Automata approach to inexact tree pattern matching using 1-degree edit distance

Eliška Šestáková Ondřej Guth Jan Janoušek

Czech Technical University in Prague

PSC 2021

1. The inexact tree pattern matching problem

2. Automata-based solution
   - Pushdown automaton
   - Finite automaton

3. Dynamic programming

1-degree edit operations[1]

Edit operations applied to a *labeled, ordered* tree $T = (V, E)$:

- **vertex relabel** change the label of a vertex $v \in V$,
- **leaf insert** insert a vertex v as a leaf of an existing vertex $u \in V$, and
- **leaf delete** delete a non-root leaf $v \in V$.

[1] Stanley M Selkow. "The tree-to-tree editing problem". In: *Inf. Process. Lett.* 6.6 (1977), pp. 184–186.

Unit cost 1-degree edit distance

Definition (Unit cost 1-degree edit distance)

The *unit cost 1-degree edit distance* is a function $d : \mathrm{TR}(\Sigma)^a \times \mathrm{TR}(\Sigma) \to \mathbb{N}_0$. Given two trees T_1 and T_2, the number $d(T_1, T_2)$ corresponds to the minimal number of 1-degree edit operations that transform T_1 into T_2.

[a] $\mathrm{TR}(\Sigma)$ denotes the set of all labeled ordered trees over alphabet Σ

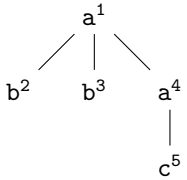
Inexact 1-degree tree pattern matching problem

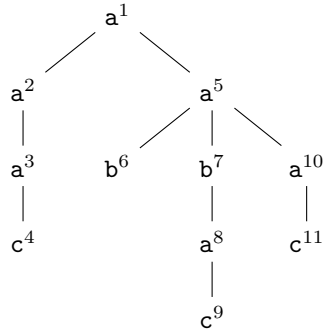
Problem (Inexact 1-degree tree pattern matching problem)

Let Σ be an alphabet. Let $T = (V_T, E_T)$ be an input tree with n vertices over Σ. Let $P = (V_P, E_P)$ be a comparatively smaller tree pattern over Σ with m vertices. Let k be a non-negative integer representing the maximum number of errors allowed. Let d be the unit cost 1-degree edit distance function. Given T, P, k, and d, the inexact 1-degree tree pattern matching problem is to return a set $\{v : v \in V_T \wedge d(T_v, P) \leq k\}$.

Inexact 1-degree tree pattern matching problem

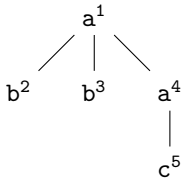
Example, $k = 2$





Prefix bar notation of a tree[2]

$$\text{PREF-BAR}(P) = a\,\text{PREF-BAR}(P_1)\,\text{PREF-BAR}(P_2)\ldots\text{PREF-BAR}(P_s)\,|,$$



$$\text{PREF-BAR}(P) = \mathtt{ab|b|ac|||}$$

[2] Jan Janoušek. *Arbology: Algorithms on trees and pushdown automata*. Habilitation thesis, Brno University of Technology, 2010.

Prefix bar notation of a tree

$$\texttt{ab|b|ac|||}$$

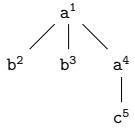
$$\texttt{(a(b)(b)(a(c)))}$$

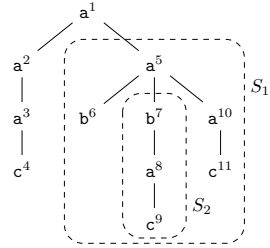
Unit cost 1-degree edit distance with prefix bar notation

relabeling $\mathrm{R}(i, b)$ that for $i \in \{1, \ldots, r-1\}$, $b \in \Sigma$, and $\boldsymbol{x}[i] \in (\Sigma \setminus \{b\})$, change the symbol $\boldsymbol{x}[i]$ into symbol b;

insertion $\mathrm{I}(i, a)$ that for $i \in \{2, \ldots, r-1\}$ and $a \in \Sigma$ inserts the substring (leaf) "$a|$" at position i; and

deletion $\mathrm{D}(i)$ that for $i \in \{2, \ldots r-2\}$, $\boldsymbol{x}[i] \in \Sigma$, and $\boldsymbol{x}[i+1] = |$, deletes the substring (leaf) $\boldsymbol{x}[i]\boldsymbol{x}[i+1]$.





