Ruby trunk - Feature #13240

Change Unicode property implementation in Onigmo from inversion lists to direct lookup

02/22/2017 08:01 AM - duerst (Martin Dürst)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

For Unicode property checks (e.g. /\p{hiragana}/), Onigmo is currently using inversion lists. See enc/unicode/9.0.0/name2ctype.h; the about 500 arrays starting with CR_NEWLINE, currently on line 39, are all inversion lists.

I propose to change this to use direct lookup. Takumi Koyama, a student of mine, has implemented direct lookup. Our new implementation uses less memory (213'920 vs. 240,976 bytes) while supporting more properties (76 vs. 62) and more property values (1009 vs. 554).

We are also faster on checking single properties, up to 9 times faster for the actual check depending on property value. This is because inversion lists use binary search, and so depends on the length of the inversion list (O(log n), Age3.0 is longest), whereas we just use direct lookup, which is a constant-time operation. But we are also somewhat faster for very short inversion lists, i.e. blocks (which by definition have only one range).

Where we may get slower is for character classes with multiple properties (e.g. /[\p{han}\p{hiragana}\p{katakana}...]/). This is because inversion lists are easily mergeable (when compiling the regular expression), and can also be combined with character class ranges. On the other hand, direct lookup isn't easily mergeable. This may need further investigation (what kinds of uses for Unicode properties in Ruby regular expressions are popular/frequent).

Related issues:

Related to Ruby trunk - Feature #13241: Method(s) to access Unicode propertie...	**Open**

History

#1 - 02/22/2017 08:04 AM - duerst (Martin Dürst)

- Related to Feature #13241: Method(s) to access Unicode properties for characters/strings added