Site-Directed Deletion

Da-Jung Cho, Yo-Sub Han, Hwee Kim, Kai Salomaa

CNRS & LSV, ENS Paris-Saclay
LRI, Université Paris-Sud

Developments in Language Theory, 2018

Outline

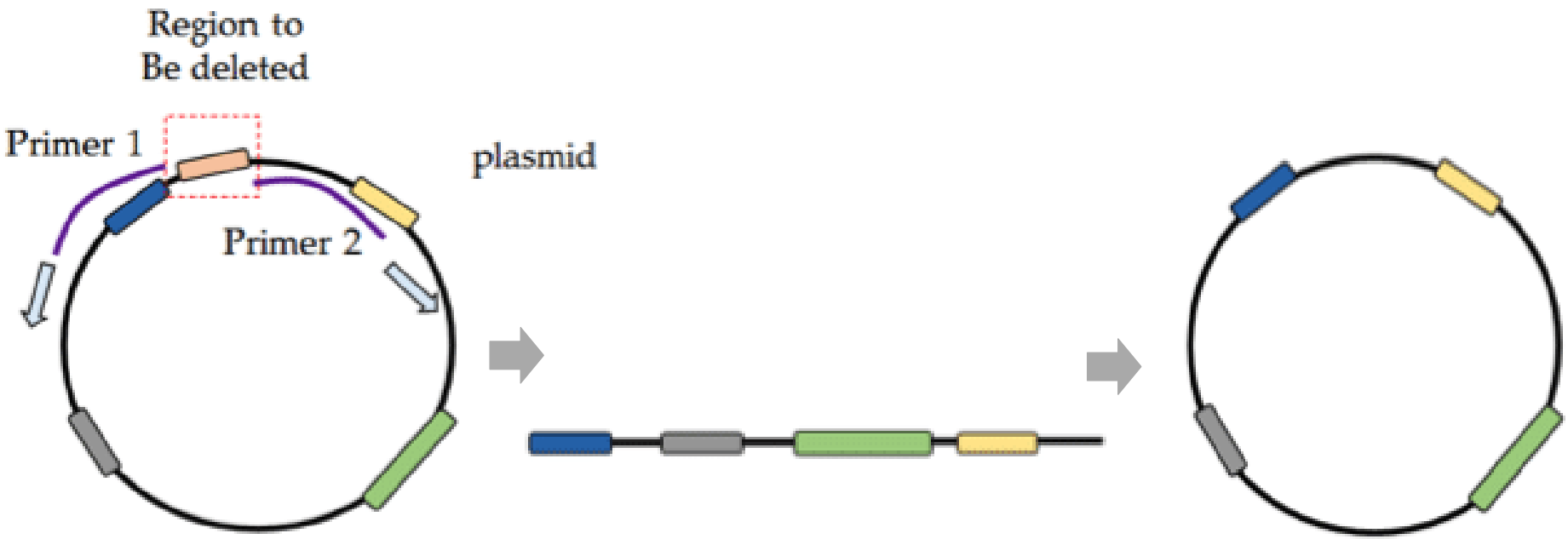
1. Introduction
   - Background
   - Definition
   - Problems
2. Main Results
   - Closure Properties of Site-Directed Deletion
   - Decidability of Site-Directed Deletion
3. Conclusion
   - Summary
   - Future Works

Outline

1. Introduction
   - Background
   - Definition
   - Problems
2. Main Results
   - Closure Properties of Site-Directed Deletion
   - Decidability of Site-Directed Deletion
3. Conclusion
   - Summary
   - Future Works

Site-Directed Deletion from Molecular Biology

- The site-directed deletion operation models *site-directed deletion mutagenesis*.
- It creates a deletion on specific sites of a given gene sequence under enzymatic activities.



Outline

1. Introduction
   - Background
   - Definition
   - Problems
2. Main Results
   - Closure Properties of Site-Directed Deletion
   - Decidability of Site-Directed Deletion
3. Conclusion
   - Summary
   - Future Works

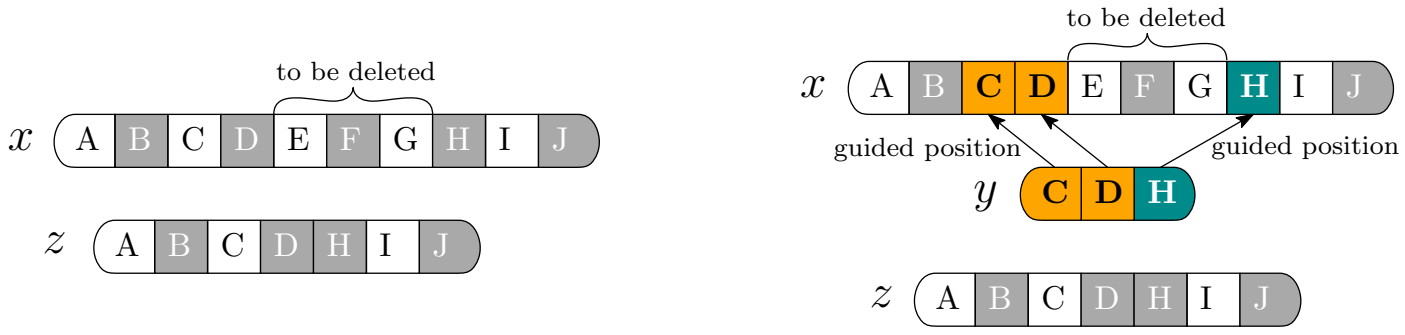
Site-Directed Deletion

Definition

The site-directed deletion from a string x by y is

$$x \overset{sdd}{\leftarrow} y = \{x_1 uvx_2 \mid x = x_1 uwvx_2, y = uv, x_1, x_2, w \in \Sigma^* \text{ and } u, v \neq \lambda\}.$$

- a string y is a *deletion guide* of x if $x \overset{sdd}{\leftarrow} y \neq \emptyset$
- if $x \overset{sdd}{\leftarrow} x$, we call it a *trivial deletion step*
- $L_1 \overset{sdd}{\leftarrow} L_2 = \bigcup_{w_i \in L_i, i=1,2} w_1 \overset{sdd}{\leftarrow} w_2$



(a) Ordinary deletion on x (b) Site-directed deletion on x and y

Iterated Site-Directed Deletion

Definition

Site-directed deletion of L is inductively defined as

$$\mathbb{SDD}^{(0)}(L) = L, \text{ and } \mathbb{SDD}^{(i+1)}(L) = \mathbb{SDD}^{(i)}(L) \overset{sdd}{\leftarrow} \mathbb{SDD}^{(i)}(L).$$

The iterated site-directed deletion $\mathbb{SDD}^*$ of L is

$$\mathbb{SDD}^*(L) = \bigcup_{i=1}^{\infty} \mathbb{SDD}^{(i)}(L).$$

Outline

1. Introduction
   - Background
   - Definition
   - Problems
2. Main Results
   - Closure Properties of Site-Directed Deletion
   - Decidability of Site-Directed Deletion
3. Conclusion
   - Summary
   - Future Works

Problems

We consider following problems

- Closure properties:
  - Given language languages L_1 and L_2, is $L_1 \overset{sdd}{\leftarrow} L_2$ regular?
  - Given context-free languages L_1 and L_2, is $L_1 \overset{sdd}{\leftarrow} L_2$ context-free?
  - Given context-free language L, is $\mathbb{SDD}^*(L)$ context-free?
- Decidability:
  - Does there exist an efficient algorithm to determine whether or not $x \overset{sdd}{\leftarrow} y \neq \emptyset$?
  - Does there exist an efficient algorithm to determine whether or not $z \in x \overset{sdd}{\leftarrow} y$?