$\text{PREF-BAR}(P) = \mathtt{ab|b|ac|||}$ $\quad \text{PREF-BAR}(S_1) = \mathtt{ab|bac|||ac|||}$ $\quad \text{PREF-BAR}(S_2) = \mathtt{bac|||}$

Example

$d(P, S_1) = 2$ and $d(P, S_2) = 3$ since

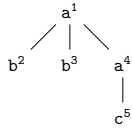
$$\mathtt{ab|b|ac|||} \xrightarrow{I(5,a)} \mathtt{ab|ba||ac|||} \xrightarrow{I(6,c)} \mathtt{ab|bac|||ac|||}$$

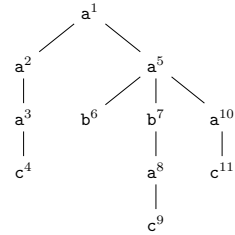
$$\mathtt{ab|b|ac|||} \xrightarrow{R(1,b)} \mathtt{bb|b|ac|||} \xrightarrow{D(2)} \mathtt{bb|ac|||} \xrightarrow{D(2)} \mathtt{bac|||}$$

Problem statement with prefix bar notation

Problem (inexact 1-degree tree pattern matching)

The problem of inexact 1-degree tree pattern matching is finding all positions $j \in \{1, \ldots, 2n\}$ *in* $\text{PREF-BAR}(T)$ *such that* $\text{PREF-BAR}(T)[j] = |$ *and* $d(\text{PREF-BAR}(P), \text{PREF-BAR}(T)[i \ldots j]) \leq k$*, where* i *is the start position of the subtree that ends at position* j*.*





Example ($k = 2$)

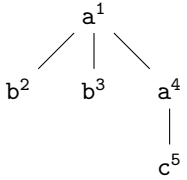
$$\text{PREF-BAR}(P) = \mathsf{ab}|\mathsf{b}|\mathsf{ac}|||$$

$$\text{PREF-BAR}(T) = \mathsf{a}\underbrace{\mathsf{aac}|||}\overbrace{\mathsf{ab}|\mathsf{bac}|||\mathsf{ac}|||}|$$

