

Faster Sliding Window String Indexing in Streams

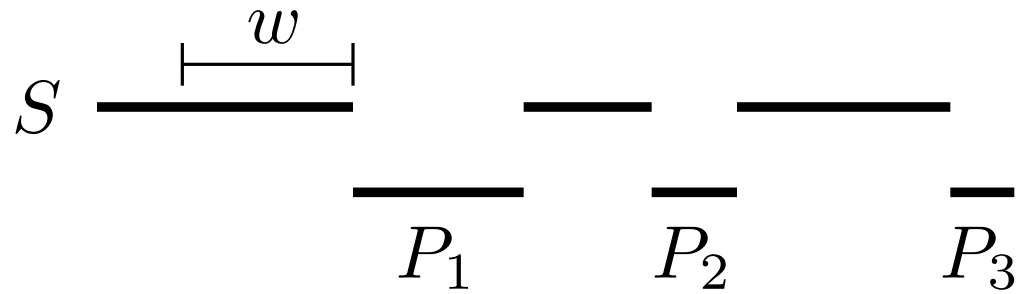
Philip Bille

Paweł Gawrychowski

Inge Li Gørtz

Simon Rumle Tarnow

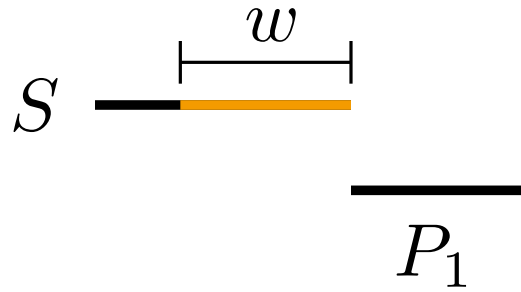
Problem Statement



The streaming sliding window string indexing (SSWSI) problem

- Maintain an index of the w most recent characters of the stream S
- At any point, given a **streamed** pattern string P report all the occurrences in the window
- $O(w)$ space, $O(\log w)$ time per character whp (Bille et al. 2023)

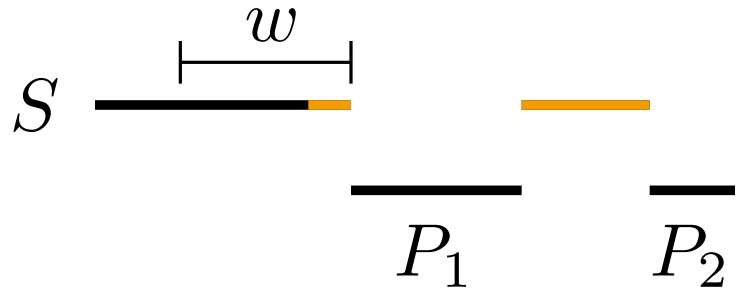
Problem Statement



The streaming sliding window string indexing (SSWSI) problem

- Maintain an index of the w most recent characters of the stream S
- At any point, given a **streamed** pattern string P report all the occurrences in the window
- $O(w)$ space, $O(\log w)$ time per character whp (Bille et al. 2023)

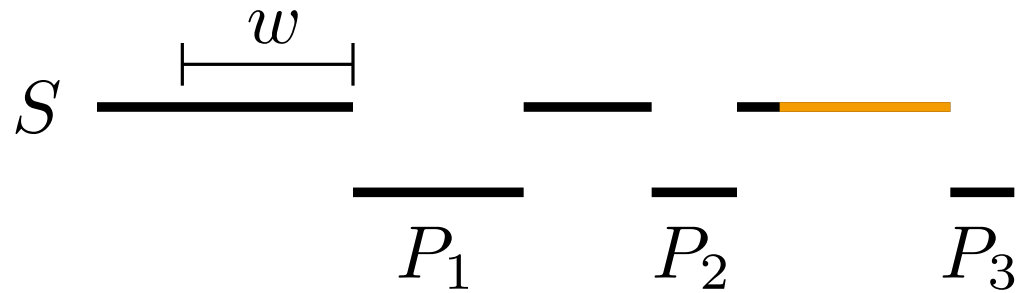
Problem Statement



The streaming sliding window string indexing (SSWSI) problem

- Maintain an index of the w most recent characters of the stream S
- At any point, given a **streamed** pattern string P report all the occurrences in the window
- $O(w)$ space, $O(\log w)$ time per character whp (Bille et al. 2023)

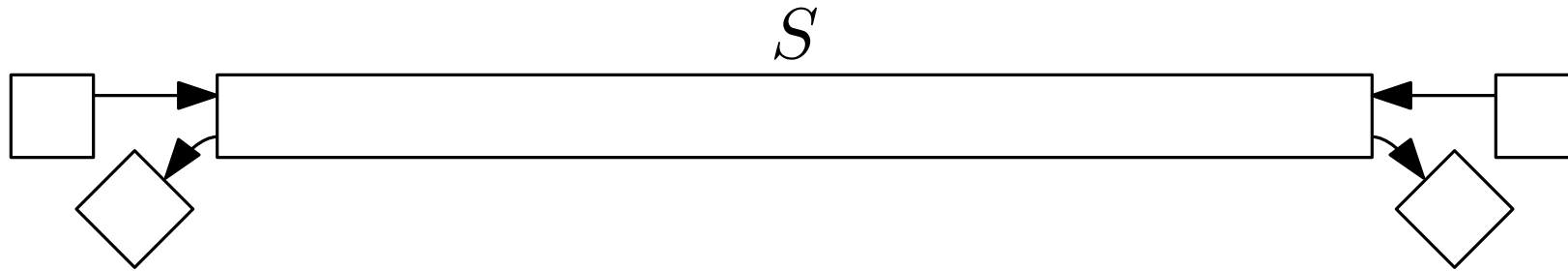
Problem Statement



The streaming sliding window string indexing (SSWSI) problem

- Maintain an index of the w most recent characters of the stream S
- At any point, given a **streamed** pattern string P report all the occurrences in the window
- $O(w)$ space, $O(\log w)$ time per character whp (Bille et al. 2023)

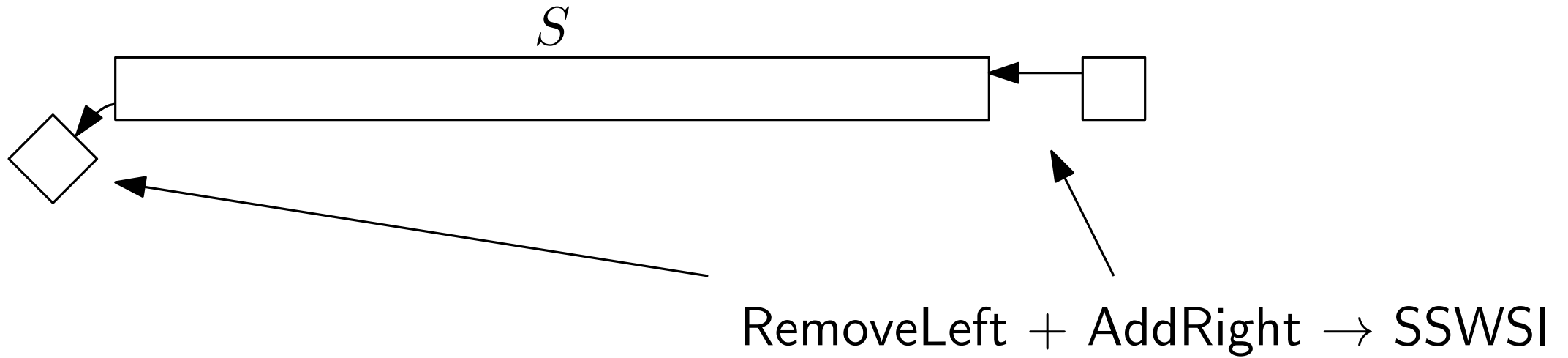
Problem Statement



The streaming dynamic window string indexing (SDWSI) problem

- Maintain an index of the dynamic string S where characters can be removed/added at either end
- At any point, given a **streamed** pattern string P report all the occurrences in S

Problem Statement



The streaming dynamic window string indexing (SDWSI) problem

- Maintain an index of the dynamic string S where characters can be removed/added at either end
- At any point, given a **streamed** pattern string P report all the occurrences in S

Results

	Space	Time per character
SSWSI	$O(w)$	$O(\log \log w + \log \log \sigma)$ whp
SSWSI	$O(w)$	$O\left(\log \log w + \frac{(\log \log \sigma)^2}{\log \log \log \sigma}\right)$
SDWSI	$O(w \cdot (\log \log w + \log \log \sigma))$	$O(\log \log w + \log \log \sigma)$ whp
SDWSI	$O\left(w \cdot \left(\log \log w + \frac{(\log \log \sigma)^2}{\log \log \log \sigma}\right)\right)$	$O\left(\log \log w + \frac{(\log \log \sigma)^2}{\log \log \log \sigma}\right)$