Outline

1. Introduction
   - Background
   - Definition
   - Problems
2. Main Results
   - Closure Properties of Site-Directed Deletion
   - Decidability of Site-Directed Deletion
3. Conclusion
   - Summary
   - Future Works

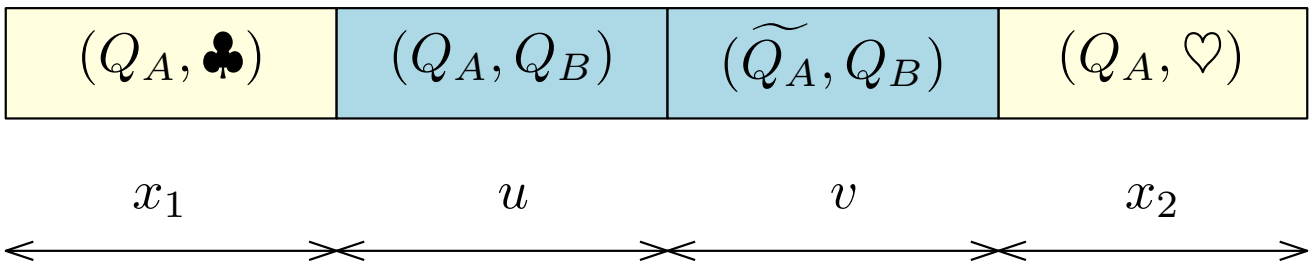
Closure Properties

Theorem

If L_1 and L_2 are regular, then $L_1 \overset{sdd}{\leftarrow} L_2$ is regular.

- Let L_1 and L_2 be recognized by NFAs $A = (Q_A, \Sigma, \delta, q_0, F_A)$ and $B = (Q_B, \Sigma, \gamma, p_0, F_B)$, respectively.
- We construct an NFA $C = (Q_C, \Sigma, \omega, s, F_C)$ recognizing all strings in $L(A) \overset{sdd}{\leftarrow} L(B)$.
- Suppose that $x_1 uwvx_2 \in L(A)$, $uv \in L(B)$ and $x_1 uvx_2 \in L(A) \overset{sdd}{\leftarrow} L(B)$

$$Q_C = Q_A \times (\{\clubsuit, \heartsuit\} \cup Q_B) \cup \widetilde{Q_A} \times Q_B$$



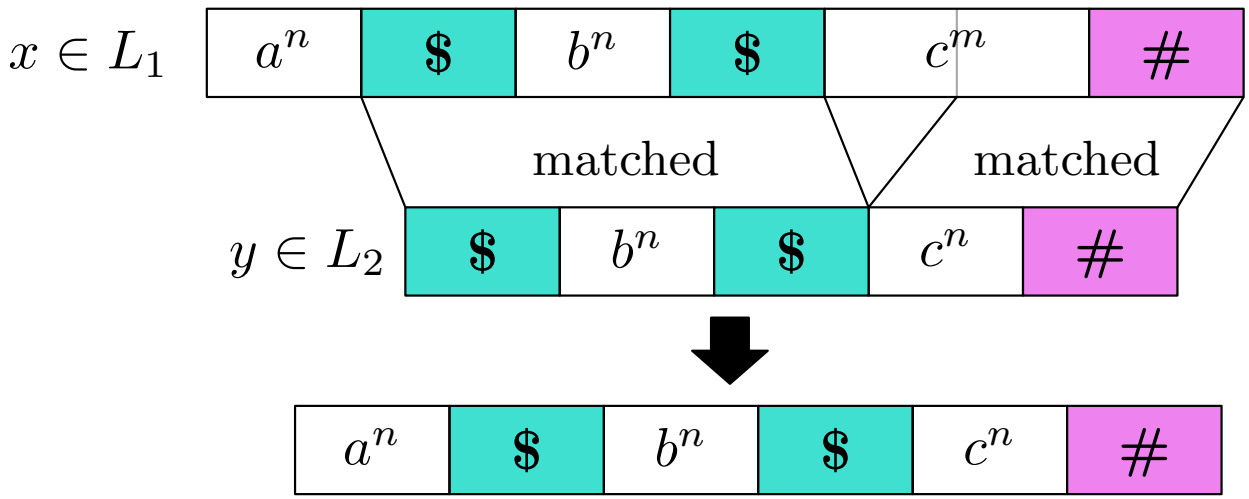
Closure Properties

Theorem

There exists context-free languages L_1 and L_2 such that $L_1 \overset{sdd}{\leftarrow} L_2$ is not context-free.

- $L_1 = \{a^n\$b^n\$c^m\# \mid m, n \geq 1\}$,
- $L_2 = \{\$b^n\$c^n\# \mid n \geq 1\}$.
- Then, we have

$$(L_1 \overset{sdd}{\leftarrow} L_2) \cap (a^*\$b^*\$c^*\#) = \{a^n\$b^n\$c^n\# \mid n \geq 1\}$$



Closure Properties

Theorem

There exists a context-free language L such that $\mathbb{SDD}^*(L)$ is not context-free.

We define $L = L_1 \cup L_2$ over $\Sigma = \{a, b, c, \$, \%, \#\}$:

- $L_1 = \{\#a^n\$b^n\$\%c^m\# \mid n, m \geq 1\}$,
- $L_2 = \{\$b^n\$c^n\# \mid n \geq 1\}$.

We will prove that

$$\mathbb{SDD}^*(L) \cap (\#a^+\$b^+\$c^+\#) = \{\#a^n\$b^n\$c^n\# \mid n \geq 1\}.$$

Closure Properties

Theorem

There exists a context-free language L such that $\mathbb{SDD}^*(L)$ is not context-free.

- We have
  - $L_1 \subseteq M_1 = \#a^+\$b^+\$\%c^+\#$, and $M_1' = \#a^+\$b^+\$c^+\#$,
  - $L_2 \subseteq M_2 = \$b^+\$c^+\#$.
- Consider four cases:
  1. $y \in M_1 \cup M_1'$ is a deletion guide of $x \in M_2$: Impossible
  2. $y \in M_1 \cup M_1'$ is a deletion guide of $x \in M_1 \cup M_1'$: a trivial deletion step
  3. $y \in M_2$ is a deletion guide of $x \in M_2$: a trivial deletion step
  4. $y \in M_2$ is a deletion guide of $x \in M_1 \cup M_1'$: ?