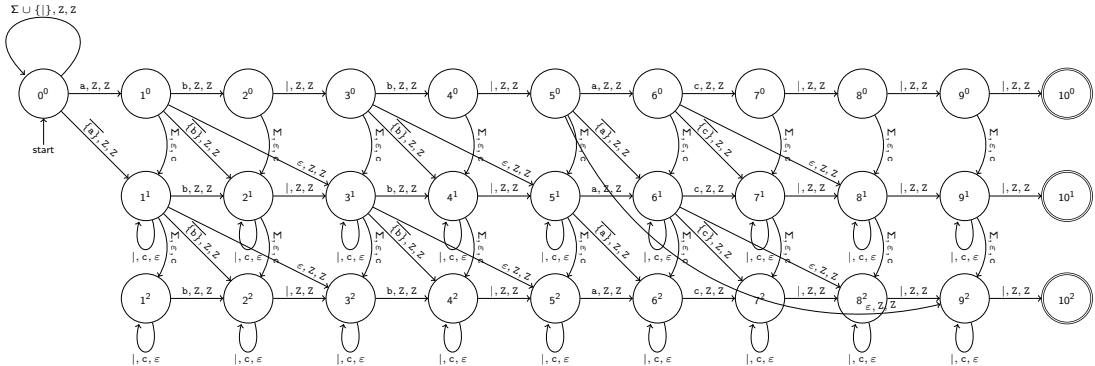
Algorithm outline

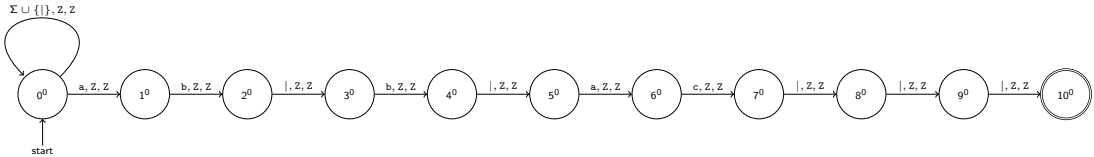
1. Construct (string) pattern-matching automaton $\mathcal{M}_{\mathrm{PDA}}$ for $\textsc{pref-bar}(P)$ and k.
2. Execute $\mathcal{M}_{\mathrm{PDA}}$ over $\textsc{pref-bar}(T)$.

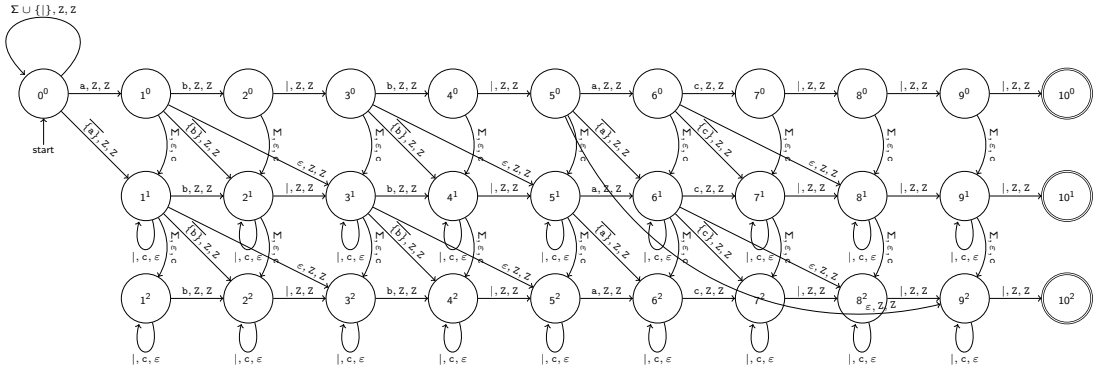
Example: Tree pattern P and $k = 2$



ab|b|ac|||


Σ ∪ {|}, Z, Z
start
a, Z, Z
b, Z, Z
|, Z, Z
[a], Z, Z
[b], Z, Z
[c], Z, Z
ε, Z, Z
|, c, ε
ε, |, c
ε, Z, Z


Σ ∪ {|}, Z, Z
0⁰
a, Z, Z
1⁰
b, Z, Z
2⁰
|, Z, Z
3⁰
b, Z, Z
4⁰
|, Z, Z
5⁰
a, Z, Z
6⁰
c, Z, Z
7⁰
|, Z, Z
8⁰
|, Z, Z
9⁰
|, Z, Z
10⁰
start

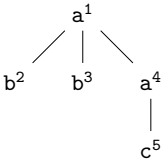

Σ ∪ {|}, Z, Z
start
a, Z, Z
b, Z, Z
|, Z, Z
[a], Z, Z
[b], Z, Z
[c], Z, Z
ε, Z, Z
|, c, ε
ε, |, c
ε, Z, Z

Subtree jump table

Definition (Subtree jump table[a])