- Alphabet of size σ
- Reporting occurrences uses $O(1)$ time per reported occurrence

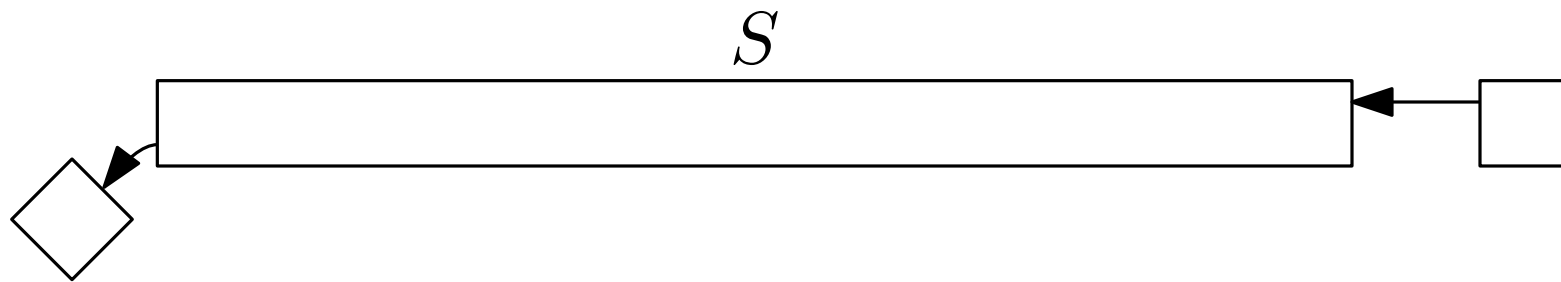
Results

Time complexity of prepend in online suffix tree

	Space	Time per character
SSWSI	$O(w)$	$O(\log \log w + \log \log \sigma)$ whp
SSWSI	$O(w)$	$O\left(\log \log w + \frac{(\log \log \sigma)^2}{\log \log \log \sigma}\right)$
SDWSI	$O(w \cdot (\log \log w + \log \log \sigma))$	$O(\log \log w + \log \log \sigma)$ whp
SDWSI	$O\left(w \cdot \left(\log \log w + \frac{(\log \log \sigma)^2}{\log \log \log \sigma}\right)\right)$	$O\left(\log \log w + \frac{(\log \log \sigma)^2}{\log \log \log \sigma}\right)$

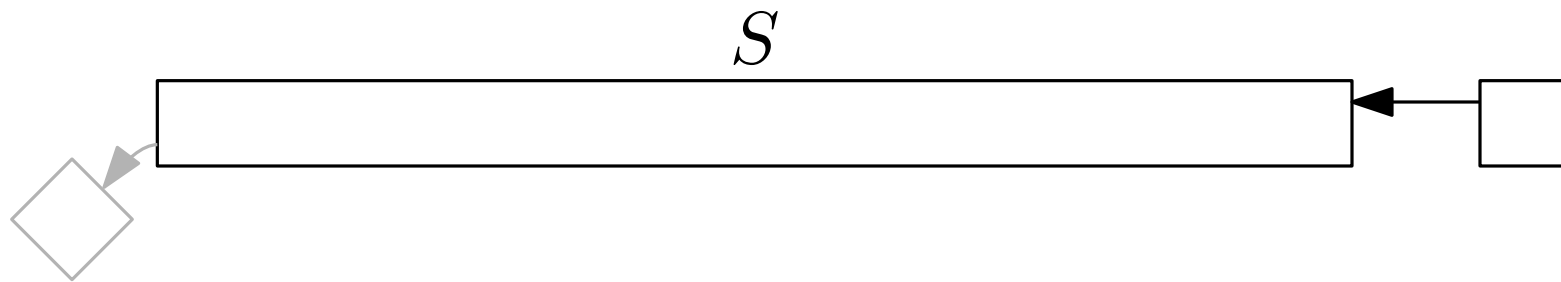
- Alphabet of size σ
- Reporting occurrences uses $O(1)$ time per reported occurrence