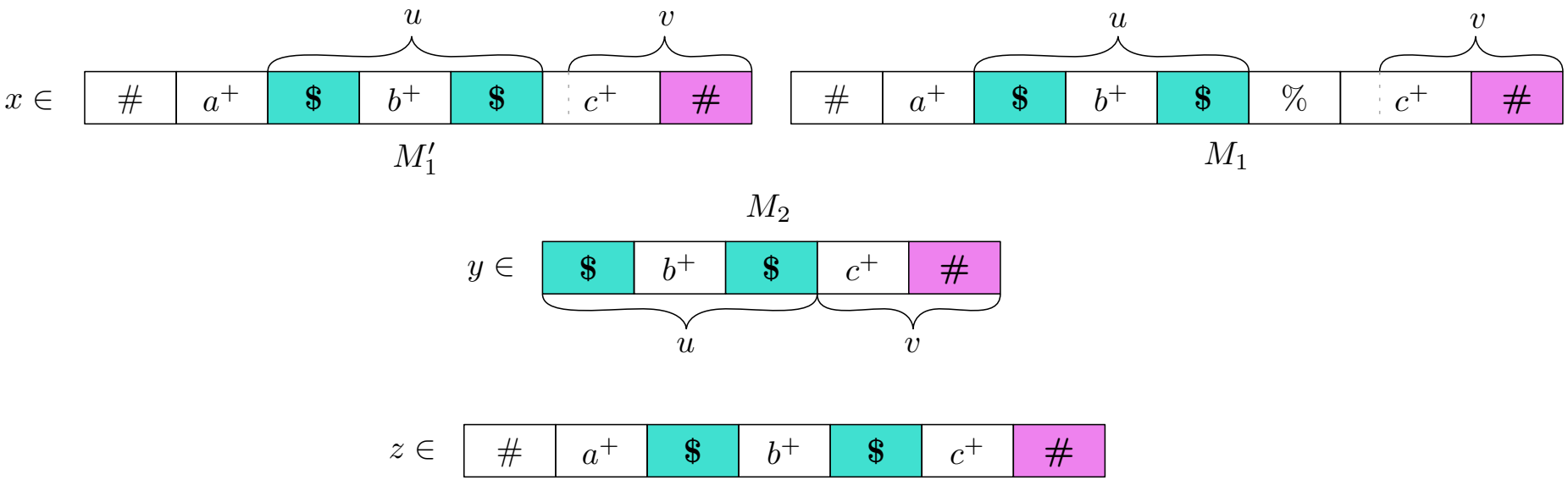
Closure Properties

Theorem

There exists a context-free language L such that $\mathbb{SDD}^*(L)$ is not context-free.

Consider a case $z \in x \overset{sdd}{\leftarrow} y$, where $x \in M_1 \cup M_1'$ and $y \in M_2$

- # of y must be matched the last symbol # of x
- two \$'s of y must be matched with two \$ of x



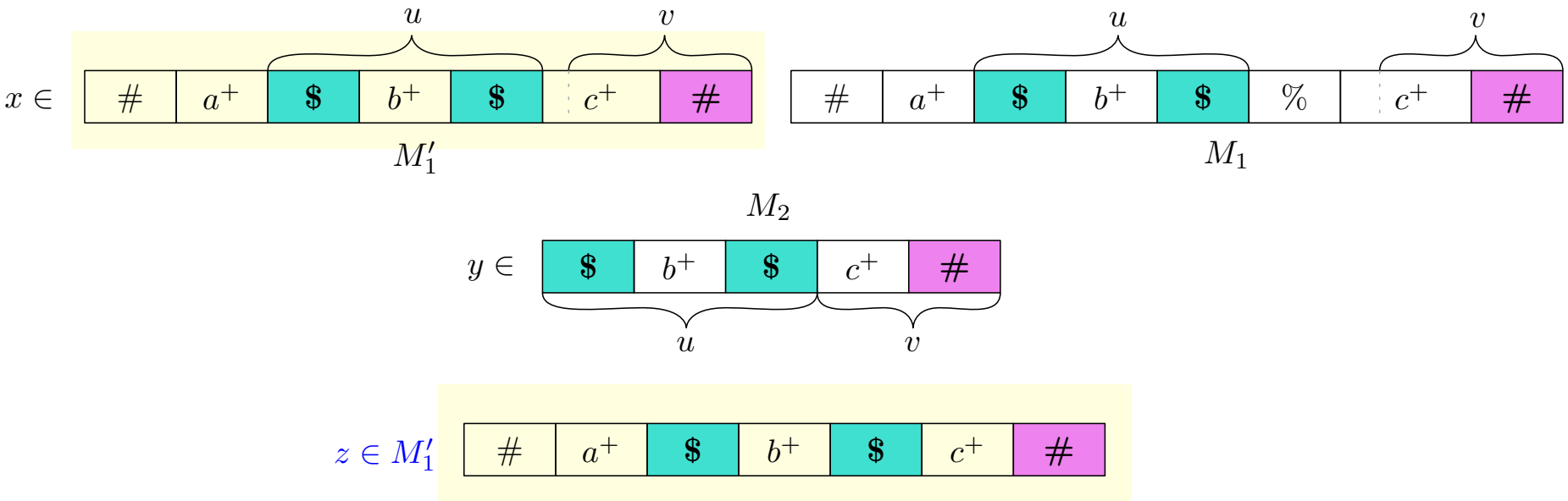
Closure Properties

Theorem

There exists a context-free language L such that $\mathbb{SDD}^*(L)$ is not context-free.

Consider a case $z \in x \overset{sdd}{\leftarrow} y$, where $x \in M_1 \cup M_1'$ and $y \in M_2$

- z must be result of deleting % and zero or more symbols c
- Then, $z \in M_1'$



Closure Properties

Theorem

There exists a context-free language L such that $\mathbb{SDD}^*(L)$ is not context-free.

1. $y \in M_1 \cup M_1'$ is a deletion guide of $x \in M_2$: Impossible
2. $y \in M_1 \cup M_1'$ is a deletion guide of $x \in M_1 \cup M_1'$: a trivial deletion step
3. $y \in M_2$ is a deletion guide of $x \in M_2$: a trivial deletion step
4. $y \in M_2$ is a deletion guide of $x \in M_1 \cup M_1'$: $z \in x \overset{sdd}{\leftarrow} y$ exists inside of M_1'

Closure Properties

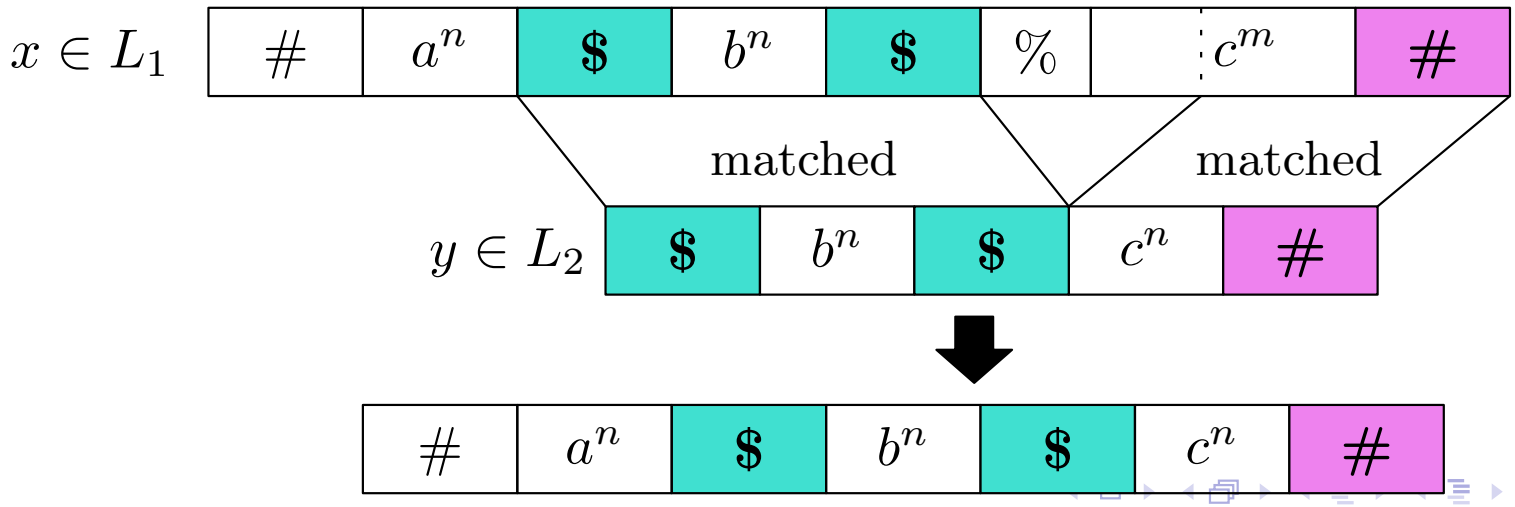
Theorem

There exists a context-free language L such that $\mathbb{SDD}^*(L)$ is not context-free.