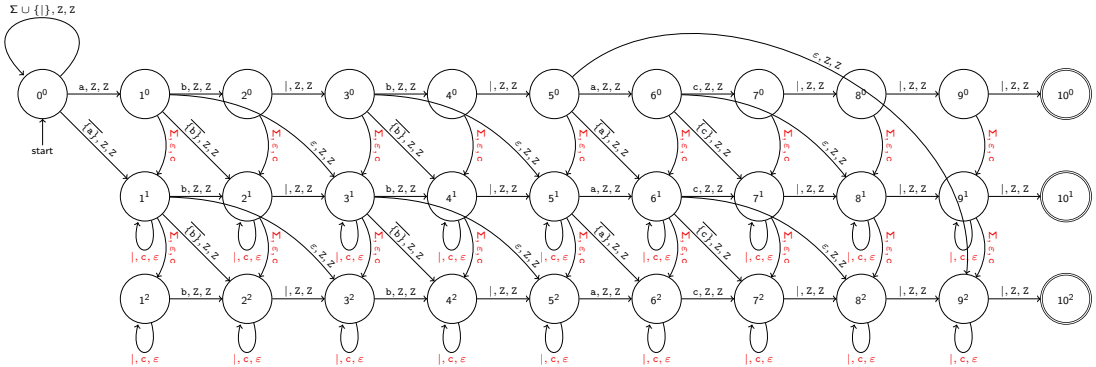
[a] Jan Trávníček. "(Nonlinear) Tree Pattern Indexing and Backward Matching". PhD thesis. Faculty of Information Technology, Czech Technical University in Prague, 2018.

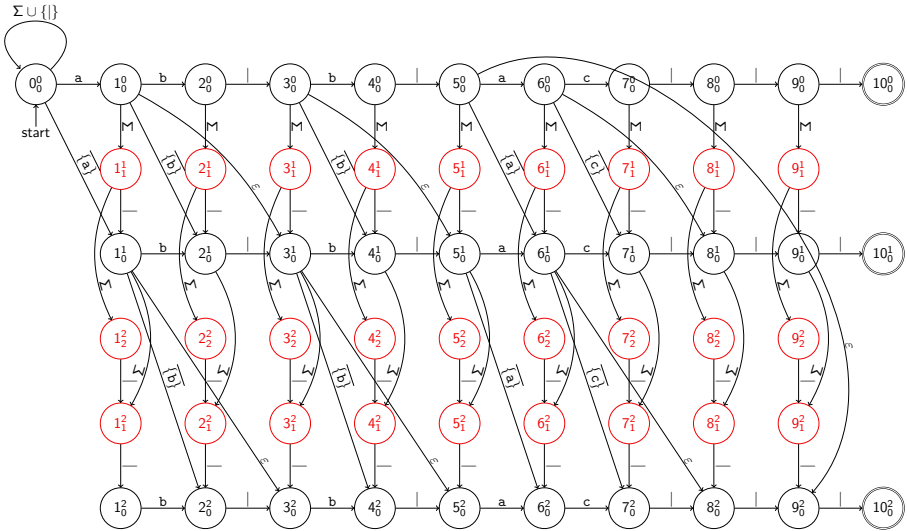
Given a tree T with n vertices and its prefix bar notation $\text{PREF-BAR}(T)$ with length $2n$, the *subtree jump table* $\boldsymbol{S}_T$ for T is a mapping from a set of integers $\{1, \ldots, 2n\}$ into a set of integers $\{0, \ldots, 2n+1\}$. If the substring $\boldsymbol{x}[i \ldots j]$, where $1 \leq i < j$ is the prefix bar representation of a subtree of T, then $\boldsymbol{S}_T[i] = j + 1$ and $\boldsymbol{S}_T[j] = i - 1$.