Solution Overview



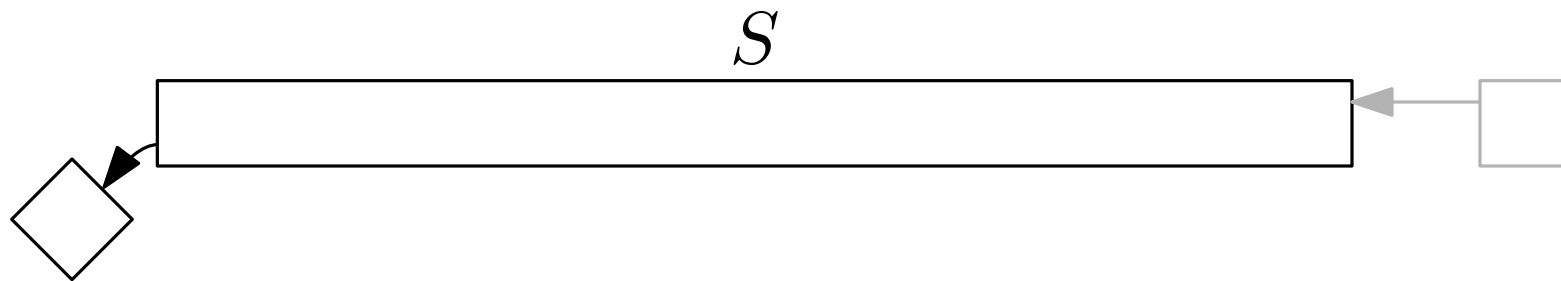
- AddRight only
- RemoveLeft only

Solution Overview



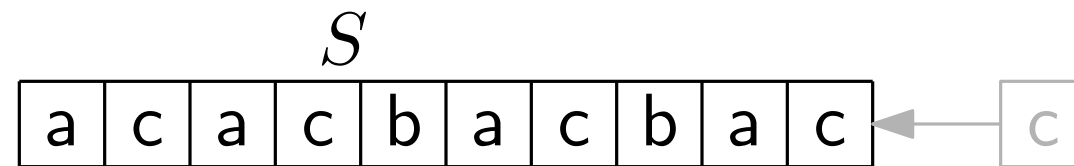
- AddRight only
- RemoveLeft only

Solution Overview

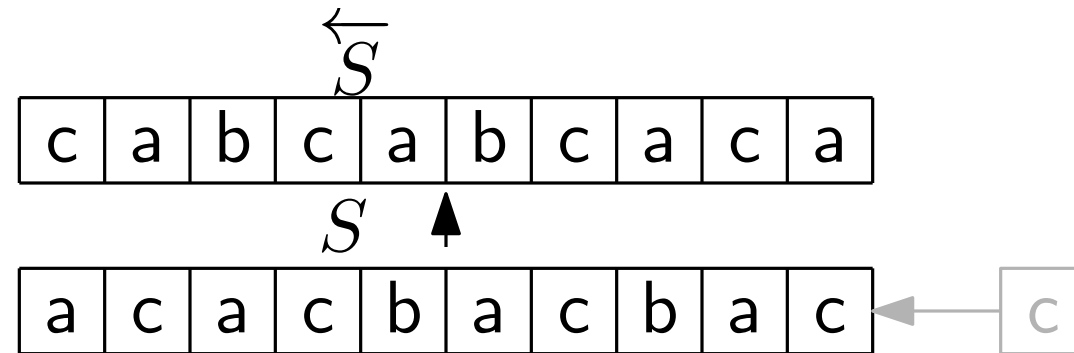


- AddRight only
- RemoveLeft only

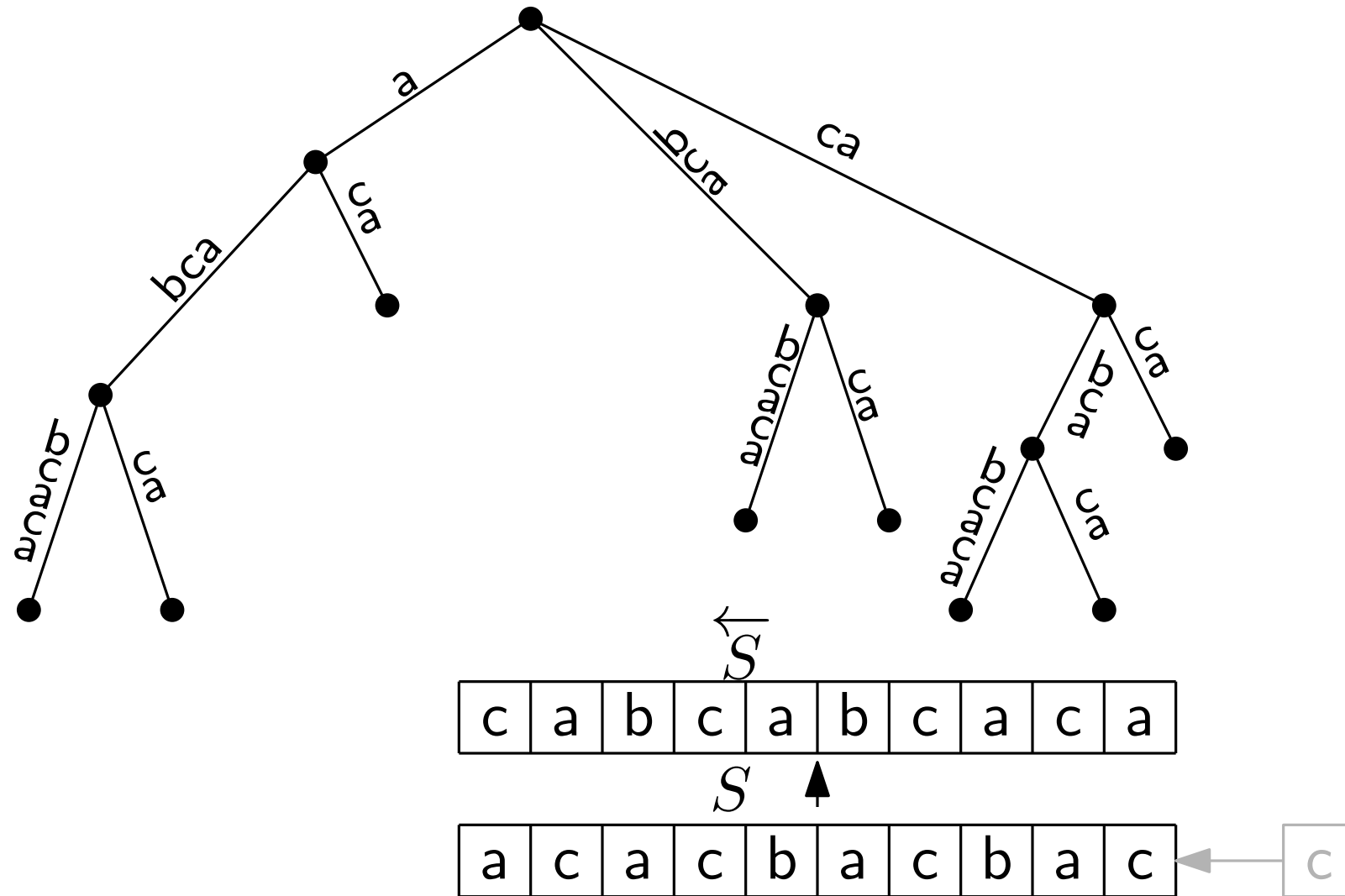
Solution - AddRight



