Closest Substring Problems for Regular Languages

Yo-Sub Han[1] Sang-Ki Ko[2] Timothy Ng[3]
Kai Salomaa[4]

[1]Department of Computer Science, Yonsei University
[2]AI Research Center, Korea Electronics Technology Institute
[3]David R. Cheriton School of Computer Science, University of Waterloo
[4]School of Computing, Queen's University

DLT 2018, Tokyo, Japan

Given a set of strings, does there exist a string that is close to all of them?

Given a set of k strings $s_1, s_2, \ldots, s_k$ of equal length and a positive integer r, does there exist a string s such that for each $1 \leq i \leq k$, the Hamming distance between s and s_i is at most r?

Given a set of k strings $s_1, s_2, \ldots, s_k$ of equal length and a positive integer r, does there exist a string s such that for each $1 \leq i \leq k$, the Hamming distance between s and s_i is at most r?

- This is the consensus string problem [Frances and Litman 1997].

Given a set of k strings $s_1, s_2, \ldots, s_k$ of equal length and a positive integer r, does there exist a string s such that for each $1 \leq i \leq k$, the Hamming distance between s and s_i is at most r?

- This is the consensus string problem [Frances and Litman 1997]. It is NP-complete.

Given a set of k strings $s_1, s_2, \ldots, s_k$ of equal length and a positive integer r, does there exist a string s such that for each $1 \leq i \leq k$, the Hamming distance between s and s_i is at most r?

- This is the consensus string problem [Frances and Litman 1997]. It is NP-complete.
- r is the radius.

Given a set of k strings $s_1, s_2, \ldots, s_k$ and two positive integers ℓ, r, does there exist a string s of length ℓ such that for each $1 \le i \le k$, there exists a substring s_i' of length ℓ in s_i with Hamming distance at most r from s?

Given a set of k strings $s_1, s_2, \ldots, s_k$ and two positive integers ℓ, r, does there exist a string s of length ℓ such that for each $1 \leq i \leq k$, there exists a substring s'_i of length ℓ in s_i with Hamming distance at most r from s?

- This is the closest substring problem [Frances and Litman 1997].

Given a language L and positive integers r, ℓ, does there exist a string w (a consensus substring) of length ℓ such that every string $w' \in L$ has a substring whose distance is at most r from w?

Given a language L and positive integers r, ℓ, does there exist a string w (a consensus substring) of length ℓ such that every string $w' \in L$ has a substring whose distance is at most r from w?

- We consider sets of strings that are not necessarily finite.

Given a language L and positive integers r, ℓ, does there exist a string w (a consensus substring) of length ℓ such that every string $w' \in L$ has a substring whose distance is at most r from w?

- We consider sets of strings that are not necessarily finite.
- We consider edit distances with variable cost.

Language class	Complexity (Lower/upper bound)
A set of strings	NP-complete [FL97]
Sub-regular (acyclic FAs)	(coNP,NP)-hard / Σ_2^{P}
Regular (FAs)	PSPACE-complete
Context-free	PSPACE-hard / EXPTIME
Context-sensitive	Undecidable

Table: The complexity results for the CLOSEST SUBSTRING problem when l and r are given in unary.

A distance is a function $d : \Sigma^* \times \Sigma^* \to [0, \infty)$ such that

1. $d(x, y) = 0$ if and only if $x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, y) \leq d(x, w) + d(w, y)$

The edit distance of two words x and y is the minimum cost to transform x into y by a sequence of insertion, deletion, and substitution operations.

The edit distance of two words x and y is the minimum cost to transform x into y by a sequence of insertion, deletion, and substitution operations.

- Assign cost $d(a, b)$ to each edit operation (a/b), for $a, b \in \Sigma \cup \{\varepsilon\}$.

The edit distance of two words x and y is the minimum cost to transform x into y by a sequence of insertion, deletion, and substitution operations.

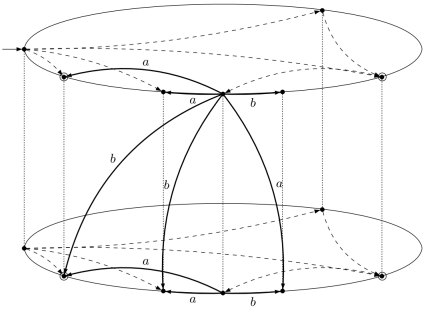
- Assign cost $d(a, b)$ to each edit operation (a/b), for $a, b \in \Sigma \cup \{\varepsilon\}$.
- The Levenshtein distance is the edit distance with unit cost for all edit operations.

