10. Regular expression matching

martes, 19 de septiembre de 2023 9:54 a.m.

Example 1

"aa", "a" → false — a does not match aa

string, pattern

Example 2

"aa", "a*" → true

a match a, and * matches zero or more, prev char.

Example 3

"ab", ".*" → true

. match any single char, and * match zero or more prev char.

Example 4

"aab", "c*a*b" → true

c* means it could be zero or more c, which is true
a* means it could be zero or more a, there are two
b means there should be a "b" after all the "a"

"aab" — a match c? → false

"c*a*b" — So, pattern shouldn't be compared 1 by 1, as there are cases where false negatives or positives may occur.

"c*a*b" — take first char, and check if next one is * if it is, start comparing the string by letter.

if first character does not match, call again the function, without the first pattern.

"c*a*b" → take c*

"aab" is "c"? → NO → Call again isMatch("a*b")

as * means zero or more the patter is fulfiled

return patternMatch && isMatch("aab", "a*b")

"a*b" → take a*

pattern → a (zero or more)

"aab" → is "a"? yes

"aab" → is "a"? yes

"aab" → is "a"? NO → call again

return patternMatch && isMatch("b", "b")

"b" → take b

"b" match "b"? → yes

1.- Find the first portion of the pattern, it can be any english char, a point (.) or a combination with *

2.- Loop the string right to left, checking if the last char match the pattern.

3.- if do match, call function recursiveley cutting the string and pattern.

4.- if don't, return false

DP Solution

"aab", "c*a*b"

DP[0][0] = true, as empty string match empty pattern

		c	*	a	*	b
	0	1	2	3	4	5
0	T	F	T	F	T	F
a 1	F	F	F	T	T	F
a 2	F	F	F	F	T	F
b 3	F	F	F	F	F	T

1.- DP[0][0] is always true

2.- Fill first row (empty string) comparing the pattern

3.- Iterate each pattern and each string

4.- IF (p(j-1) = s(i-1)) or (p(j-1) = ".")

→ dp[i][j] = dp[i-1][j-1]

5.- IF (p(j-1) = "*")

→ dp[i][j] = dp[i][j-2] or s[i-1] = p[j-2] or p(j-2) = "." and dp[i-1][j]