Purely functional palindromic trees
Iteration III results

Timur Khazhiev
Innopolis University
t.khazhiev@innopolis.ru

Nikolai Kudasov*
Innopolis University
n.kudasov@innopolis.ru

ABSTRACT

This is the results of BS-3 Project and it contains a general project results so far.

CCS CONCEPTS

• **Theory of computation → Data structures design and analysis**;

KEYWORDS

palindrome, eertree, purely functional, persistent

1 DOUBLE ENDED TREE: MERGE ALGORITHM

During this iteration idea of better merging two trees was implemented (which was described in previous iteration [1]). But more optimized way of merging two trees was found.

In the structure we hold list of maximum suffix palindromes (or prefix). ith element of that list corresponds to maximum suffix palindrome of subsequence $S[0..i]$ of processed sequence S.

For example that list for string "$abaab$" it will be $[a, b, aba, aa, baab]$ (suffixes):

$S[0..i]$:	$\underline{a}$	$a\underline{b}$	$a\underline{ba}$	$ab\underline{aa}$	$a\underline{baab}$
list:	a	b	aba	aa	$baab$

So the length of that list should be equal to the length of sequence. But in case of merging two trees we are not considering order and size of that list. Originally we grow the one of trees (simply appending or prepending which considers order) and then concatenate list from second tree, that adds redundant entries. But actually new found palindromes replaces palindromes in that list. For example merging "$abaab$" and "bab" ("$abaabbab$") should give us $[a, b, aba, aa, baab, bb, abba, bab]$

					"$abaab$"			"bab"
	$[a,$	$b,$	$aba,$	$aa,$	$baab]$	$[\underline{b},$	$\underline{a},$	$bab]$
merged:	$[a,$	$b,$	$aba,$	$aa,$	$baab,$	$\underline{bb},$	$\underline{abba},$	$bab]$

From that example we can see that we actually just replace some amount of elements from that concatenated list. Now we need find these position of replaces, which is simple: since we growing our tree with appending or prepending, new palindromes are already in list. Lastly we need add remainder element from another list. So now list of merged holds elements in right order.

New problem emerges when tree is made from combination of appending and prepending: that list will represent mixture $maxSuffixes$ and $maxPrefixes$ without any orded. And merge won't work properly.

2 TWO LISTS FOR TREE

To solve problem mentioned above we can hold two lists in tree: for prefixes and for suffixes. For append operation list of suffixes is updated as originally. To update list of prefixes we need to look in that list for palindromes which started from that appended symbol (which is actually $directLink$ from previous element from that list). Then we update these elements.

3 OBSERVATION ON MAXSUFFIX LENGTH

REFERENCES

[1] Timur Khazhiev. 2019. Iteration III detailed report. https://github.com/khazhix/ds-project/blob/iteration_iii/Iteration_III.pdf. (2019).

---

*Project supervisor.