# A Suffix Tree or Not a Suffix Tree?

Tatiana Starikovskaya & Hjalte Wedel Vildhøj

Higher School of Economics Russia Technical University of Denmark

IWOCA 2014 October 16







## What do suffix trees look like?



## What does suffix trees look like?







Input: An unlabeled ordered tree T on n nodes

**Output:** YES if T is a suffix tree NO otherwise



- Input: An unlabeled ordered tree T on n nodes
- **Output:** YES if T is a suffix tree NO otherwise



### **Naive approaches**

We can decide whether T is a **\$**-suffix tree in  $O^*(\sigma^{\# leaves})$  time ..... but not clear how to decide if T is a suffix tree by exhaustive search.

- Input: An unlabeled ordered tree T on n nodes
- **Output:** YES if T is a suffix tree NO otherwise

### Naive approaches

We can decide whether T is a **\$**-suffix tree in  $O^*(\sigma^{\# eaves})$  time ..... but not clear how to decide if T is a suffix tree by exhaustive search.

### Theorem





1. Suppose S realizes T

## S = acacbacbacbaccbacb

1. Suppose S realizes T



## $S = \begin{array}{cccc} \begin{smallmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ acacbacbacbacbaccbacbacb$

- 1. Suppose S realizes T
- 2. Assume wlog that ST(S) has a leaf edge of length 1



## S = acacbacbacbacbacbacb

- 1. Suppose S realizes T
- 2. Assume wlog that ST(S) has a leaf edge of length 1



## $S = \begin{array}{cccc} \begin{smallmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ acacbacbacbaccbaccbaccbacb \\ \end{array}$

- 1. Suppose S realizes T
- 2. Assume wlog that ST(S) has a leaf edge of length 1



## S = acacbacbacc

- 1. Suppose S realizes T
- 2. Assume wlog that ST(S) has a leaf edge of length 1
- 3. Let v be the last leaf, and p be v's parent



- 1. Suppose S realizes T
- 2. Assume wlog that ST(S) has a leaf edge of length 1
- 3. Let v be the last leaf, and p be v's parent
- 4. The edge  $p \rightarrow v$  has length 1



- 1. Suppose S realizes T
- 2. Assume wlog that ST(S) has a leaf edge of length 1
- 3. Let v be the last leaf, and p be v's parent
- 4. The edge  $p \rightarrow v$  has length 1
- 5. |S| = #leaves + |str(p)|  $\leq$  #leaves + #internal-1 = n-1



- 1. Suppose S realizes T
- 2. Assume wlog that ST(S) has a leaf edge of length 1
- 3. Let v be the last leaf, and *p* be v's parent
- 4. The edge  $p \rightarrow v$  has length 1
- 5. |S| = # |eaves + |str(p)| $\leq$  #leaves + #internal-1 = n-1



- 1. Suppose S realizes T
- 2. Assume wlog that ST(S) has a leaf edge of length 1
- 3. Let v be the last leaf, and *p* be v's parent
- 4. The edge  $p \rightarrow v$  has length 1
- 5. |S| = # |eaves + |str(p)| $\leq$  #leaves + #internal-1 = n-1





Trivial observation:  $|S| \ge \# \text{leaves} = n-1$ 



Trivial observation:  $|S| \ge \# \text{leaves} = n-1$ 

T can be realized by S = abcdef..... of length |S|=n-1

Input: An unlabeled ordered tree T on n nodes

**Output:** YES if T is a suffix tree NO otherwise



# The Suffix Tree Light Version Decision Problem

**Input:** An unlabeled ordered tree T on n nodes with suffix links and first letters

**Output:** YES if T is a \$-suffix tree NO otherwise

# The Suffix Tree Light Version Decision Problem

**Input:** An unlabeled ordered tree T on n nodes with suffix links and first letters

**Output:** YES if T is a **\$**-suffix tree NO otherwise

Previous work

I et al. [Discrete Appl. Math. 163, 2014] gave an O(n) time algorithm for deciding if T is a \$-suffix tree.

Cazaux & Rivals [J. Discrete Algorithms, 2014] considered the problem of deciding if T is a \$-suffix tree when no first letters are given.

# The Suffix Tree Light Version Decision Problem

**Input:** An unlabeled ordered tree T on n nodes with suffix links and first letters

**Output:** YES if T is a suffix tree NO otherwise

Previous work

I et al. [Discrete Appl. Math. 163, 2014] gave an O(n) time algorithm for deciding if T is a \$-suffix tree.

Cazaux & Rivals [J. Discrete Algorithms, 2014] considered the problem of deciding if T is a \$-suffix tree when no first letters are given.

Our contribution: We show how to decide if T is a suffix tree in O(n) time.













## **Open Problems**



# **Open Problems**

Input: An unlabeled ordered tree T on n nodesOutput: YES if T is a suffix tree NO otherwise



Can this problem be solved in poly(n) time?