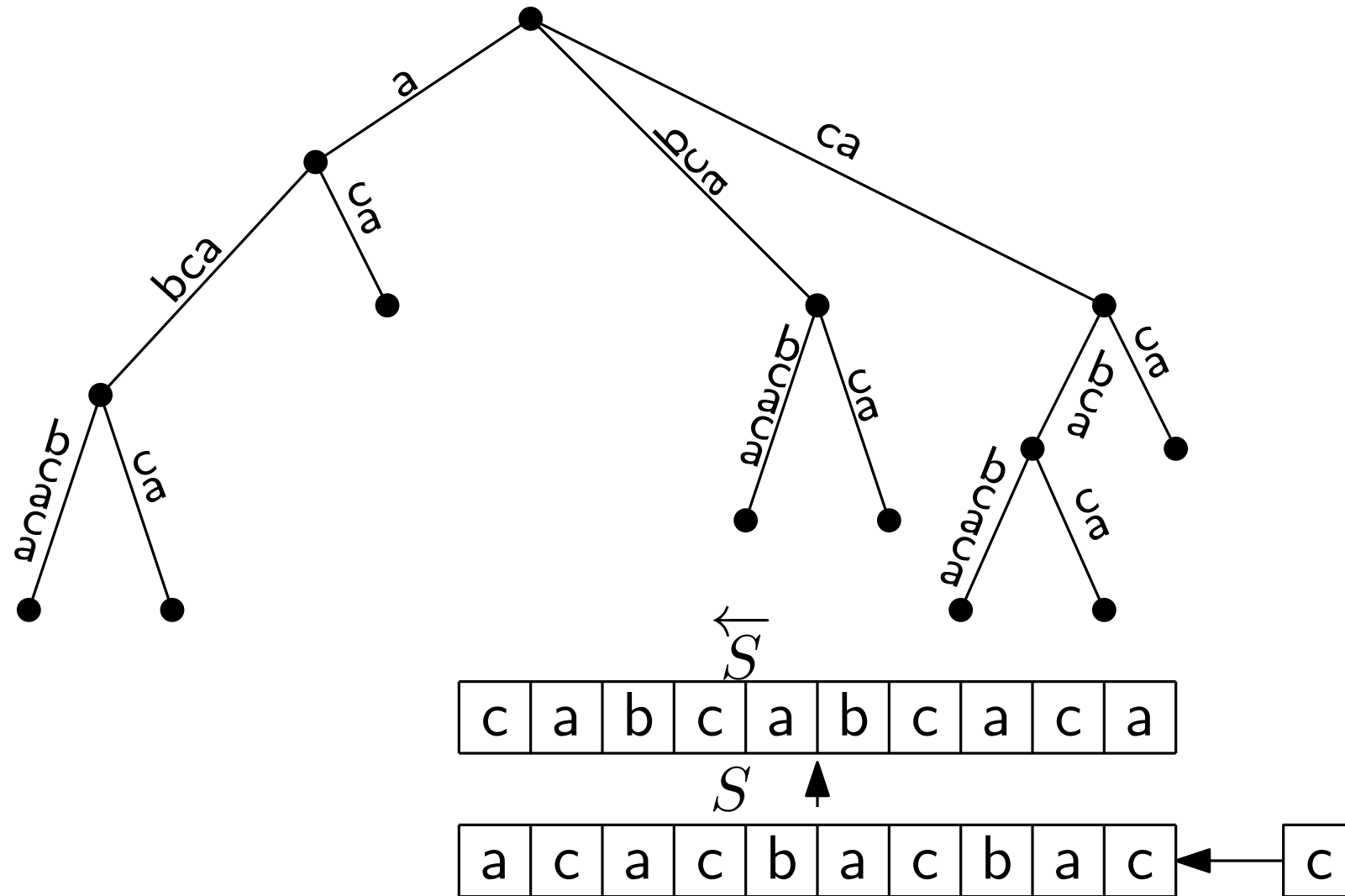
Solution - AddRight



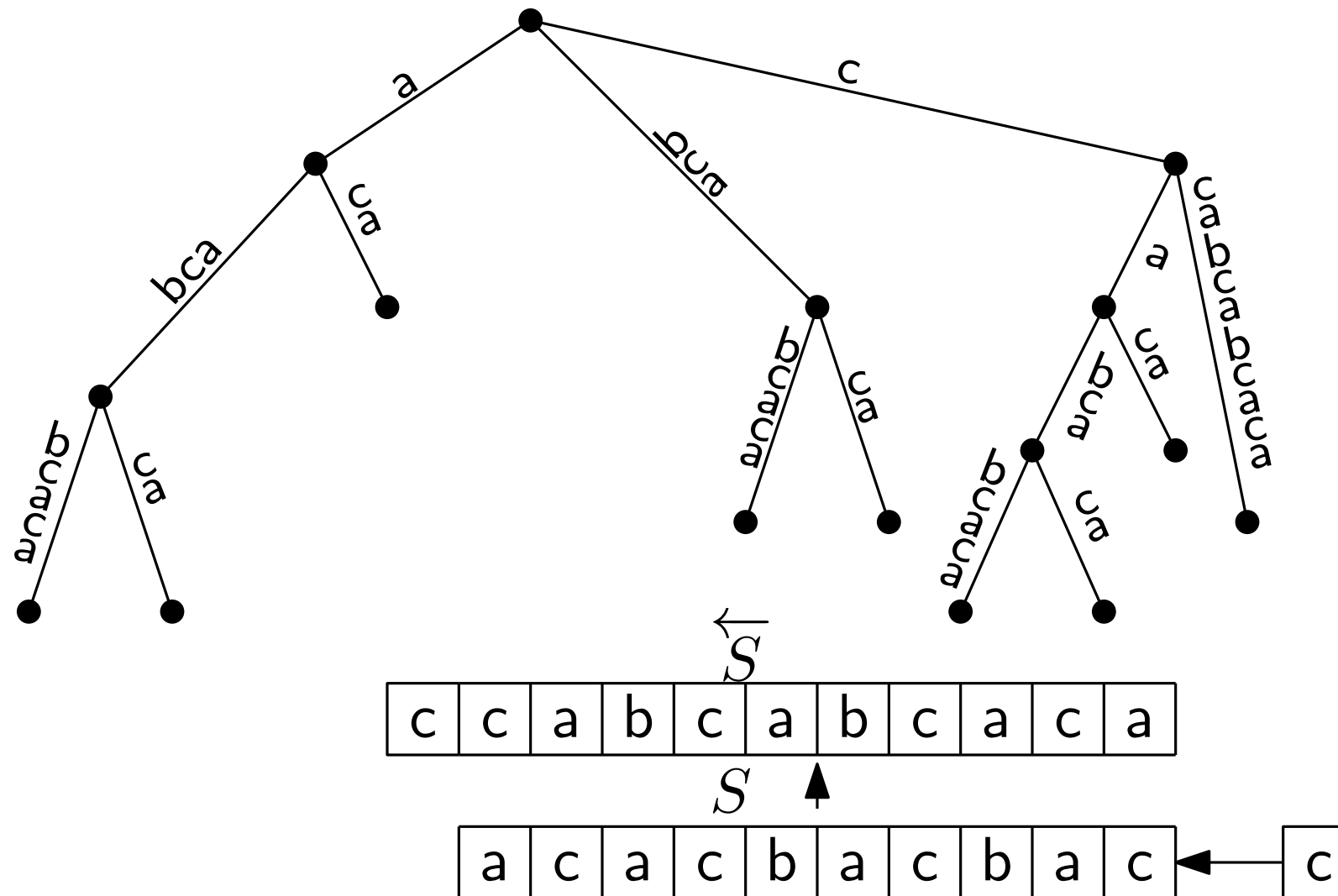
Solution - AddRight



Solution - AddRight

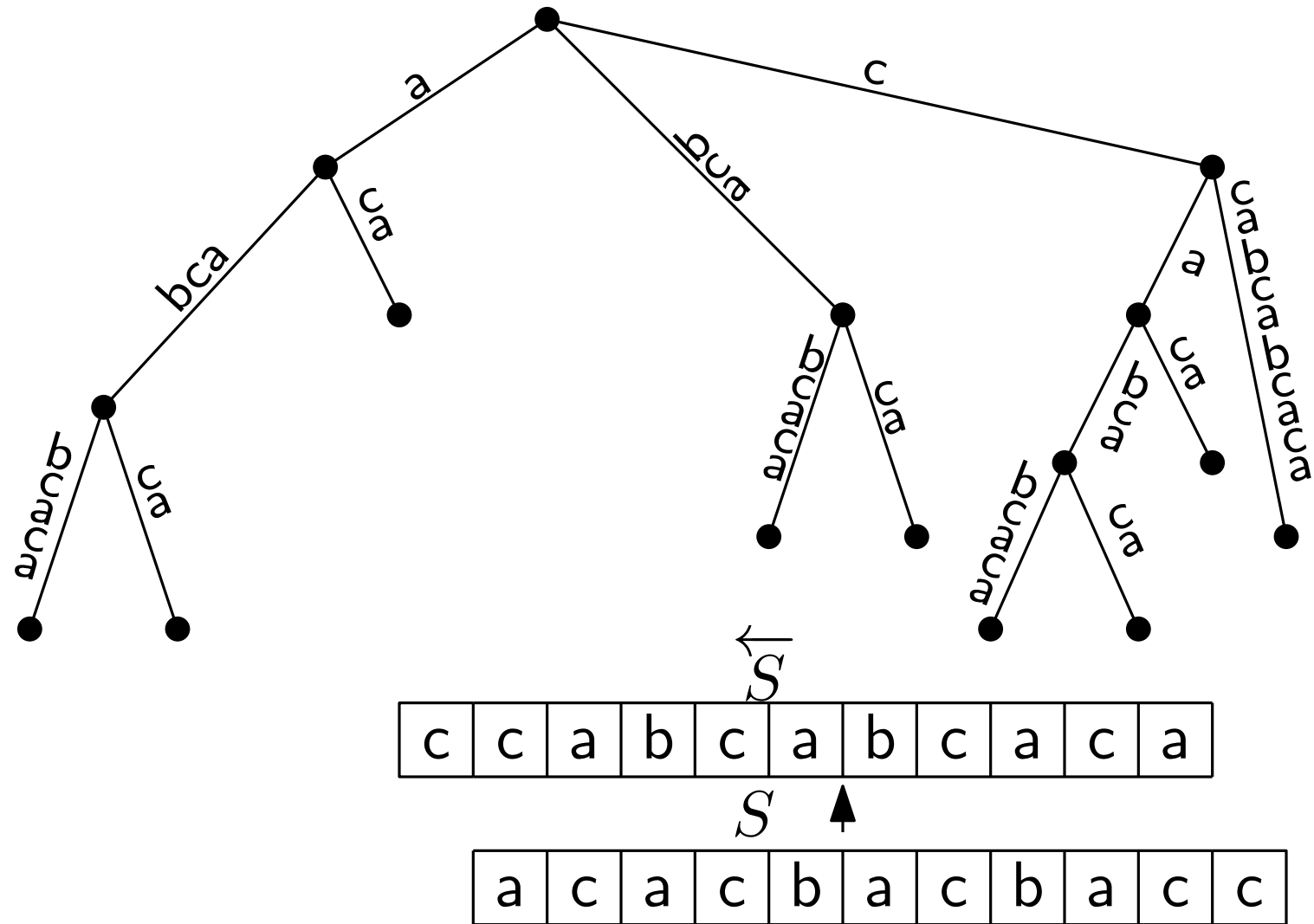


Solution - AddRight



- Maintain online suffix tree of reverse string

Solution - AddRight

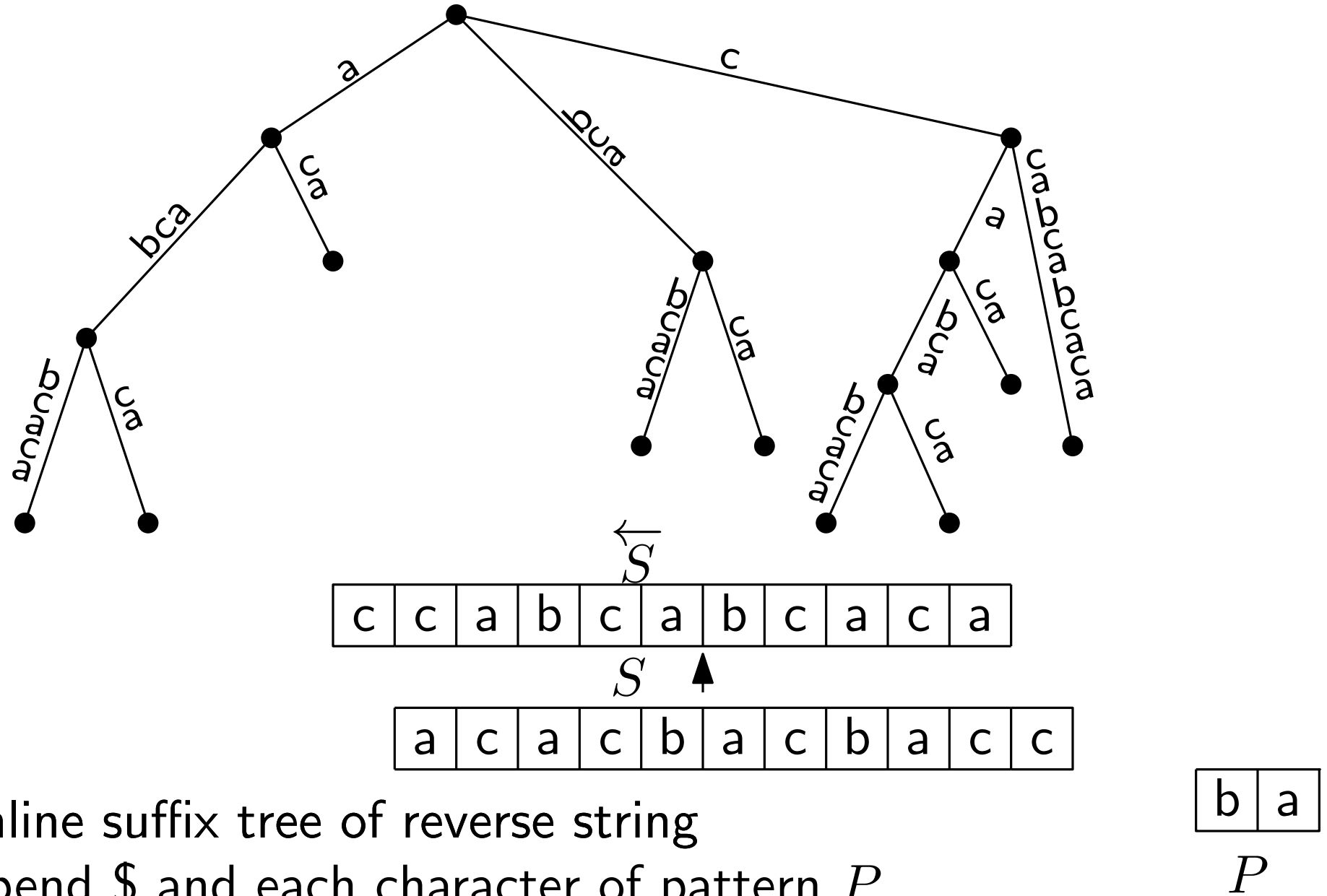


- Maintain online suffix tree of reverse string