We observe $L \cup \mathbb{SDD}^{(1)}(L) = L_1 \cup L_2 \cup (L_1 \overset{sdd}{\leftarrow} L_2)$

$$\Leftrightarrow L_1 \cup L_2 \cup \{\#a^n\$b^n\$c^n\# \mid n \geq 1\}.$$

- $L_1 = \{\#a^n\$b^n\$\%c^m\# \mid n, m \geq 1\}$,
- $L_2 = \{\$b^n\$c^n\# \mid n \geq 1\}$.



Closure Properties

Theorem

There exists a context-free language L such that $\mathbb{SDD}^*(L)$ is not context-free.

- We observe $L \cup \mathbb{SDD}^{(1)}(L) = L_1 \cup L_2 \cup (L_1 \overset{sdd}{\leftarrow} L_2)$

$$\Leftrightarrow L_1 \cup L_2 \cup \{\#a^n\$b^n\$c^n\# \mid n \geq 1\}.$$

- $\bigcup_{i=0}^{k+1} \mathbb{SDD}^{(i)}(L) = L_1 \cup L_2 \cup (\mathbb{SDD}^{(k)}(L) \overset{sdd}{\leftarrow} L_2)$
- Thus, $\mathbb{SDD}^*(L) = L \cup \mathbb{SDD}^{(1)}(L)$

Closure Properties

Theorem

There exists a context-free language L such that $\mathbb{SDD}^*(L)$ is not context-free.

We have

- $L \cup \mathbb{SDD}^{(1)}(L) = L_1 \cup L_2 \cup \{\#a^n\$b^n\$c^n\# \mid n \geq 1\}$,
- $\mathbb{SDD}^*(L) = L \cup \mathbb{SDD}^{(1)}(L)$.

Then, $\mathbb{SDD}^*(L)$ is not context-free by

$$\mathbb{SDD}^*(L) \cap \#a^+\$b^+\$c^+\# = \{\#a^n\$b^n\$c^n\# \mid n \geq 1\}.$$

Outline

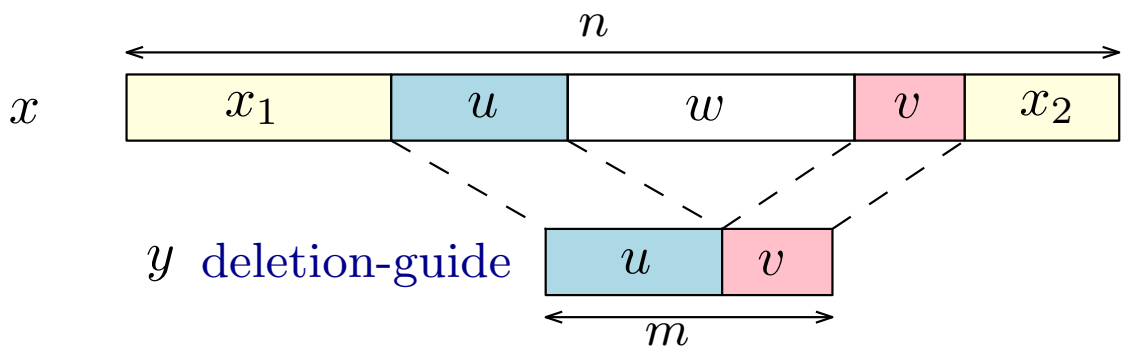
1. Introduction
   - Background
   - Definition
   - Problems
2. Main Results
   - Closure Properties of Site-Directed Deletion
   - Decidability of Site-Directed Deletion
3. Conclusion
   - Summary
   - Future Works

Decidability

Theorem

Given two strings x and y, we can determine whether or not $x \overset{sdd}{\leftarrow} y \neq \emptyset$ in $\mathcal{O}(n)$ time, where $|x| = n, |y| = m$ and $m \leq n$.

Determine whether or not there exist two substrings u and v of x, whose catenation is y.

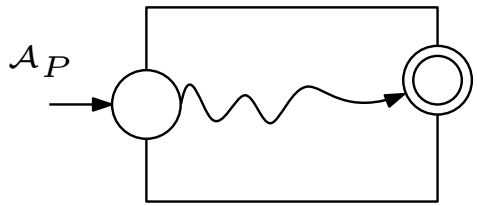


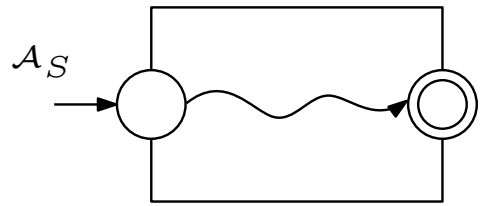
Decidability (continued)

Theorem

Given two strings x and y, we can determine whether or not $x \overset{sdd}{\leftarrow} y \neq \emptyset$ in $\mathcal{O}(n)$ time, where $|x| = n, |y| = m$ and $m \leq n$.

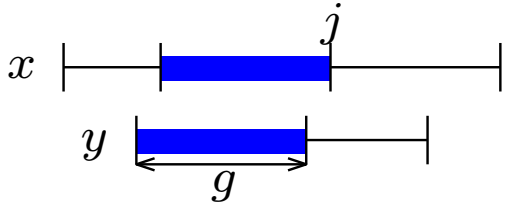
Construct two Aho-Corasick automata $\mathcal{A}_P$ ($\mathcal{A}_S$) with all prefixes of $y(y^R)$, and run them on x.

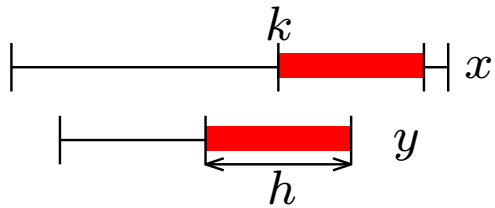




-Pattern: prefixes of y
-Result: pairs (j, g)

-Pattern: prefixes of y^R
-Result: pairs (k, h)





