

Ruby master - Bug #13663

`String#upto` doesn't work as expected

06/15/2017 07:24 AM - sos4nt (Stefan Schüßler)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.4.1p111 (2017-03-22 revision 58053) [x86_64-darwin15]	Backport: 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN
Description	
Given that String#upto uses String#succ to generate successive values, I'd expect	
<pre>'x'.upto('ac').to_a #=> []</pre>	
to return:	
<pre>["x", "y", "z", "aa", "ab", "ac"]</pre>	
Instead, an empty array is returned.	
This seems to depend on whether the the receiver is greater than the argument or not:	
<pre>'x' <=> 'ac' #=> 1</pre>	
It works just fine in this case:	
<pre>'b'.upto('ca').to_a #=> ["b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", # "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", "aa", # "ab", "ac", "ad", "ae", "af", "ag", "ah", "ai", "aj", "ak", "al", # "am", "an", "ao", "ap", "aq", "ar", "as", "at", "au", "av", "aw", # "ax", "ay", "az", "ba", "bb", "bc", "bd", "be", "bf", "bg", "bh", # "bi", "bj", "bk", "bl", "bm", "bn", "bo", "bp", "bq", "br", "bs", # "bt", "bu", "bv", "bw", "bx", "by", "bz", "ca"]</pre>	
Presumably because of:	
<pre>'b' <=> 'ca' #=> -1</pre>	

History

#1 - 06/15/2017 12:53 PM - zverok (Victor Shepelev)

I believe that problem here is how to provide consistency between succ and <=> for arbitrary length strings.

1. For most of the real use cases, 'x' > 'ac' is sane (like sorting strings);
2. Things using succ (like upto and ranges) should check that begin is lower than end;

So... I believe that it is only reasonable to have (1) and (2), though sometimes it leads to "inconsistencies", like described above. If you do a lot of stuff with making ranges from "x" to "ac" it is probably better to have dedicated value object class, with redefined <=> and succ

#2 - 06/15/2017 01:08 PM - Hanmac (Hans Mackowiak)

Also i think its a bit to much optimised.

because i tried to overwrite the String#<=> method, directly or with Refinements, and it didnt work as i want it to.

what works is using my own Value Object Class.

#3 - 06/23/2017 11:47 AM - sos4nt (Stefan Schüßler)

I'm not asking for consistency between String#succ and String#<=> (although that would be desirable). I do understand that the discrepancy may result in surprising results when generating sequences via Range because Range relies on these methods. Range checks via <=> if the value generated by succ is indeed a successor.

But `String#upto` doesn't have the limitations `Range` has. It doesn't have to work with arbitrary objects and it doesn't have to depend on `<=>`. It can apply whatever logic is needed to ensure a sane result.