b	a
---	---

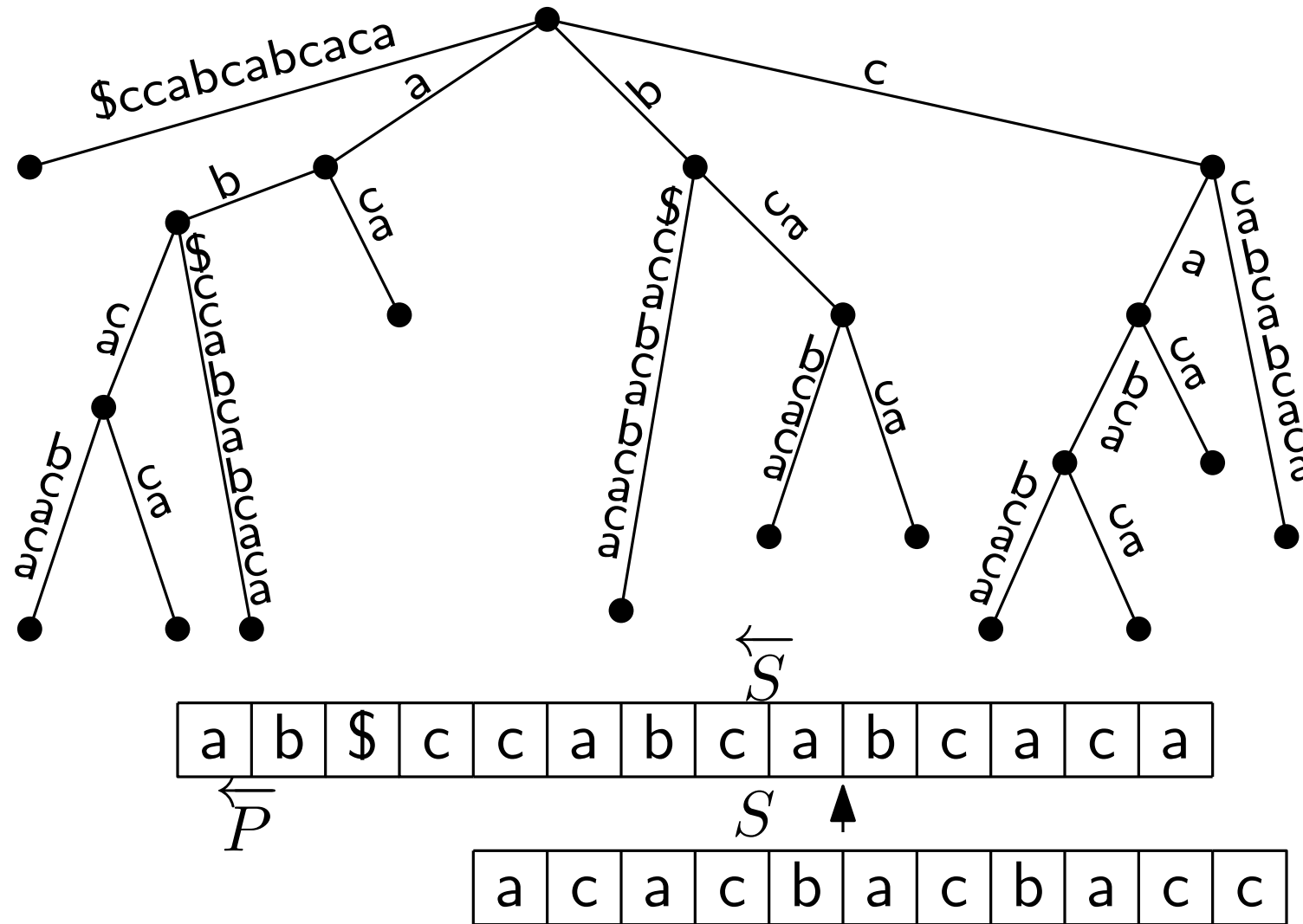
P

Solution - AddRight



- Maintain online suffix tree of reverse string
- Query: Prepend \$ and each character of pattern P

Solution - AddRight

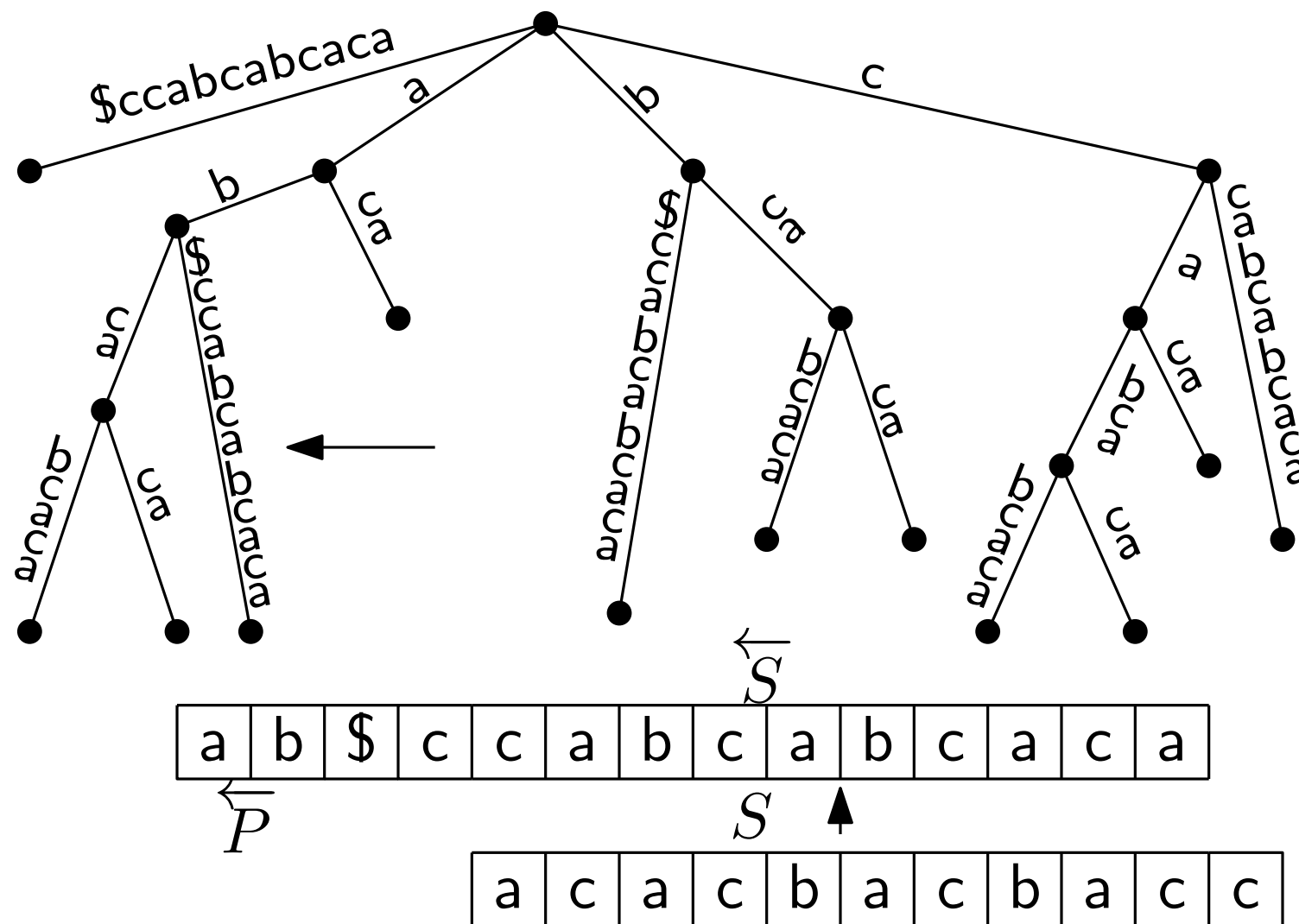


- Maintain online suffix tree of reverse string
- Query: Prepend \$ and each character of pattern P

