

# Substring Range Reporting

---

Philip Bille  
Inge Li Gørtz

# Outline

---

- Problem definition
- Results
  - 2D range reporting approach
  - New solution
- Applications
  - Position-restricted substring searching
  - Indexing substrings with intervals
  - Indexing substrings with gaps
- Remarks and Open Problems

# Classic String Indexing

---

$S = \text{senselessness}$

- Preprocess string  $S$  of length  $n$ .
- Report( $P$ ): Given pattern  $P$  of length  $m$ , report all occurrences of  $P$  in  $S$ .
- Suffix tree + perfect hashing:  $O(n)$  space and  $O(m + \text{occ})$  query time.

# Substring Range Reporting

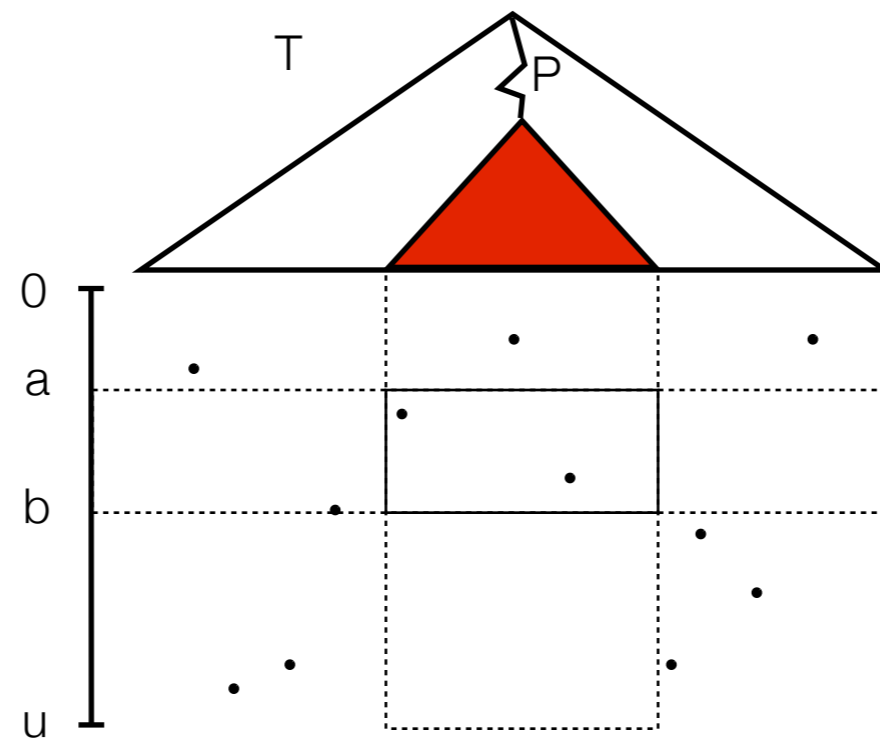
---

S = senselessness  
1 7 3 2 1 1 5 3 2 9 0 5 6

- Preprocess S. Each position in S has an *integer label* in  $[0,u]$ .
- Report(P, a,b): Report all occurrences of P whose startpos label is in  $[a,b]$

# 2DRR and SRR

---



- Build suffix tree  $T$  for  $S + 2DRR$  data structure over leaves of  $T$  and  $[0, u]$ .
- Suffix  $i$  represented by  $(\text{lex-order}(i), \text{label}(i))$  in 2DRR.
- $\text{Report}(P, a, b)$ : Search for  $P$  in  $T$ . Do 2DRR query with interval of leaves and  $[a, b]$ .

