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 = senselessness

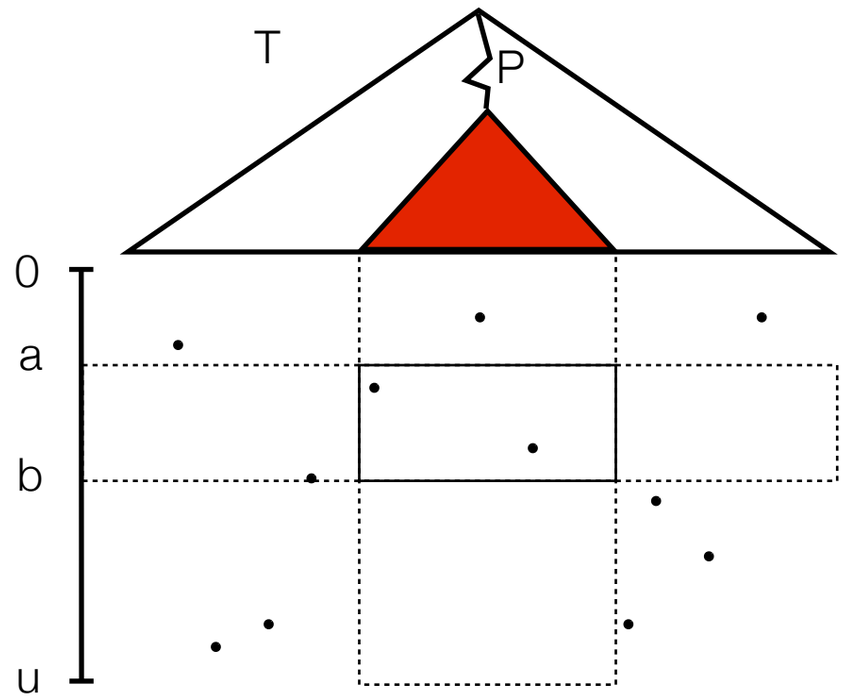
- Preprocess string S of length n.
- Report(P): Given pattern P of length m, report all occurrences of P in S.
- Suffix tree + perfect hashing: O(n) space and O(m + occ) query time.

Substring Range Reporting

S = senselessness
1 7 3 2 11 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



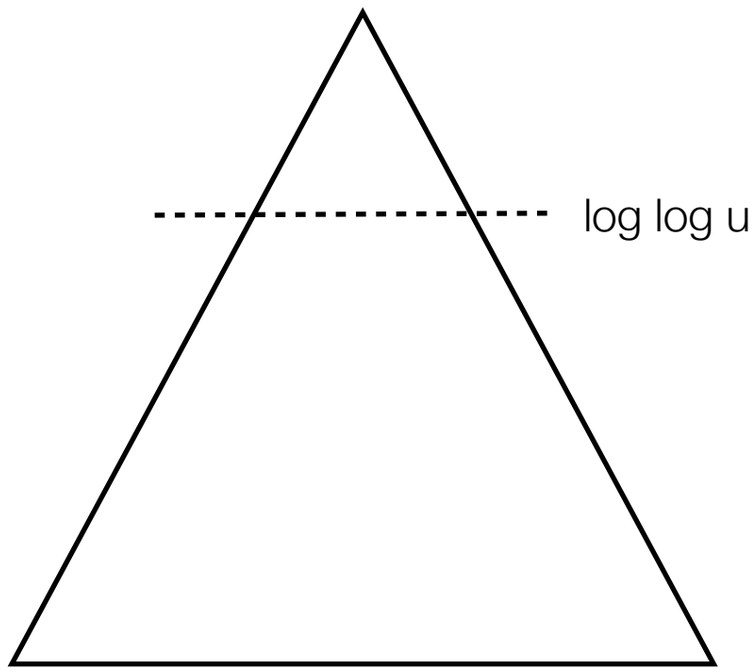
- Build suffix tree T for S + 2DRR data structure over leaves of T and [0,u].
- Suffix i represented by (lex-order(i), label(i)) in 2DRR.
- Report(P, a, b): Search for P in T. Do 2DRR query with interval of leaves and [a,b].

