNEED, 2.6: REQUIRED

Description

```
irb(main):105:0> ('A'..'Z').cover? 'ANA'
=> true
irb(main):106:0> ('A'..'Z') === 'ANA'
=> false
```

Is this expected? Should === use cover according to NEWS in Ruby 2.6?

```
> ruby -v
ruby 2.6.0dev (2018-12-20 trunk 66466) [x86_64-linux]
```

Associated revisions

Revision 6954ff1d - 08/14/2019 09:14 PM - jeremyevans (Jeremy Evans)

Make Range#=== operate like cover? instead of include? for string ranges

Previously, Range#=== treated string ranges that were not endless or
beginless the same as include?, instead of the same as cover?.
I think this was an oversight in 989e07c0f2fa664a54e52a475c2fcc145f06539d,
as the commit message did not indicate this behavior was desired.

This also makes some previously dead code no longer dead. Previously,
the conditionals were doing this:

```
if (RB_TYPE_P(beg, T_STRING)
    if (NIL_P(beg)) # can never be true
```

This restructures it so at the NIL_P(beg) check, beg could possibly
be nil (beginless ranges).

Fixes [Bug #15449]

History

#1 - 12/21/2018 01:27 PM - zverok (Victor Shepelev)

According to code, it first tries to use range_include_internal, which has special handling for strings.

Justification is not obvious, but probably it is an attempt to preserve backwards compatibility?

#2 - 12/21/2018 01:35 PM - ana06 (Ana Maria Martinez Gomez)

If that the case, this should be clarified in NEWS, because it sounds as it was changed for all cases...

#3 - 12/21/2018 01:55 PM - Hanmac (Hans Mackowiak)

That isn't new, that is since 1.9 as #cover? was added

#4 - 12/21/2018 01:58 PM - osyo (manga osyo)

Sorry, Range#=== is called Range#cover? in Ruby 2.6.
see: https://bugs.ruby-lang.org/issues/14575

#5 - 07/29/2019 11:36 PM - jeremyevans0 (Jeremy Evans)

- Backport changed from 2.4: UNKNOWN, 2.5: UNKNOWN to 2.5: DONTNEED, 2.6: REQUIRED

- File range-case-cover-string.patch added

I agree that this is a bug, and not the intended behavior now that Range#=== calls Range#cover? in most cases. Hopefully nobu can confirm. Attached is a patch that fixes the behavior.

I think this should be backported to 2.6, assuming this behavior is not intentional.

#6 - 08/14/2019 11:16 PM - jeremyevans (Jeremy Evans)

- Status changed from Open to Closed

Applied in changeset git|6954ff1dcb538ee6c042872088b64464a1ef6089.

Make Range#=== operate like cover? instead of include? for string ranges

Previously, Range#=== treated string ranges that were not endless or
beginless the same as include?, instead of the same as cover?.
I think this was an oversight in 989e07c0f2fa664a54e52a475c2fcc145f06539d,
as the commit message did not indicate this behavior was desired.

This also makes some previously dead code no longer dead. Previously,
the conditionals were doing this:

```
if (RB_TYPE_P(beg, T_STRING)
    if (NIL_P(beg)) # can never be true
```

This restructures it so at the NIL_P(beg) check, beg could possibly
be nil (beginless ranges).

Fixes [Bug #15449]

Files

range-case-cover-string.patch	3.69 KB	07/29/2019	jeremyevans0 (Jeremy Evans)