# Results

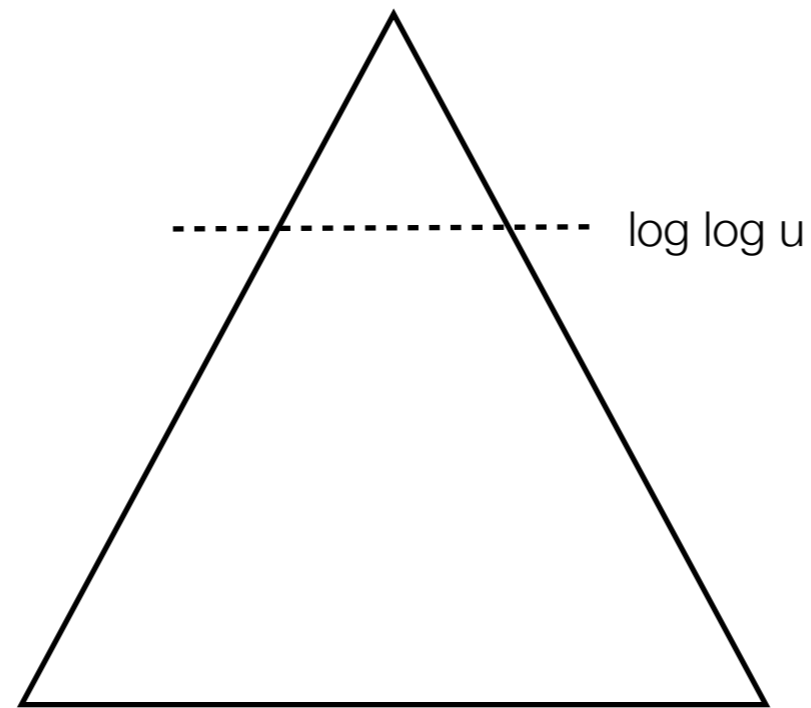
---

Space	Time	Reference
$O(n \log^\epsilon n)$	$O(m + \log \log u + \text{occ})$	[MN2006]
$O(nu^\epsilon)$	$O(m + \text{occ})$	[CIKR2008]
$O(n \log^\epsilon n + n \log \log u)$	$O(m + \text{occ})$	<b>This paper</b>

- Also many succinct versions.
- In all our applications  $u = O(n) \Rightarrow$  space is  $O(n \log^\epsilon n)$
- Not based on 2DRR approach: Any 2DRR data structure with  $O(n \log^{O(1)} n)$  space must use  $\Omega(\log \log u)$  time [PT2006].

# The Data Structure

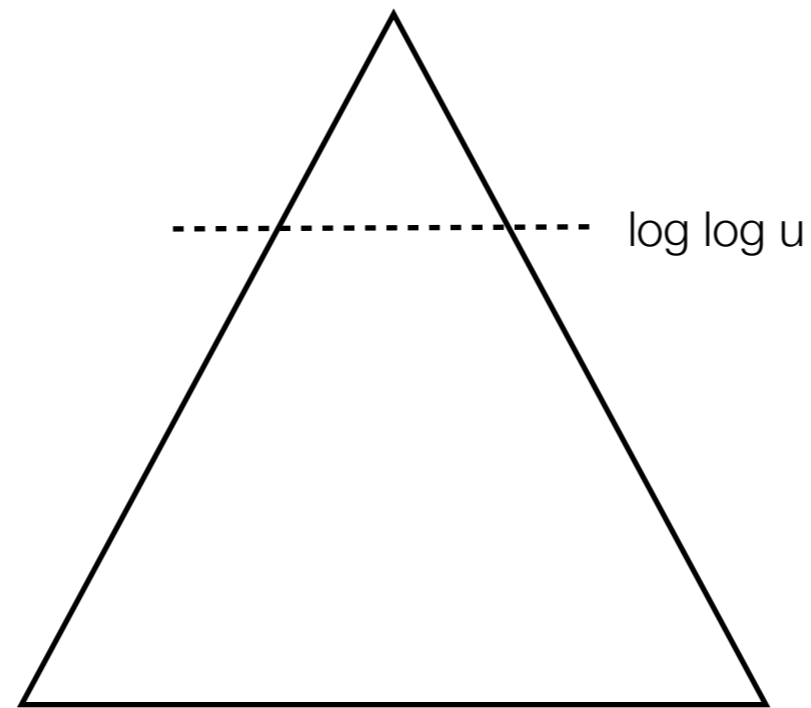
---



- Split suffix tree into *top-tree* and *bottom-trees* at depth  $\log \log u$
- Two cases depending on if search for  $P$  ends in a top-tree or a bottom-tree