Results

Space	Time	Reference
$O(n\log^{\varepsilon} n)$	$O(m + \log\log u + occ)$	[MN2006]
$O(nu^{\varepsilon})$	$O(m + occ)$	[CIKR2008]
$O(n\log^{\varepsilon} n + n\log\log u)$	$O(m + occ)$	**This paper**

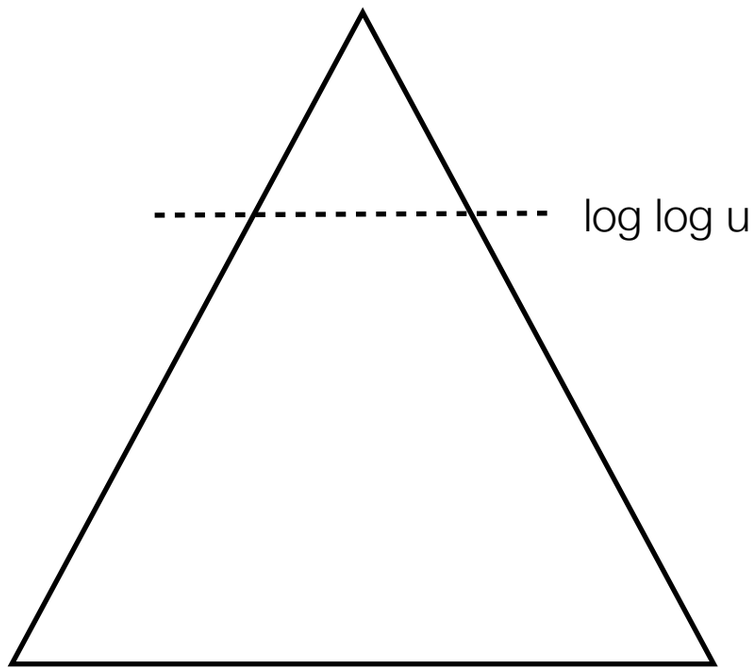
- Also many succinct versions.
- In all our applications $u = O(n)$ => space is $O(n\log^{\varepsilon} n)$
- Not based on 2DRR approach: Any 2DRR data structure with $O(n \log^{O(1)} n)$ space must use $\Omega(\log\log u)$ time [PT2006].

The Data Structure



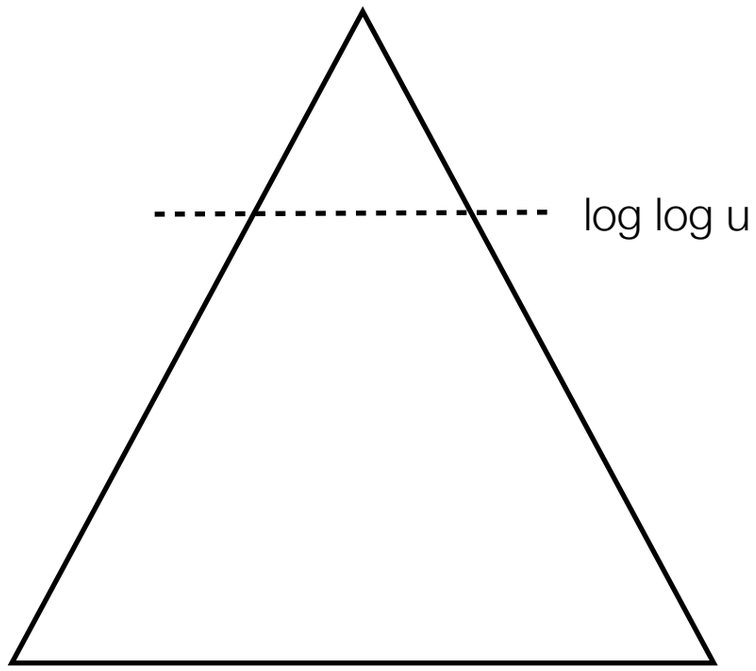
- Split suffix tree into *top-tree* and *bottom-trees* at depth log log u
- Two cases depending on if search for P ends in a top-tree or a bottom-tree

The Data Structure



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

The Data Structure



- Case 2: m ≤ 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 + occ)
- Space: O(Number of descendant leaves of all nodes in top-tree) = O(nloglogu)

The Data Structure

- Combining both cases:
- Total space: $O(n\log^{\varepsilon} n + n \log \log u)$
- Total time: $O(m + occ)$

- 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 = senselessness

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

Results

Space	Time	Reference
$O(n\log^{\varepsilon} n)$	$O(m + \log\log n + occ)$	[MN2006]
$O(n^{1+\varepsilon})$	$O(m + occ)$	[CIKR2008]
$O(n\log^{\varepsilon} n)$	$O(m + occ)$	**This paper**

Indexing Substrings with Intervals

S = senselessness

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

Results

Space	Time	Reference
$O(n\log^2 n)$	$O(m + \log\log n + occ)$	[CIKRW2010]
$O(n^{1+\varepsilon})$	$O(m + occ)$	[CIKR2008]
$O(n\log^{\varepsilon} n)$	$O(m + occ)$	**This paper**