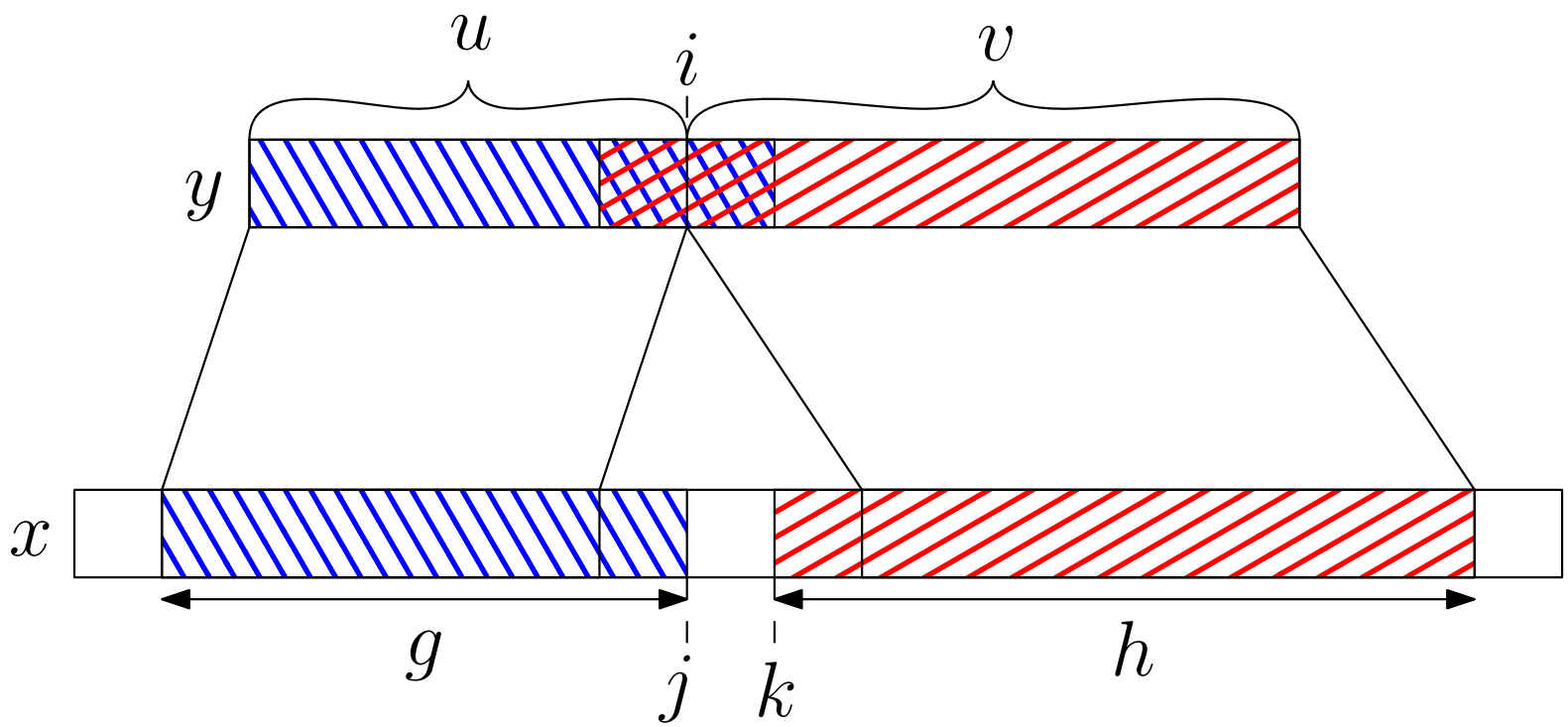
$x[j - g + 1 : j] = y[1 : g]$ and $x[k : k + h - 1] = y[m - h + 1 : m]$

Decidability (continued)

Theorem

Given two strings x and y, we can determine whether or not $x \overset{sdd}{\leftarrow} y \neq \emptyset$ in $\mathcal{O}(n)$ time, where $|x| = n, |y| = m$ and $m \leq n$.

Construct two Aho-Corasick automata $\mathcal{A}_P$ ($\mathcal{A}_S$) with all prefixes of $y(y^R)$, and run them on x.



Decidability (continued)

Theorem

Given two strings x and y, we can determine whether or not $x \overset{sdd}{\leftarrow} y \neq \emptyset$ in $\mathcal{O}(n)$ time, where $|x| = n, |y| = m$ and $m \leq n$.

Construct two Aho-Corasick automata $\mathcal{A}_P$ ($\mathcal{A}_S$) with all prefixes of $y(y^R)$, and run them on x.

1. $y[1 : g] = x[j - g + 1 : j]$
2. $y[m - h + 1 : m] = x[k : k + h - 1]$
3. $j - g + i \leq k + h - m + i$
4. $i \leq g$
5. $m - i \leq h$

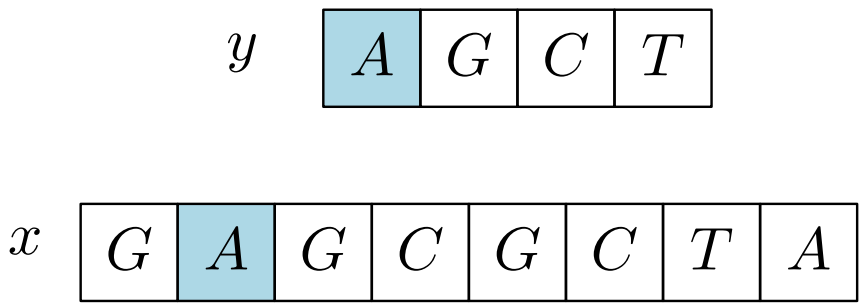
Decidability (continued)

Theorem

Given two strings x and y, we can determine whether or not $x \overset{sdd}{\leftarrow} y \neq \emptyset$ in $\mathcal{O}(n)$ time, where $|x| = n, |y| = m$ and $m \leq n$.

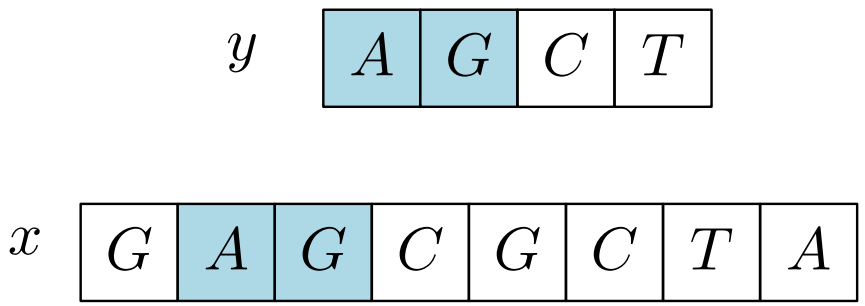
1. Construct two arrays $j[g]$ and $k[h]$
   - $j[g]$ keeps the smallest j for g
   - $k[h]$ keeps the largest k for h
2. For two arrays $j[g]$ and $k[h]$,
   - Check the condition $m \leq (g - j) + (k + h)$
   - Check the condition $j[g] \leq k[h] + 1$
   - if both conditions satisfy, $x \overset{sdd}{\leftarrow} y \neq \emptyset$
   - otherwise, increase g and decrease h by 1

Decidability (continued)



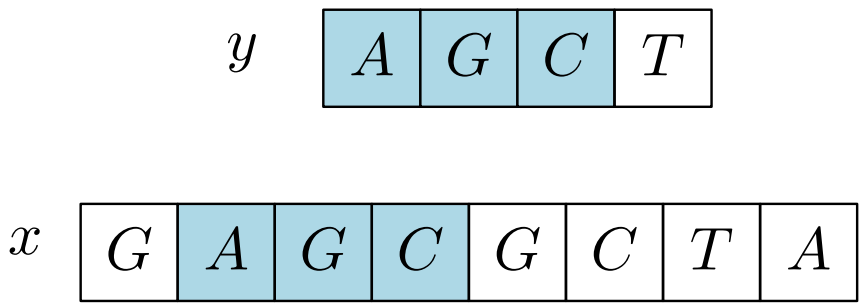
(j, g)
$(2, 1)$
$(3, 2)$
$(4, 3)$
$(-, 4)$

Decidability (continued)



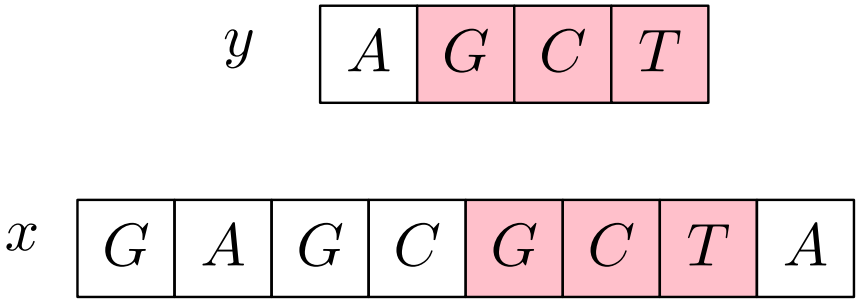
(j, g)
$(2, 1)$
$(3, 2)$
$(4, 3)$
$(-, 4)$