b	a
---	---

 P

Solution - AddRight

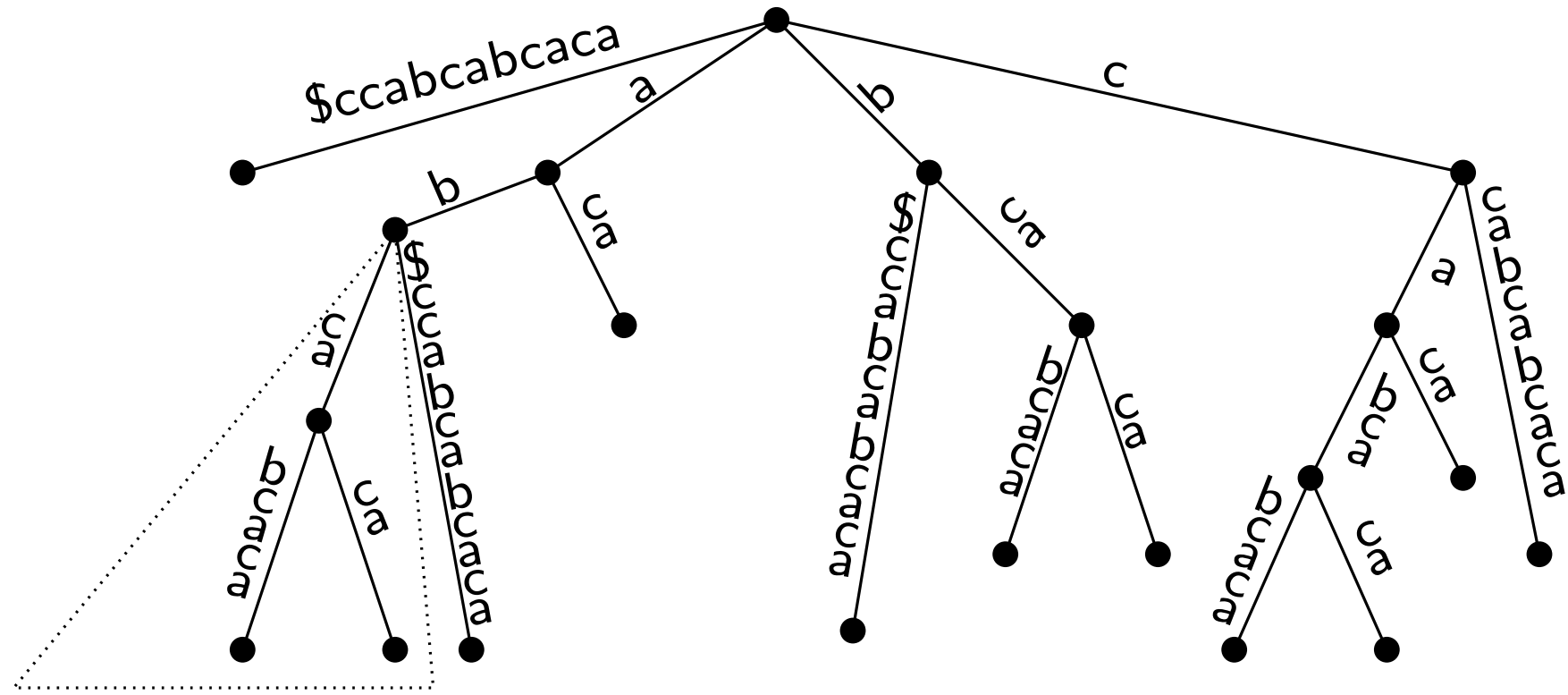


- Maintain online suffix tree of reverse string
- Query: Prepend \$ and each character of pattern P

b	a
---	---

 P

Solution - AddRight

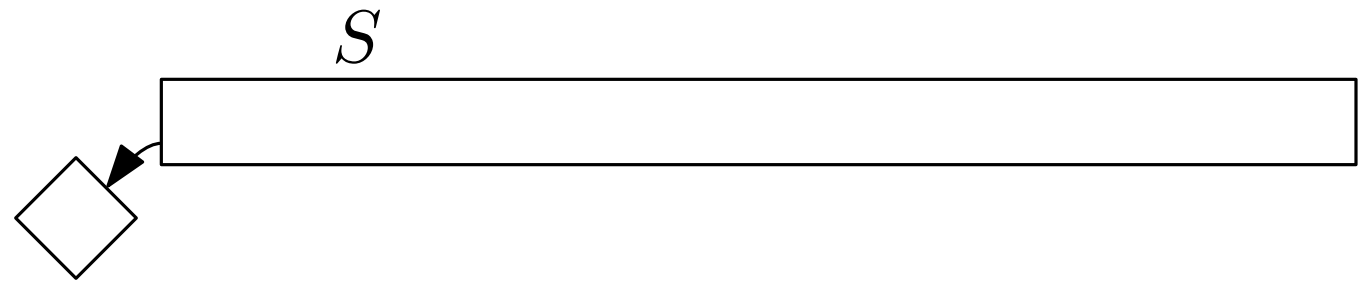


- Linear space

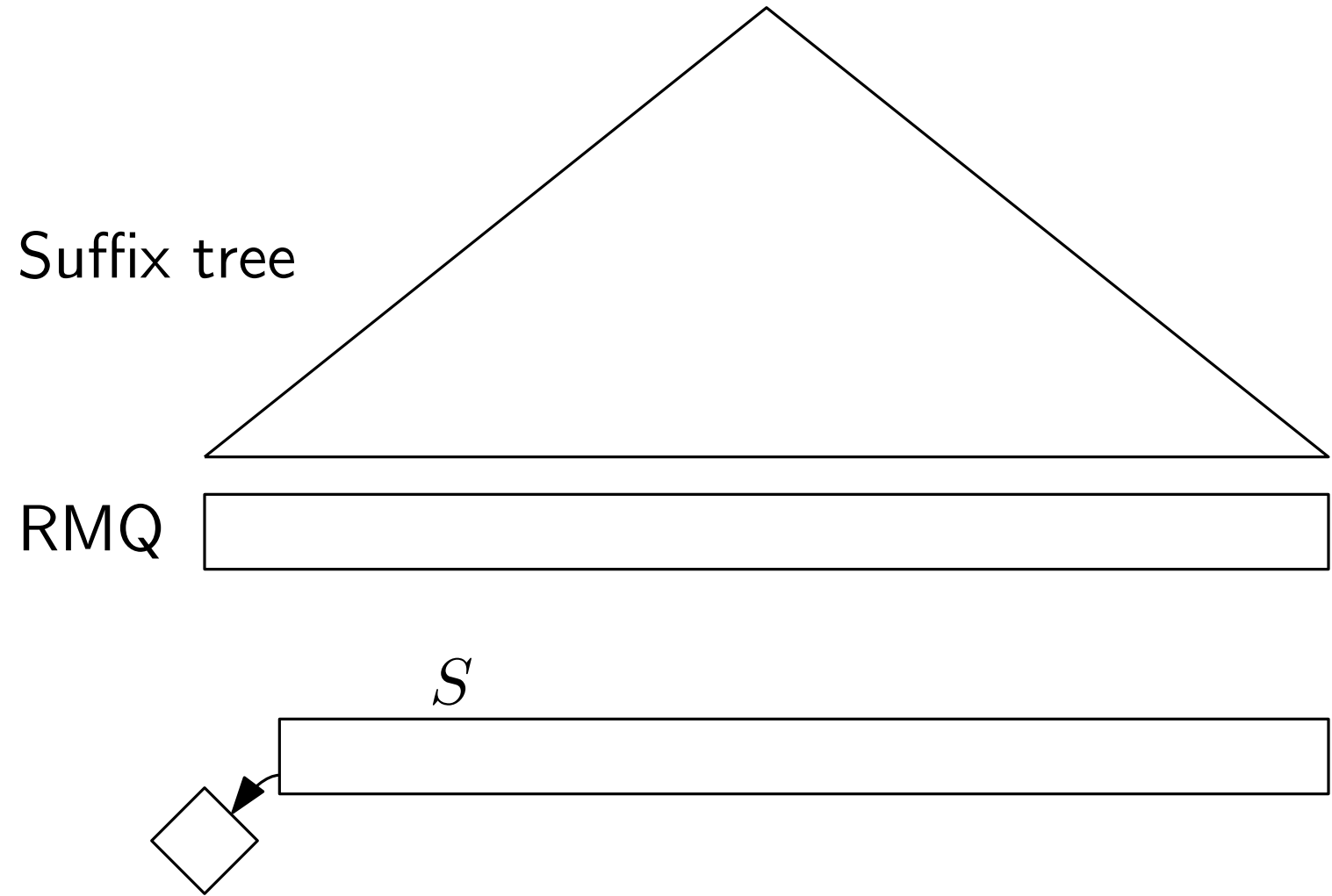
- $O\left(\log \log |S| + \frac{(\log \log \sigma)^2}{\log \log \log \sigma}\right)$ time or $O(\log \log |S| + \log \log \sigma)$ time whp.