Indexing Substrings with Gaps

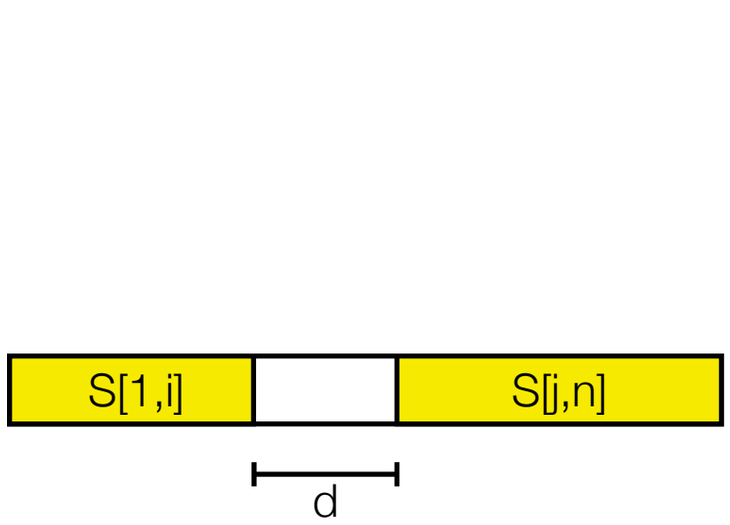
S = senselessness

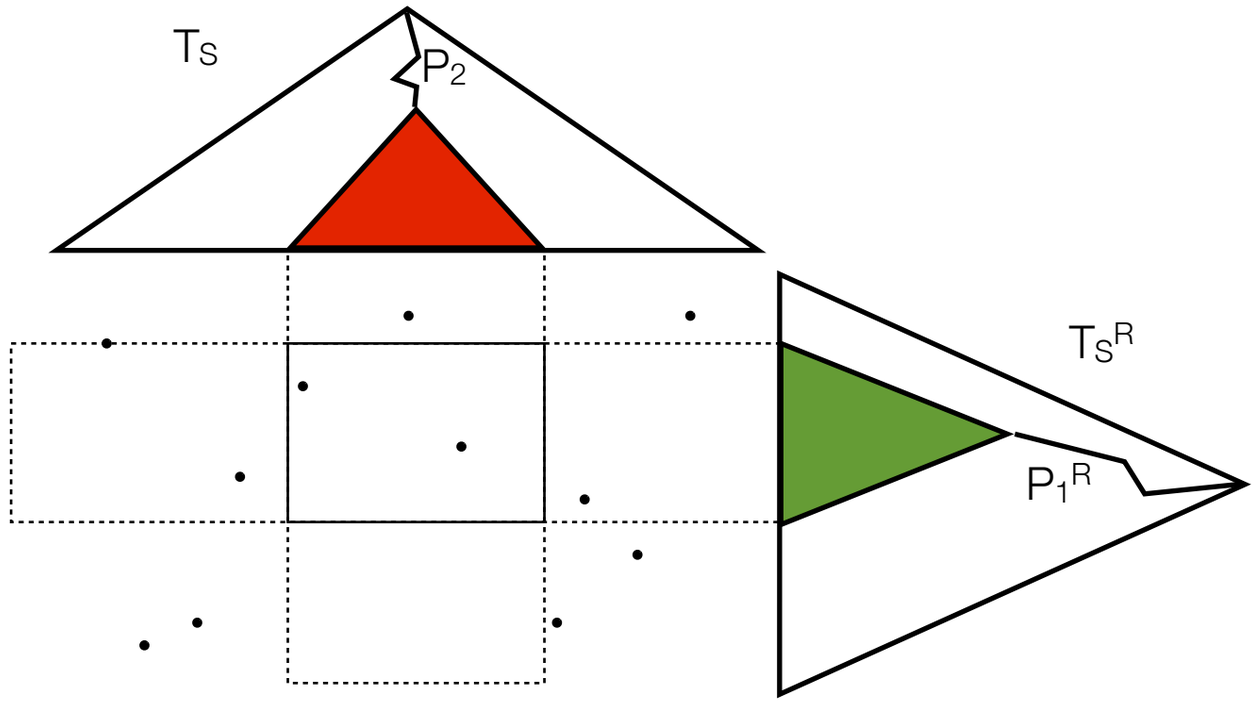
- Preprocess string S with parameter d = size of gaps
- Report(P_1, P_2): Report occurrences of $P_1 \bigstar^d P_2$

Results

Space	Time	Reference
O(nlog$^{\varepsilon}$n)	O(m + log log n + occ)	[IL2009]
O(nlog$^{\varepsilon}$n)	O(m + occ)	**This paper**