Decidability (continued)



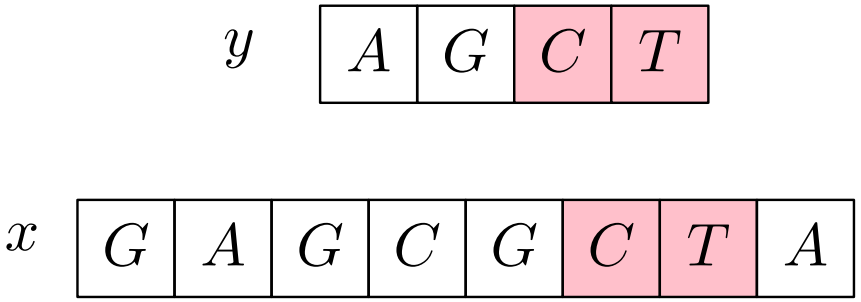
(j, g)
$(2, 1)$
$(3, 2)$
$(4, 3)$
$(-, 4)$

Decidability (continued)



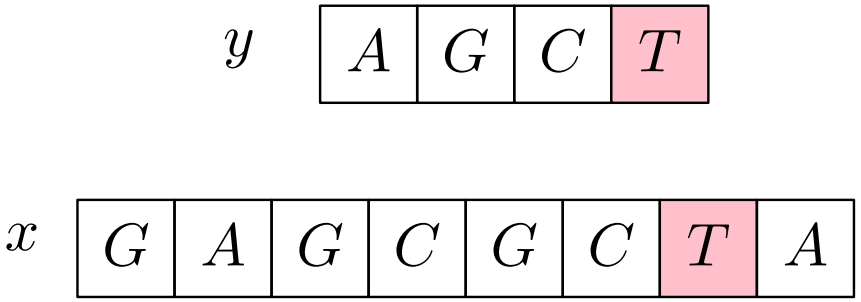
(j, g)	(k, h)
$(2, 1)$	$(-, 4)$
$(3, 2)$	$(5, 3)$
$(4, 3)$	$(6, 2)$
$(-, 4)$	$(7, 1)$

Decidability (continued)



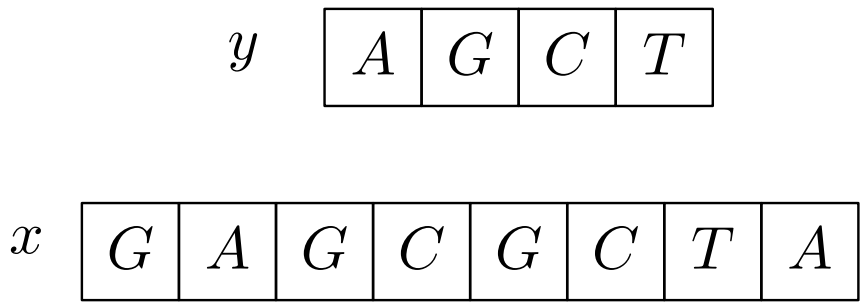
(j, g)	(k, h)
$(2, 1)$	$(-, 4)$
$(3, 2)$	$(5, 3)$
$(4, 3)$	$(6, 2)$
$(-, 4)$	$(7, 1)$

Decidability (continued)



(j, g)	(k, h)
$(2, 1)$	$(-, 4)$
$(3, 2)$	$(5, 3)$
$(4, 3)$	$(6, 2)$
$(-, 4)$	$(7, 1)$

Decidability (continued)



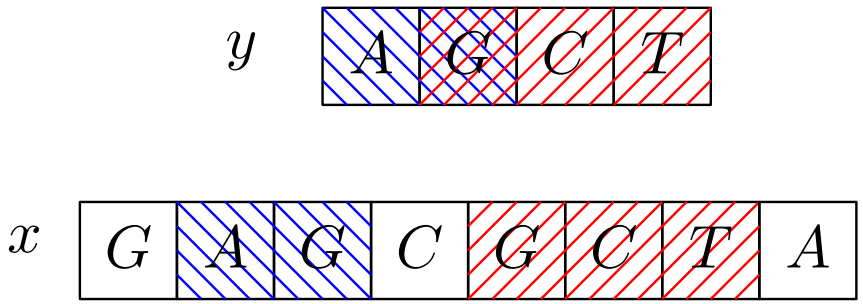
start from
$g = 1$ and $h = m$

g	$j[g]$	h	$k[h]$
1	2	4	0
2	3	3	5
3	4	2	6
4	0	1	7

$j[g] \leq k[h] + 1$?

$2 > 0 + 1$

Decidability (continued)



$g = 2$ and $h = 3$

g	$j[g]$	h	$k[h]$
1	2	4	0
2	3	3	5
3	4	2	6
4	0	1	7

$j[g] \leq k[h] + 1?$

$3 \leq 5 + 1$

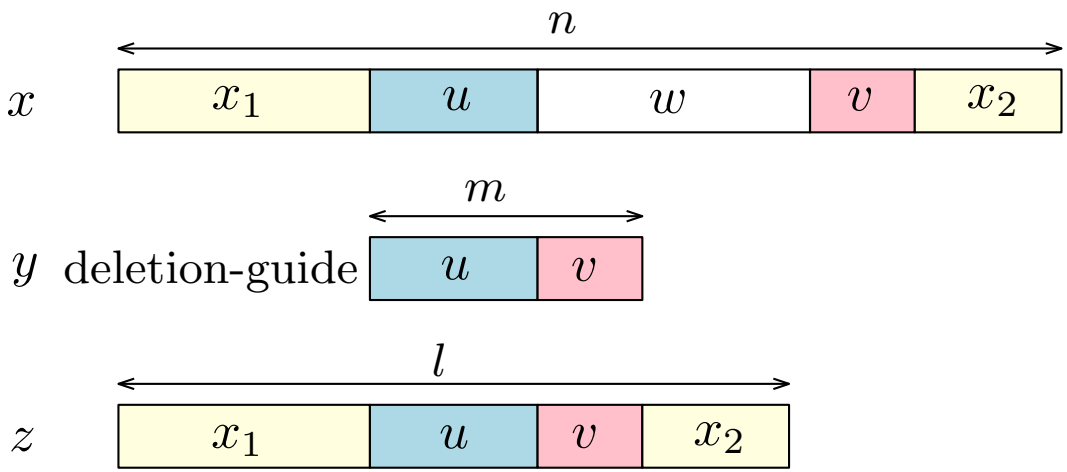
□

Decidability

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.

Suppose that there exist x, y, and z such that $z \in x \overset{sdd}{\leftarrow} y$.

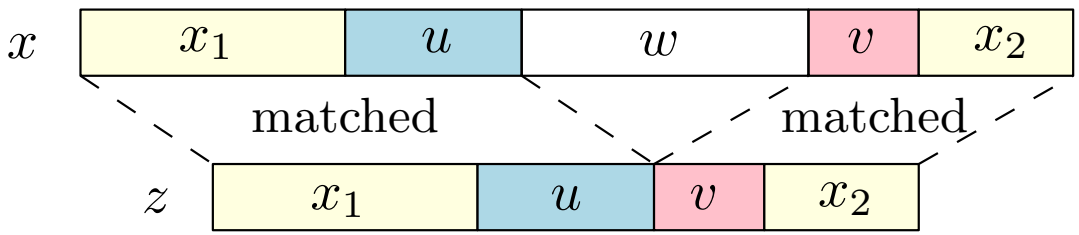


Decidability (continued)

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.

Scan both ends of x and z until a mismatch occurs.