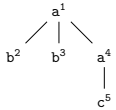


$\text{PREF-BAR}(P)$	a	b	|	b	|	a	c	|	|	|
j	1	2	3	4	5	6	7	8	9	10
$\boldsymbol{S}_P[j]$	11	4	1	6	3	10	9	6	5	0

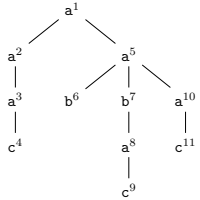
Use of pushdown store



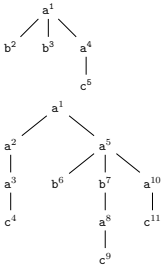

Σ ∪ {|}
start
a
b
c
ε

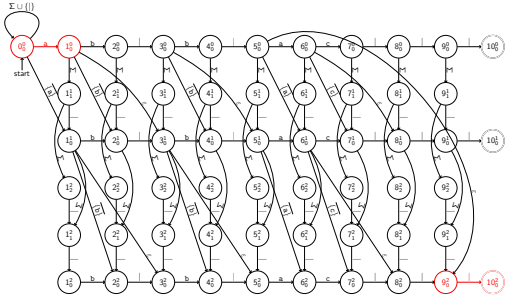


ab|b|ac|||



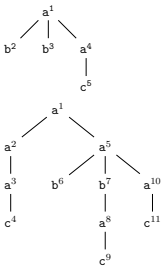
aaac|||ab|bac|||ac||||

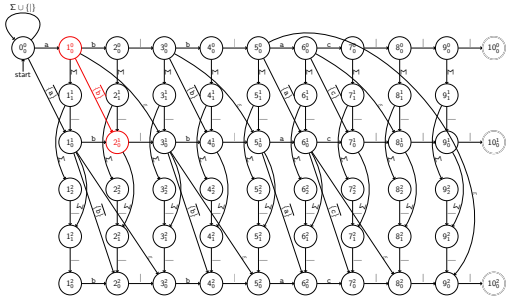




		1	2	3	4	5	6	7	8	
		a	a	a	c				a	...
	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	
a	∞, ∞, ∞	0, ∞, ∞	0, 1, ∞	0, 1, 2	1, 1, 2	1, 2, ∞	2, ∞, ∞	∞, ∞, ∞	0, ∞, ∞	
b	∞, ∞, ∞	∞, ∞, ∞	1, ∞, ∞	1, 2, ∞	1, 2, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
b	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	1, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
a	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	2, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
c	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	

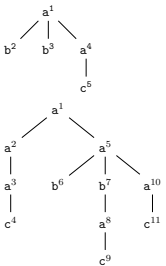
$$\boldsymbol{D}_{i,j,0} = \boldsymbol{D}_{i-1,j-1,0} : \boldsymbol{t}[i] = \boldsymbol{p}[j] \wedge 1 \leq i \leq 2n \wedge 1 \leq j \leq 2m$$

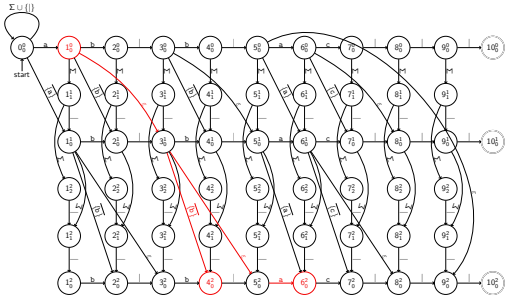




		1	2	3	4	5	6	7	8	. . .
		a	a	a	c	|	|	|	a	
	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	
a	∞, ∞, ∞	0, ∞, ∞	0, 1, ∞	0, 1, 2	1, 1, 2	1, 2, ∞	2, ∞, ∞	∞, ∞, ∞	0, ∞, ∞	
b	∞, ∞, ∞	∞, ∞, ∞	1, ∞, ∞	1, 2, ∞	1, 2, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	1, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
b	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	2, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
a	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
c	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	

$$\boldsymbol{D}_{i,j,0} = \boldsymbol{D}_{i-1,j-1,0} + 1 : \boldsymbol{t}[i], \boldsymbol{p}[j] \in \Sigma \wedge 1 \leq i \leq 2n \wedge 1 \leq j \leq 2m$$