# The Data Structure

---

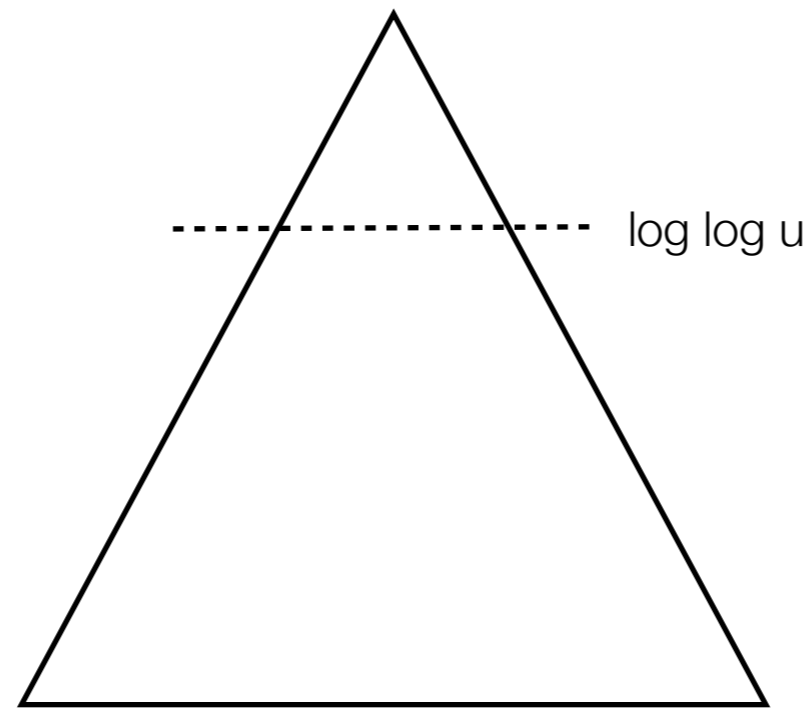


- Case 1:  $m > \log \log u$  (search ends in a bottom-tree)
- Use 2DRR as before
- Space:  $O(n \log^\epsilon n)$
- Time:  $O(m + \log \log u + \text{occ}) = O(m + \text{occ})$



# The Data Structure

---



- Case 2:  $m \leq \log \log u$  (search ends in top-tree)
- For each node  $v$  in top-tree store all descendant leaf labels in 1DRR. With [ABR2001] each 1DRR use linear space and optimal query time. Search with  $[a,b]$  in 1DRR.
- Time:  $O(m + \text{occ})$
- Space:  $O(\text{Number of descendant leaves of all nodes in top-tree}) = O(n \log \log u)$

# The Data Structure

---

- Combining both cases:
- Total space:  $O(n \log^\epsilon n + n \log \log u)$
- Total time:  $O(m + \text{occ})$
  
- Basic idea related to *filtering search* [Cha1986].

# Outline

---

- Problem definition
- Results
  - 2D range reporting approach
  - New solution
- **Applications**
  - Position-restricted substring searching
  - Indexing substrings with intervals
  - Indexing substrings with gaps
- Remarks and Open Problems

# Position-restricted Substring Searching

---

$S = \text{senselessness}$

- Preprocess string  $S$ .
- $\text{Report}(P, a, b)$ : Report occurrences of  $P$  that start at position in  $[a, b]$
- Special case of SRR where  $\text{label}(i) = i$ .
- Note  $u = n$ .