(Fischer & Gawrychowski 2015, Kopelowitz 2012)

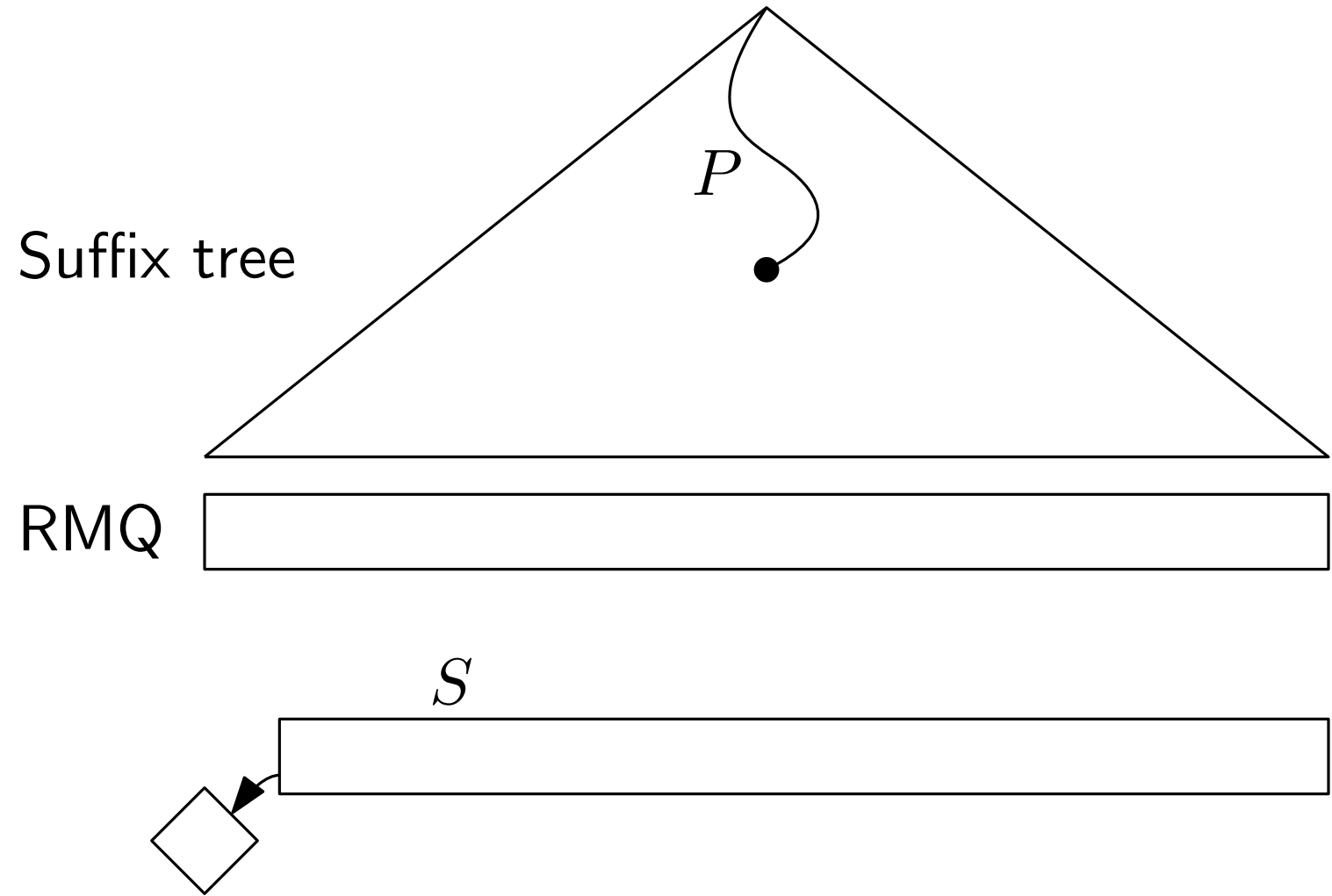
Solution - RemoveLeft



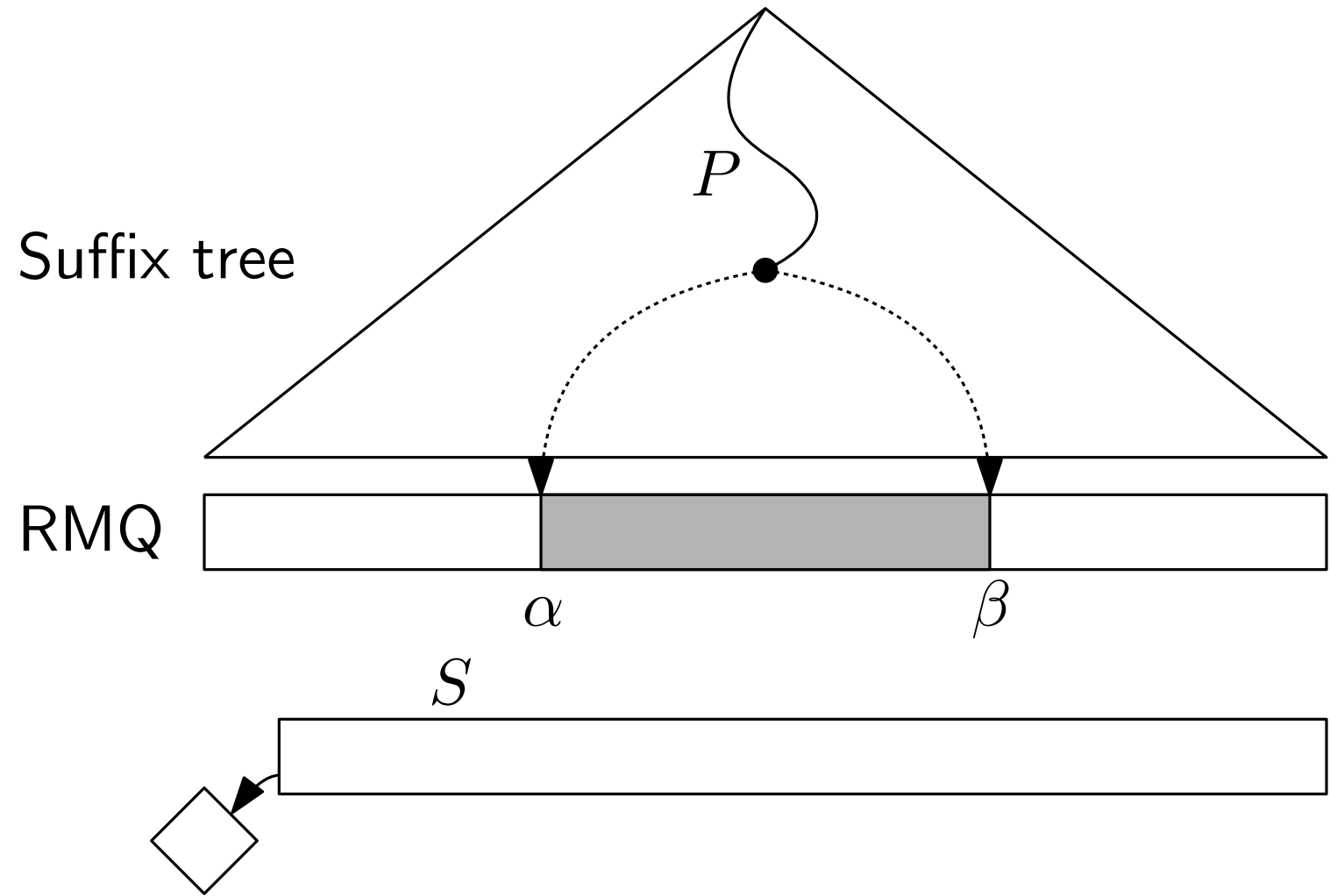
Solution - RemoveLeft



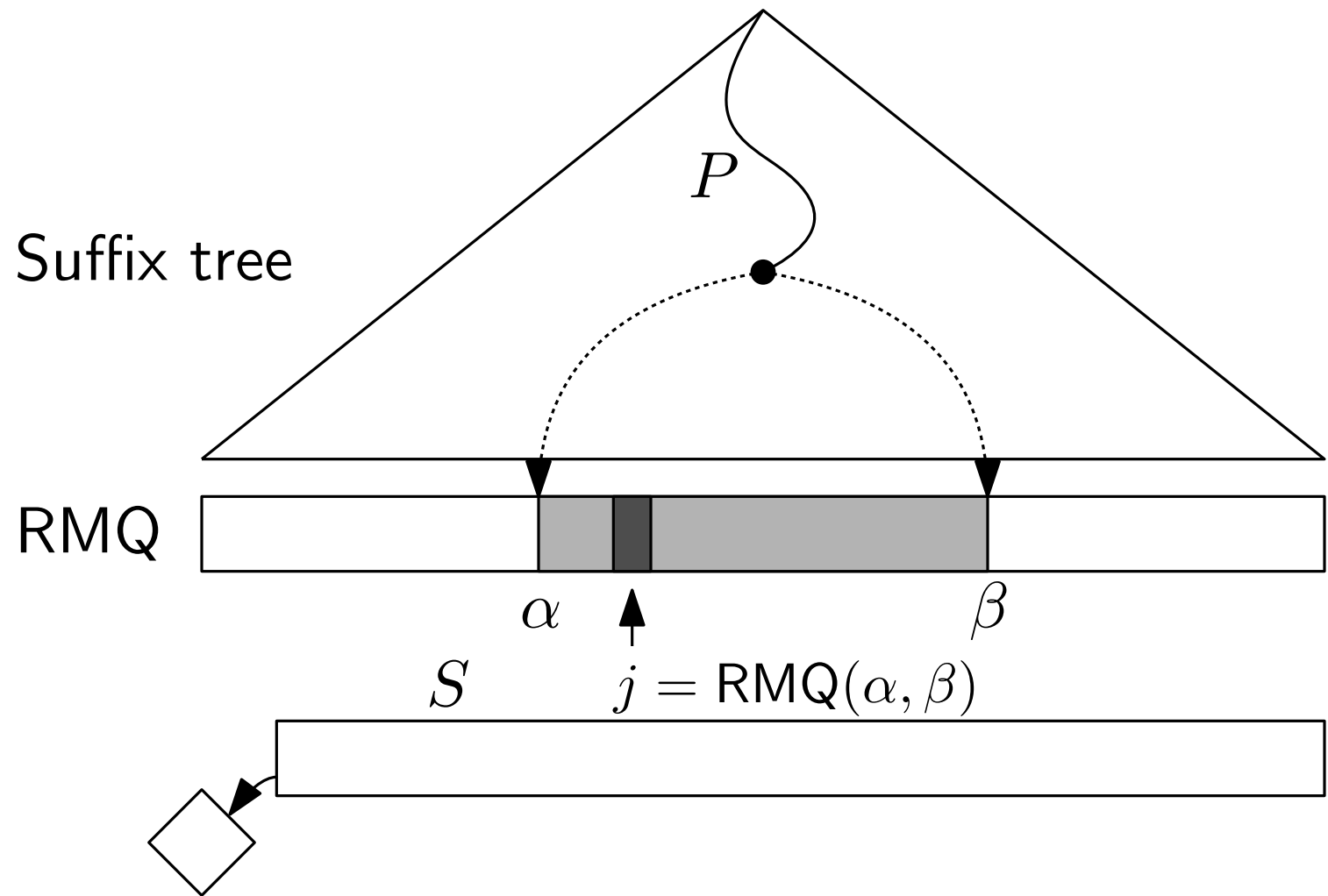
Solution - RemoveLeft



Solution - RemoveLeft

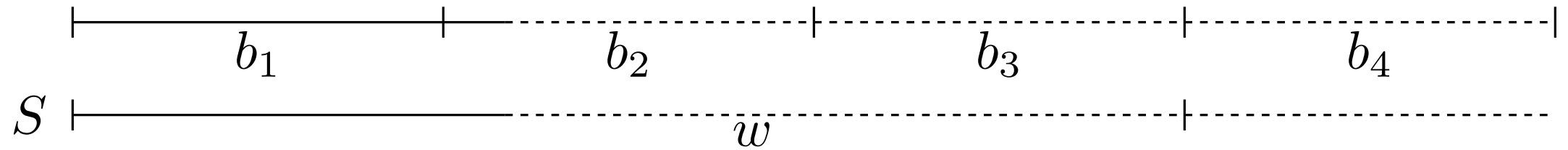


Solution - RemoveLeft