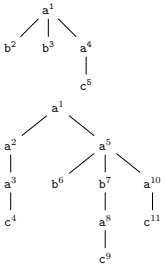


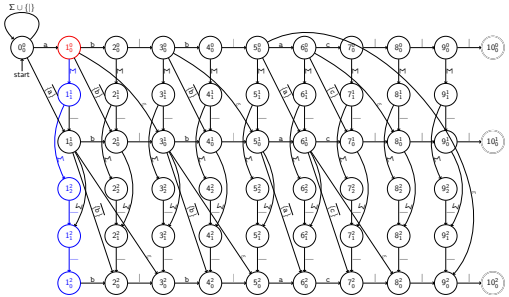


		1	2	3	4	5	6	7	8	
		a	a	a	c				a	...
	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	
a	∞, ∞, ∞	0, ∞, ∞	0, 1, ∞	0, 1, 2	1, 1, 2	1, 2, ∞	2, ∞, ∞	∞, ∞, ∞	0, ∞, ∞	
b	∞, ∞, ∞	∞, ∞, ∞	1, ∞, ∞	1, 2, ∞	1, 2, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	1, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
b	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	2, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
a	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
c	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	

$$\boldsymbol{D}_{i,j,0} = \boldsymbol{D}_{i-1,\boldsymbol{S}_P[h],0} + \frac{j - \boldsymbol{S}_P[h] + 1}{2} : \boldsymbol{t}[i] = \boldsymbol{p}[j] \wedge \boldsymbol{p}[j-1] = | \ \wedge 1 \leq i \leq 2n \wedge 2 \leq j \leq 2m \wedge 1 \leq h \leq 2m \wedge \boldsymbol{S}_P[h] < j$$

$$\boldsymbol{D}_{i,j,0} = \boldsymbol{D}_{i-1,\boldsymbol{S}_P[h],0} + \frac{j - \boldsymbol{S}_P[h] + 2}{2} : \boldsymbol{t}[i], \boldsymbol{p}[j] \in \Sigma \wedge \boldsymbol{t}[i] \neq \boldsymbol{p}[j] \ \wedge \boldsymbol{p}[j-1] = | \wedge 1 \leq i \leq 2n \wedge 2 \leq j \leq 2m \wedge 1 \leq h \leq 2m \wedge \boldsymbol{S}_P[h] < j$$





		1	2	3	4	5	6	7	8	
		a	a	a	c				a	...
	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	0, ∞, ∞	
a	∞, ∞, ∞	0, ∞, ∞	0, 1, ∞	0, 1, 2	1, 1, 2	1, 2, ∞	2, ∞, ∞	∞, ∞, ∞	0, ∞, ∞	
b	∞, ∞, ∞	∞, ∞, ∞	1, ∞, ∞	1, 2, ∞	1, 2, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	1, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
b	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	2, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
a	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
c	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	
|	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	∞, ∞, ∞	2, ∞, ∞	∞, ∞, ∞	

$$\boldsymbol{D}_{i,j,c} = \boldsymbol{D}_{i-1,j,c-1} + 1 : \boldsymbol{t}[i] \in \Sigma \wedge 1 \leq i \leq 2n \wedge 1 \leq j \leq 2m \wedge 1 \leq c \leq k$$

$$\boldsymbol{D}_{i,j,c} = \boldsymbol{D}_{i-1,j,c+1} : \boldsymbol{t}[i] = | \wedge 1 \leq i \leq 2n \wedge 1 \leq j \leq 2m \wedge 0 \leq c < k$$

Complexity

Theorem (Space complexity)

Inexact 1-degree tree pattern matching can be solved using $\mathcal{O}(km)$ space.

Complexity

Theorem (Space complexity)

Inexact 1-degree tree pattern matching can be solved using $\mathcal{O}(km)$ space.

Theorem (Time complexity)

Inexact 1-degree tree pattern matching can be solved in $\mathcal{O}(kmn)$ time.

Conclusions

- solution for unit cost inexact 1-degree tree pattern matching
- string solution for unranked prefix bar notation
- pushdown automaton
- finite automaton
- dynamic programming
- all algorithms can be extended for non unit cost 1-degree distance