# Results

---

Space	Time	Reference
$O(n \log^\epsilon n)$	$O(m + \log \log n + \text{occ})$	[MN2006]
$O(n^{1+\epsilon})$	$O(m + \text{occ})$	[CIKR2008]
$O(n \log^\epsilon n)$	$O(m + \text{occ})$	<b>This paper</b>

# Indexing Substrings with Intervals

---

$S = \text{senselessness}$

- Preprocess string  $S$  and set of intervals  $\pi$  in  $S$ .
- $\text{Report}(P, a, b)$ : Report occurrences of  $P$  that start at position in  $[a,b]$  and within  $\pi$ .
- Reduction to SRR:  $\text{label}(i) = i$  if  $i$  is covered by  $\pi$  and 0 otherwise.

# Results

---

Space	Time	Reference
$O(n \log^2 n)$	$O(m + \log \log n + \text{occ})$	[CIKRW2010]
$O(n^{1+\epsilon})$	$O(m + \text{occ})$	[CIKR2008]
$O(n \log^\epsilon n)$	$O(m + \text{occ})$	<b>This paper</b>

# Indexing Substrings with Gaps

---

S = senselessness

- Preprocess string S with parameter  $d$  = size of gaps
- Report( $P_1, P_2$ ): Report occurrences of  $P_1 \star^d P_2$



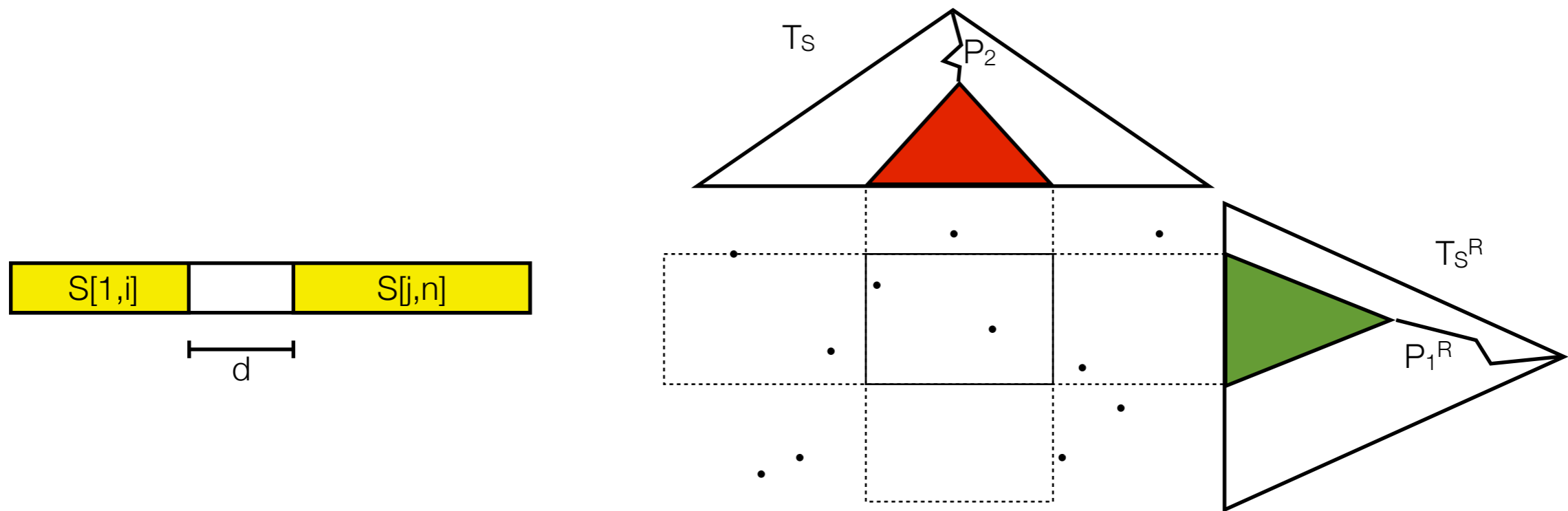
# Results

---

Space	Time	Reference
$O(n \log^\epsilon n)$	$O(m + \log \log n + \text{occ})$	[IL2009]
$O(n \log^\epsilon n)$	$O(m + \text{occ})$	<b>This paper</b>