The neighbourhood of a language $L \subseteq \Sigma^*$ of radius $r \geq 0$ with respect to a distance measure d is the set of all words u with $d(w, u) \leq r$ for some $w \in L$,

$$E(L, d, r) = \{u \in \Sigma^* \mid (\exists w \in L)\, d(w, u) \leq r\}.$$

Proposition (Povarov 2007)

Let A be an NFA with n states and $r \in \mathbb{N}$. The neighbourhood of $L(A)$ of radius r with respect to the additive distance d can be recognized by an NFA B with $n \cdot (r+1)$ states. The NFA B can be constructed in time that depends polynomially on n and r.



Let $c_{\min}$ be the smallest deletion or insertion cost of a symbol of the alphabet Σ and let $c_{\max}$ be the largest cost of an insertion or deletion operation.

Let $c_{\min}$ be the smallest deletion or insertion cost of a symbol of the alphabet Σ and let $c_{\max}$ be the largest cost of an insertion or deletion operation.

Some observations relating L, ℓ, and r:

Let $c_{\min}$ be the smallest deletion or insertion cost of a symbol of the alphabet Σ and let $c_{\max}$ be the largest cost of an insertion or deletion operation.
Some observations relating L, ℓ, and r:

- $\ell \leq \min(L) + \frac{r}{c_{\min}}$ since otherwise no substring of a shortest string x can be transformed into a consensus string of length ℓ by a sequence of edit operations with cost r.

Let $c_{\min}$ be the smallest deletion or insertion cost of a symbol of the alphabet Σ and let $c_{\max}$ be the largest cost of an insertion or deletion operation.

Some observations relating L, ℓ, and r:

- $\ell \leq \min(L) + \frac{r}{c_{\min}}$ since otherwise no substring of a shortest string x can be transformed into a consensus string of length ℓ by a sequence of edit operations with cost r.
- $\ell > \frac{r}{c_{\max}}$ since otherwise ε is within a distance of r of any string w of length ℓ via deleting all symbols of w.

Let $c_{\min}$ be the smallest deletion or insertion cost of a symbol of the alphabet Σ and let $c_{\max}$ be the largest cost of an insertion or deletion operation.
Some observations relating L, ℓ, and r:

- $\ell \leq \min(L) + \frac{r}{c_{\min}}$ since otherwise no substring of a shortest string x can be transformed into a consensus string of length ℓ by a sequence of edit operations with cost r.
- $\ell > \frac{r}{c_{\max}}$ since otherwise ε is within a distance of r of any string w of length ℓ via deleting all symbols of w.
- Together, this gives us

$$\frac{r}{c_{\max}} < \ell \leq \min(L) + \frac{r}{c_{\min}}.$$

Lemma

The Closest Substring problem for NFAs can be solved in PSPACE when the length ℓ of consensus substring is given in unary.

Lemma

The CLOSEST SUBSTRING problem for NFAs can be solved in PSPACE when the length ℓ of consensus substring is given in unary.

- Given an NFA A with n states, guess a string $w \in \Sigma^\ell$.

Lemma

The CLOSEST SUBSTRING problem for NFAs can be solved in PSPACE when the length ℓ of consensus substring is given in unary.

- Given an NFA A with n states, guess a string $w \in \Sigma^{\ell}$.
- Construct an NFA B for $\Sigma^* E(w, d_e, r)\Sigma^*$.

Lemma

The Closest Substring problem for NFAs can be solved in PSPACE when the length ℓ of consensus substring is given in unary.

- Given an NFA A with n states, guess a string $w \in \Sigma^\ell$.
- Construct an NFA B for $\Sigma^* E(w, d_e, r)\Sigma^*$.
- If r is given in binary, $r < c_{\max} \cdot \ell$, and the size of B is polynomial in the size of the input.

Lemma

The CLOSEST SUBSTRING problem for NFAs can be solved in PSPACE when the length ℓ of consensus substring is given in unary.

- Given an NFA A with n states, guess a string $w \in \Sigma^\ell$.
- Construct an NFA B for $\Sigma^* E(w, d_e, r)\Sigma^*$.
- If r is given in binary, $r < c_{\max} \cdot \ell$, and the size of B is polynomial in the size of the input.
- $L(A) \subseteq L(B)$ is decidable in PSPACE.

Corollary

The Closest Substring problem for NFAs can be solved in PSPACE when the radius r is given in unary.

- This follows from $\ell \leq \min(L) + \frac{r}{c_{\min}}$.

Theorem

Let d_e be an edit distance where the cost of deleting a single character σ does not depend on σ. Then the CLOSEST SUBSTRING problem for NFAs under the edit distance d_e can be solved in PSPACE.

Theorem

Let d_e be an edit distance where the cost of deleting a single character σ does not depend on σ. Then the Closest Substring problem for NFAs under the edit distance d_e can be solved in PSPACE.