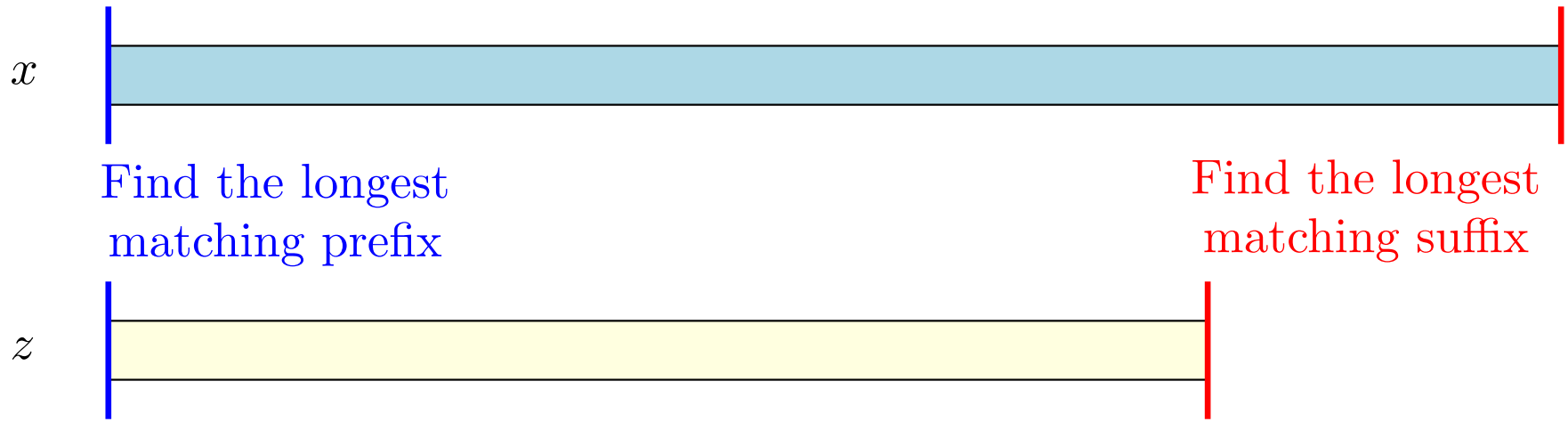


Decidability (continued)

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.

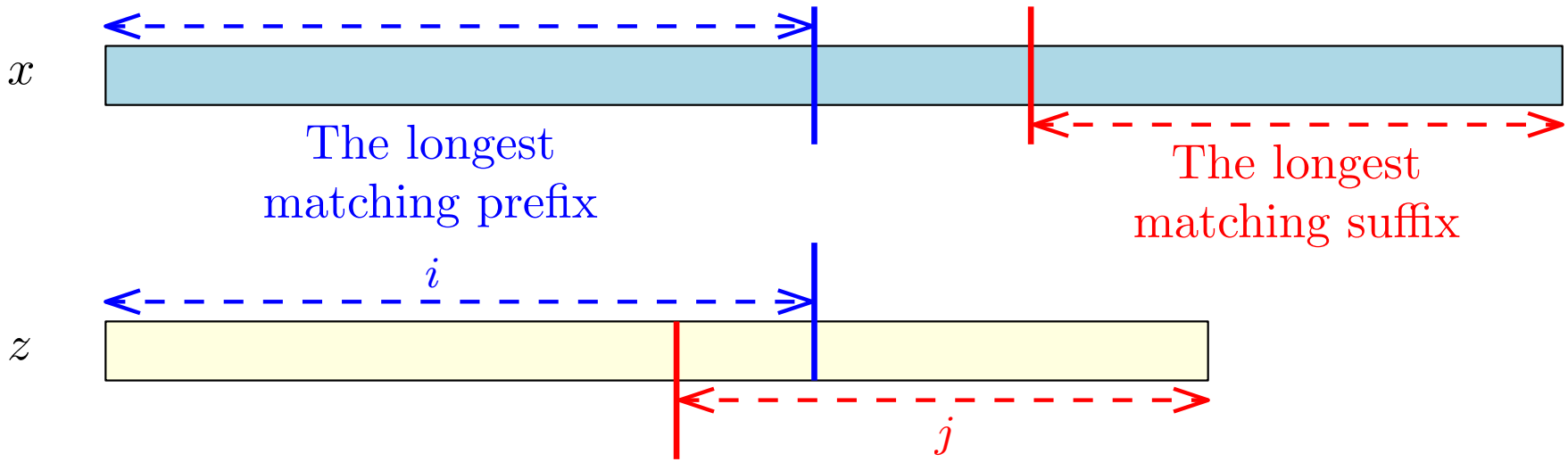
Scan both ends of x and z until a mismatch occurs.



Decidability (continued)

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.



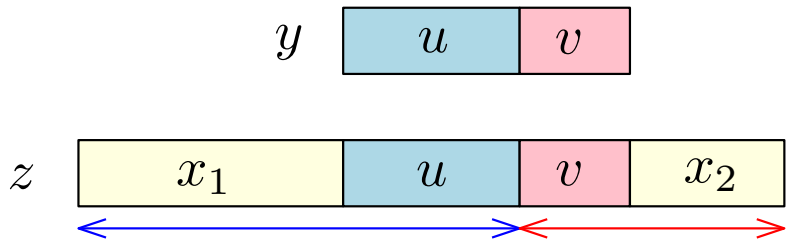
$x[1 : i] = z[1 : i]$ and $x[n-j+1 : n] = z[l-j+1 : l]$

Decidability (continued)

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.

We check whether or not $y = uv$ is a substring of z.

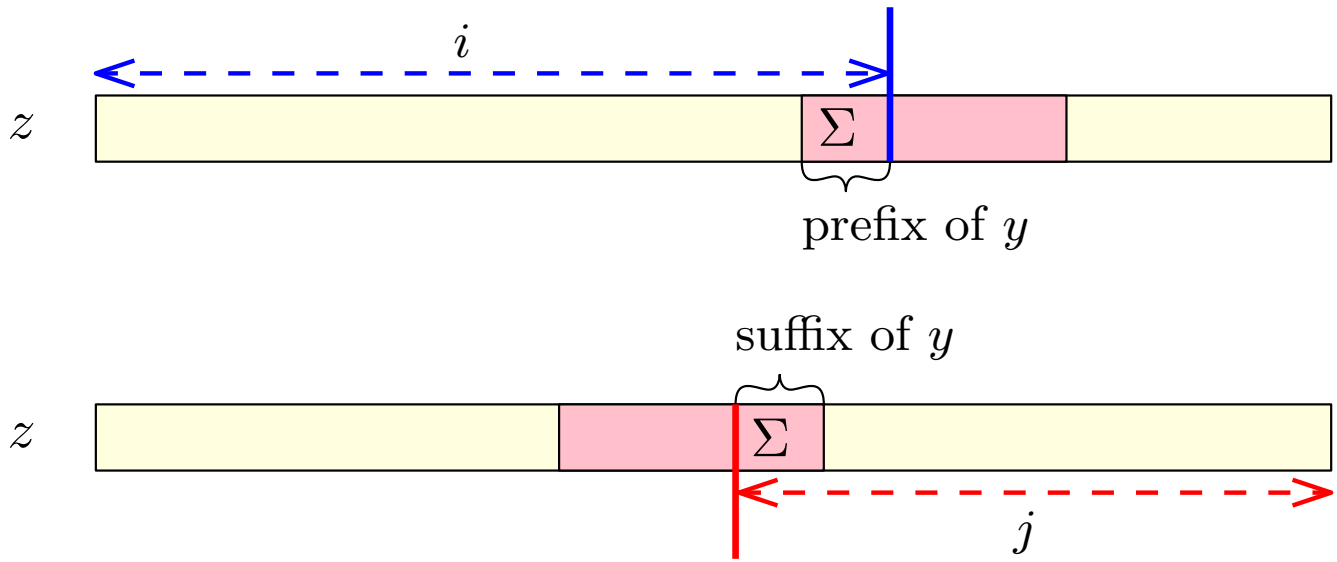


Decidability (continued)

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.