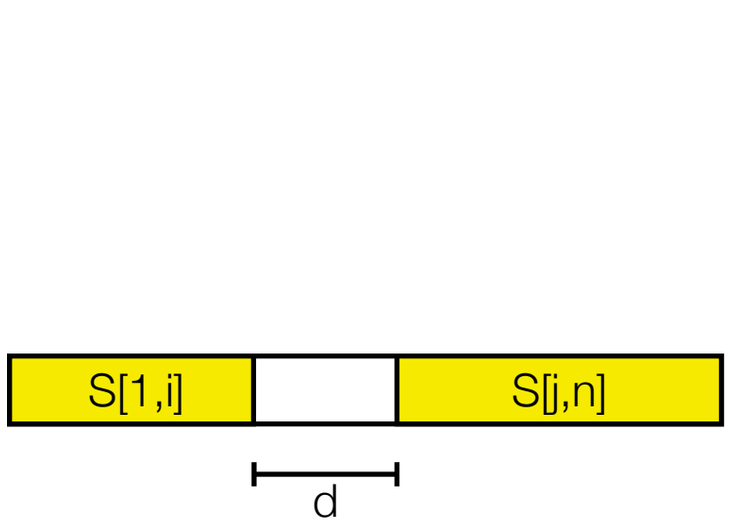
2DRR and Gaps

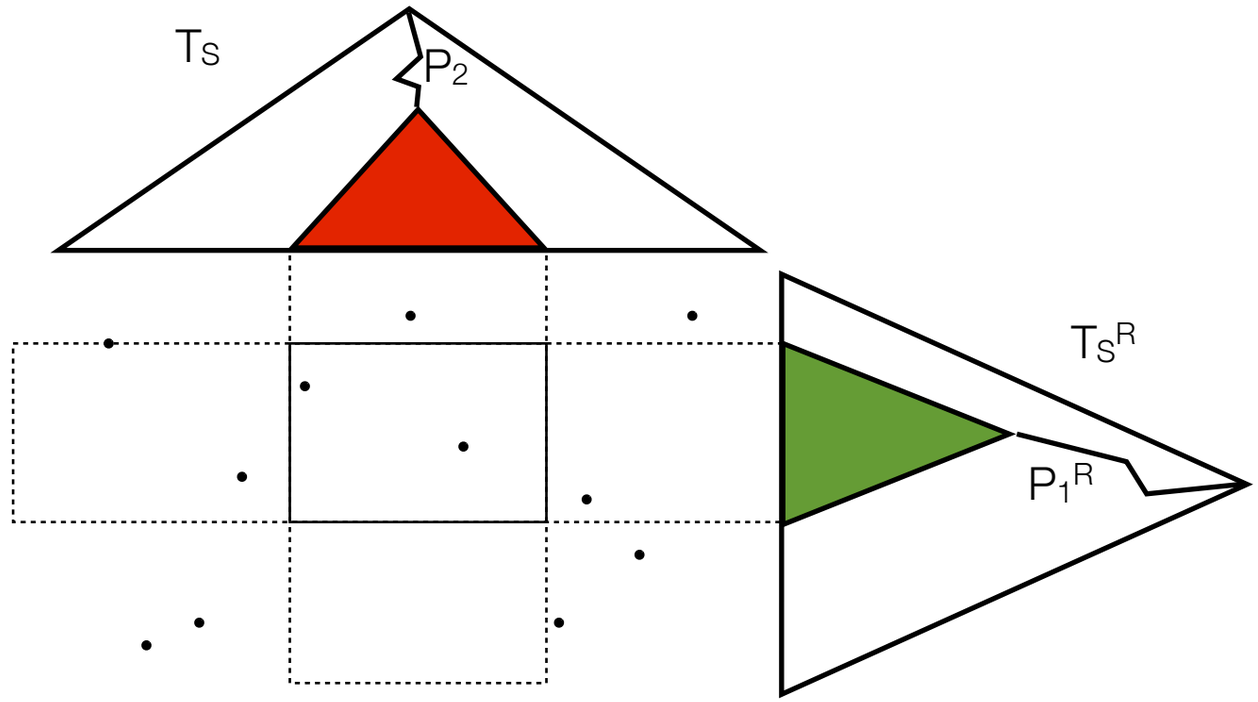




- Build suffix trees T_S and T_S^R for S and S^R + 2DRR data structure over pairs of leaves. (T_S and T_S^R stores suffixes and prefixes of S)
- Point in 2DRR if distance between prefix and suffix is d.
- Report(P_1, P_2): Search for P_2 in T_S and P_1^R in T_S^R. Do 2DRR query with intervals of leaves.

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: SRR for S + T_S^R = O(nlog$^{\varepsilon}$n) + O(n) = O(nlog$^{\varepsilon}$n)
- Total time: O(m_1 + m_2 + occ) = O(m + 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?

The end