Ruby - Bug #6258

String#succ has suprising behavior for "\u1036" (MYANMAR SIGN ANUSVARA), producing "\u1000" instead of "\u1037"

04/05/2012 10:59 AM - dbenhur (Devin Ben-Hur)

Status:	Closed		
Priority:	Normal		
Assignee:	duerst (Martin Dürst)		
Target version:			
ruby -v:	ruby 1.9.3p125, ruby 1.9.2p180,	**Backport:**	

Description

"\u1036".succ.ord.to_s(16) # => "1000"

Discovered when investigating StackOverflow question
http://stackoverflow.com/questions/10020230/anomalous-behavior-while-comparing-a-unicode-character-to-a-unicode-character-range

Range#=== ultimately invokes String#upto which uses String#succ

("\u1036".."\u1037").to_a.map{|c| c.ord.to_s(16)}
=> ["1036"] # expected ["1036","1037"]

Also once #succ! proceeds past U+1036 it continues to produce U+1000 indefinitely

irb(main):115:0> c = "\u1036"
=> "□"
irb(main):116:0> c.ord.to_s(16)
=> "1035"
irb(main):117:0> c.succ!.ord.to_s(16)
=> "1036"
irb(main):118:0> c.succ!.ord.to_s(16)
=> "1000"
irb(main):119:0> c.succ!.ord.to_s(16)
=> "1000"

But if one starts naturally at U+1000 #succ! increments as expected
irb(main):001:0> c = "\u1000"
=> "□"
irb(main):002:0> c.ord.to_s(16)
=> "1000"
irb(main):003:0> c.succ!.ord.to_s(16)
=> "1001"
irb(main):004:0> c.succ!.ord.to_s(16)
=> "1002"

Related issues:

Related to Ruby - Feature #5607: Inconsistent reaction in Range of String	**Closed**

History

#1 - 04/05/2012 01:10 PM - shyouhei (Shyouhei Urabe)

- Category changed from core to M17N

- Status changed from Open to Assigned

- Assignee set to akr (Akira Tanaka)

Sounds like a bug to me, but no idea what's going on. Tanaka-san, what do you think?

#2 - 04/05/2012 01:43 PM - akr (Akira Tanaka)

"\u1036".succ is "\u1000\u1000", not a single character.

% ruby -ve 'puts "\u1036".succ.dump'

ruby 2.0.0dev (2012-03-16 trunk 35049) [x86_64-linux]
"\u{1000}\u{1000}"

It is similar that "z".succ is "aa".

It is because U+1000 to U+1036 are alphabet characters and
U+0fff and U+1037 is not.

% ruby -e '0xfff.upto(0x1037) {|c| p ["%x" % c, /[[:alpha:]]/ =~ c.chr("UTF-8")] }'
["fff", nil]
["1000", 0]
...
["1036", 0]
["1037", nil]

What I'm not sure is U+1036 is alphabet or not.
I think nurse-san or martin-sensei is appropriate for this matter.

#3 - 04/06/2012 11:50 PM - mame (Yusuke Endoh)

- Assignee changed from akr (Akira Tanaka) to duerst (Martin Dürst)

#4 - 09/18/2015 09:18 AM - duerst (Martin Dürst)

- Status changed from Assigned to Feedback

Some information gathered during today's commiters' meeting:
This is the relevant information from http://www.unicode.org/Public/UCD/latest/ucd/UnicodeData.txt:

1035;MYANMAR VOWEL SIGN E ABOVE;Mn;0;NSM;;;;;N;;;;;
1036;MYANMAR SIGN ANUSVARA;Mn;0;NSM;;;;;N;;;;;
1037;MYANMAR SIGN DOT BELOW;Mn;7;NSM;;;;;N;;;;;
1038;MYANMAR SIGN VISARGA;Mc;0;L;;;;;N;;;;;
1039;MYANMAR SIGN VIRAMA;Mn;9;NSM;;;;;N;;;;;
103A;MYANMAR SIGN ASAT;Mn;9;NSM;;;;;N;;;;;

The only difference between U+1036 and U+1037 is the Canonical Combining Class (fourth item, 0 vs. 7).

The code chart for Myanmar is at http://www.unicode.org/charts/PDF/U1000.pdf.
Relevant information about the script in the Unicode Standard is at http://www.unicode.org/versions/Unicode8.0.0/ch12.pdf (pp. 11ff, in paricular the table at p. 13).

The idea behind the behavior of String#succ is to use each character as a digit and circle through the characters in the same alphabet. The simplest case is a..z or A..Z. The implementation works to some extent for many other scripts, but is dependent on things such as whether the characters appear contiguously in the relevant character encoding,...

It is unclear what characters 'ideally' should be looped though for Myanmar. For example, the W3C does not (yet?) have an alphabetic list style for Myanmar (see http://www.w3.org/TR/predefined-counter-styles/#myanmar-styles); the same applies for most related scripts (Indic/South East Asian). There are good arguments for looking only through the (base) consonants (U+1000..U+1020). Some variations might include independent vowels, and language-specific variants may include the relevant extension characters.

In the current implementation, the behavior observed seems to be a consequence of how the String#succ method uses character data provided by Oniguruma/Onigumo. As the subject of the bug say, the current behavior is indeed surprising. But the current implementation isn't really of any use for any but some very selected scripts, and Myanmar is definitely not among them.

Once we have information from some reliable source what characters are most suitable to loop through in Myanmar, we can think about how to fix this problem. So I'm going to set this to "feedback".

#5 - 07/15/2019 04:54 AM - jeremyevans0 (Jeremy Evans)

- Status changed from Feedback to Closed

This was fixed between 2.0 and 2.1:

```
$ ruby20 -e 'p "\u1036".succ'
"\u1000\u1000"
$ ruby21 -e 'p "\u1036".succ'
"\u1038"
```