A prefix of y should be a suffix of the longest matching prefix of z,
A suffix of y should be a prefix of the longest matching suffix of z.

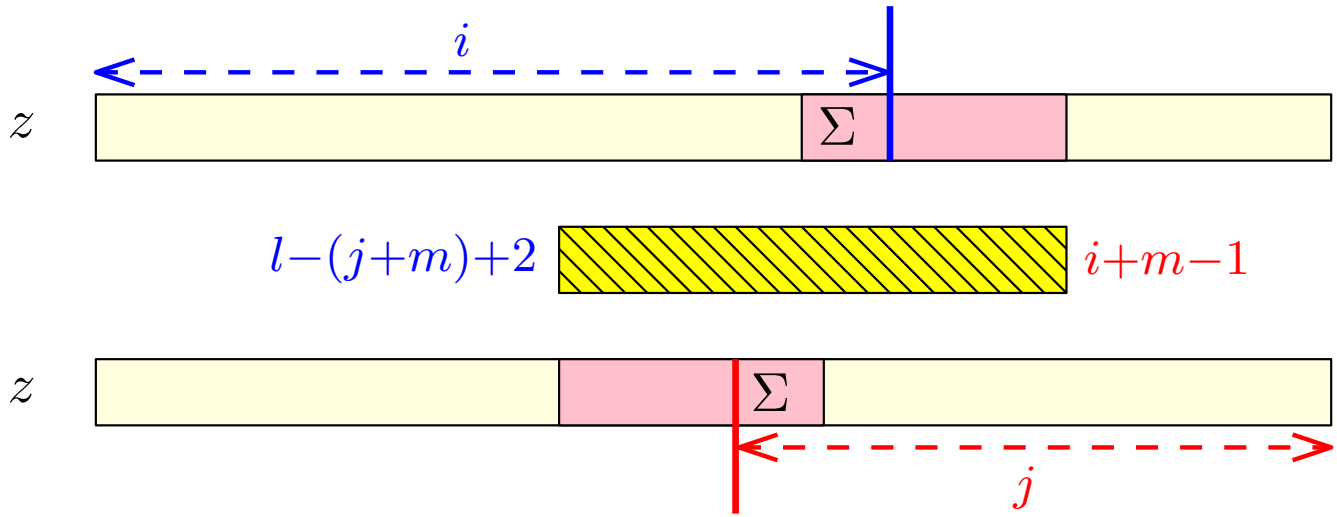


Decidability (continued)

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.

We check for an occurrence of y within $z[l-(j+m)+2 : i+m-1]$.

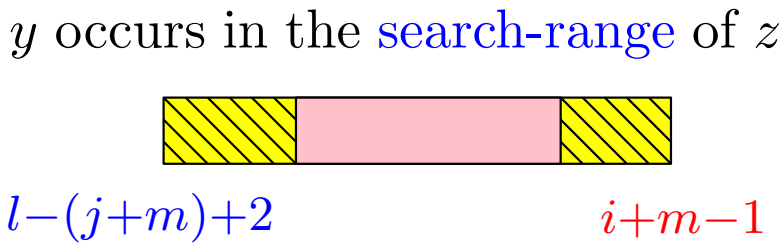


Decidability (continued)

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.

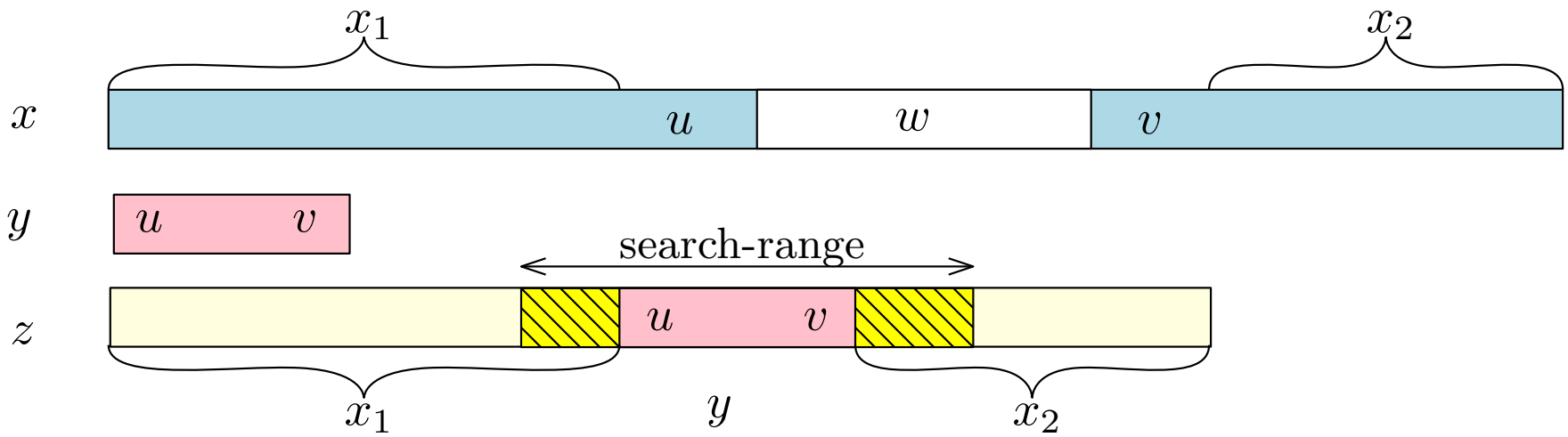
KMP pattern matching algorithm returns 1 if y occurs in the search-range.



Decidability (continued)

Theorem

Given three strings $x, y, z \in \Sigma^*$, we can determine whether of not $z \in x \overset{sdd}{\leftarrow} y$ in $\mathcal{O}(n)$ time, where $|x| = n$ and $|x| \geq |z| \geq |y| \geq 2$.



Outline

1. Introduction
   - Background
   - Definition
   - Problems
2. Main Results
   - Closure Properties of Site-Directed Deletion
   - Decidability of Site-Directed Deletion
3. Conclusion
   - Summary
   - Future Works

Summary

The site-directed deletion operation on strings:

$$x \overset{sdd}{\leftarrow} y = \{x_1 uvx_2 \mid x = x_1 uwvx_2, y = uv, \text{ and } u, v \neq \lambda\}$$

- Regular languages are closed under the site-directed deletion operation whereas context-free languages are not.
- $\mathbb{SDD}^*(L)$ is not context-free when L is context-free.
- We can decide whether or not $x \overset{sdd}{\leftarrow} y \neq \emptyset$ and $z \in x \overset{sdd}{\leftarrow} y$ in linear time.

Outline

1. Introduction
   - Background
   - Definition
   - Problems
2. Main Results
   - Closure Properties of Site-Directed Deletion
   - Decidability of Site-Directed Deletion
3. Conclusion
   - Summary
   - Future Works

Open Problems and Future Works

- Closure property of regular languages under $\mathbb{SDD}^*$ remains open:
  - Given a regular language L, is $\mathbb{SDD}^*(L)$ is regular?

- Research for variants of site-directed deletion:
  - The maximal/minimal site-directed deletion of strings
  - Closure property, decidability, language equations with respect to maximal/minimal site-directed deletions

Thank you for your attention!