- If $r < c \cdot k \cdot \min(L(A))$, where c is the cost of deletion/insertion and k is the size of the alphabet, then $\ell, r \in O(n)$, and we can use the same algorithm as in the previous lemma.

- If $r \geq c \cdot k \cdot \min(L(A))$,
  - Choose $w = (a_1 a_2 \cdots a_k)^{\min(L(A))} \cdot a_1^{\ell - \min(L(A))}$. We claim w is a consensus substring for $L(A)$ with radius r.
  - For any $z \in L(A)$, take a substring z_1 of length $\min(L(A)$.
  - To transform w into z_1, delete $(k-1) \cdot \min(L(A))$ symbols from the prefix of w to attain the word $z_1 a_1^{\ell - \min(L(A))}$ then delete the a_1's.
  - Since $\ell \leq \min(L(A)) + \frac{r}{c}$, at most $\frac{r}{c}$ deletions were performed.

Theorem

There exists an edit distance d_e such that the CLOSEST SUBSTRING problem for DFAs under the edit distance d_e is PSPACE-hard even when the length ℓ of consensus substring and radius r are given in unary.

Theorem

There exists an edit distance d_e such that the CLOSEST SUBSTRING problem for DFAs under the edit distance d_e is PSPACE-hard even when the length ℓ of consensus substring and radius r are given in unary.

- Via reduction from deciding $L(A) \subseteq \Sigma^* a \Sigma^n b \Sigma^*$, which is PSPACE-complete [Björklund, Martens, Schwentick 2013].

- Define the distance d_0 by

$$d_0(\#, a) = d_0(\#, b) = d_0(\#, c) = d_0(\#, \natural) = 1,$$
$$d_0(\sigma_1, \sigma_2) = 2 \text{ when } \sigma_1, \sigma_2 \in \{a, b, c, \natural\},\ \sigma_1 \neq \sigma_2,$$
$$d_0(\sigma, \varepsilon) = 2 \text{ for } \sigma \in \Sigma'.$$

- Let $L_n = \{awb \mid w \in \{a, b\}^n \cup \{c^n, \natural^n\}\}$ for $n \in \mathbb{N}$. The string $a\#^n b$ has inner distance n to L_n. There is no other string of length $n + 2$ with inner distance n to L_n.

Corollary

There exists an edit distance d_e such that the CLOSEST SUBSTRING problem under the edit distance d_e is PSPACE-complete both for NFAs and for DFAs.

Theorem

The CLOSEST SUBSTRING problem for acyclic NFAs is in Σ_2^{P} when the length ℓ of consensus substring is given in unary.

Theorem

The Closest Substring problem for acyclic NFAs is in Σ_2^{P} when the length ℓ of consensus substring is given in unary.

- For an acyclic NFA A, check that there exists $w \in \Sigma^\ell$ such that all strings of $L(A)$ of length at most n have a substring in $E(w, d_e, r)$.

Theorem

The CLOSEST SUBSTRING problem for acyclic DFAs is coNP-hard even when the length ℓ and radius r are given in unary.

Theorem

The CLOSEST SUBSTRING problem for acyclic DFAs is coNP-hard even when the length ℓ and radius r are given in unary.

- Via reduction from complement of SQUARE TILING; SQUARE TILING is NP-complete [van Emde Boas 1997].

Theorem

The CLOSEST SUBSTRING problem for context-free languages can be solved in EXPTIME when the length ℓ of consensus substring is given in unary.

Theorem

The Closest Substring problem for context-free languages can be solved in EXPTIME when the length ℓ of consensus substring is given in unary.

- Given a PDA P, for every string w of length ℓ, construct an NFA B for $\Sigma^* E(w, d_e, r)\Sigma^*$.

Theorem

The Closest Substring problem for context-free languages can be solved in EXPTIME when the length ℓ of consensus substring is given in unary.

- Given a PDA P, for every string w of length ℓ, construct an NFA B for $\Sigma^* E(w, d_e, r)\Sigma^*$.
- $L(P) \subseteq L(B)$ is decidable in EXPTIME since testing $L(P) \cap L(B)^c = \emptyset$ is decidable in exponential time.

Corollary

The CLOSEST SUBSTRING problem for context-sensitive languages is undecidable.

Corollary

The CLOSEST SUBSTRING problem for context-sensitive languages is undecidable.

- Since testing emptiness for context-sensitive languages is undecidable.

Language class	Complexity (Lower/upper bound)
A set of strings	NP-complete [FL97]
Sub-regular (acyclic FAs)	(coNP,NP)-hard / Σ_2^{P}
Regular (FAs)	PSPACE-complete
Context-free	PSPACE-hard / EXPTIME
Context-sensitive	Undecidable

Table: The complexity results for the CLOSEST SUBSTRING problem when l and r are given in unary.