# 2DRR and Gaps

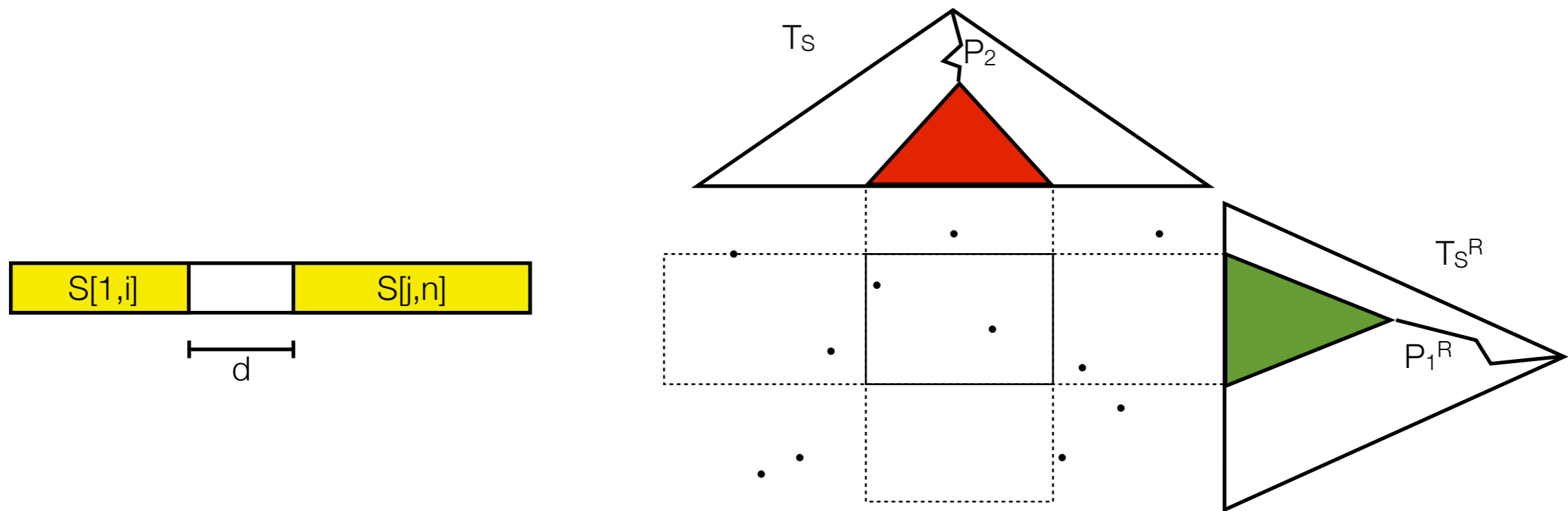
---



- Build suffix trees  $T_S$  and  $T_S^R$  for  $S$  and  $S^R$  + 2DRR data structure over pairs of leaves. ( $T_S$  and  $T_S^R$  stores suffixes and prefixes of  $S$ )
- Point in 2DRR if distance between prefix and suffix is  $d$ .
- Report( $P_1, P_2$ ): Search for  $P_2$  in  $T_S$  and  $P_1^R$  in  $T_S^R$ . Do 2DRR query with intervals of leaves.

# SRR and Gaps

---



- Replace  $T_s$  with SRR for  $S$ .
- Label of suffix  $i = y$ -coordinate from 2DRR data structure.
- Report( $P_1, P_2$ ): Search for  $P_1^R$  in  $T_s^R$  to get range  $[a,b]$ . Do SRR query for  $P_2$  with  $[a,b]$ .

# SRR and Gaps

---

- Total space:  $\text{SRR for } S + T_S^R = O(n \log^\epsilon n) + O(n) = O(n \log^\epsilon n)$
- Total time:  $O(m_1 + m_2 + \text{occ}) = O(m + \text{occ})$

# Remarks and Open Problems

---

- Remarks
  - Basic idea for SRR extends to the numerous (succinct) variants of 2DRR.
- Open problems
  - What other problems does this idea apply to?
  - Are the ideas practical?