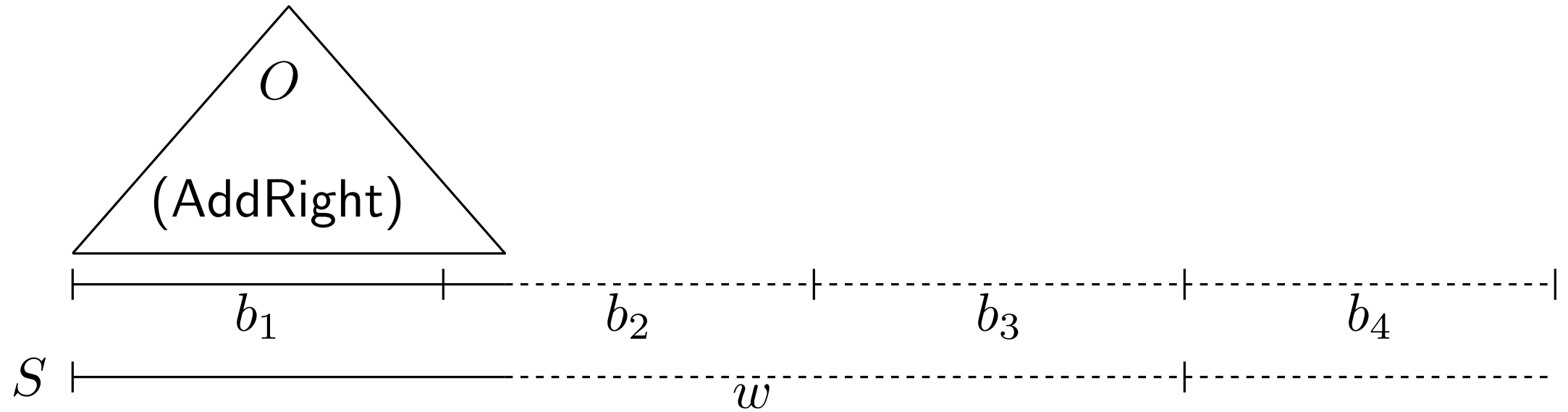
- Find rightmost (according to position in S) occurrence
- Recurse on $[\alpha, j - 1]$ and $[j + 1, \beta]$
- RMQ $O(|S|)$ preprocessing, $O(1)$ query time (Bender & Farach-Colton 2000)

Solution - Combining



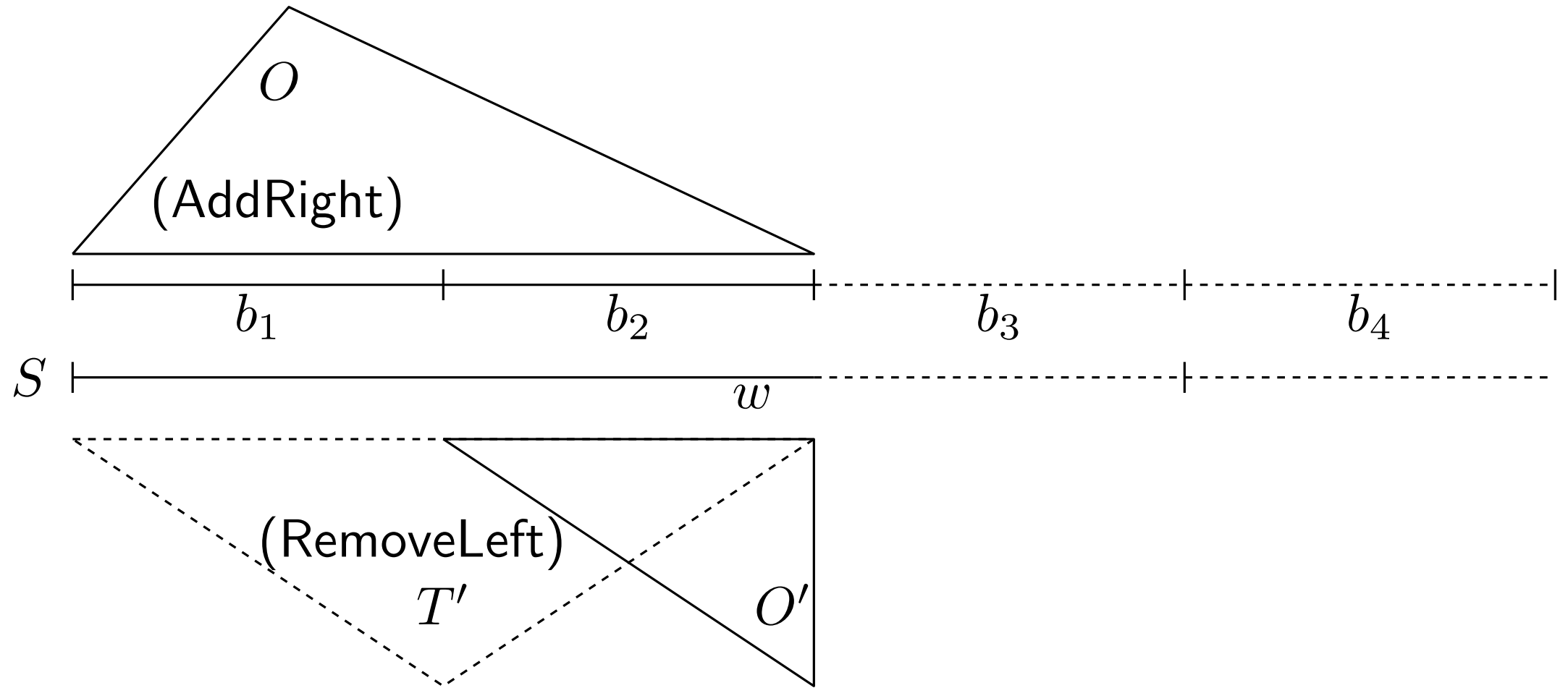
- Blocks of length $B = \lceil w/3 \rceil$

Solution - Combining



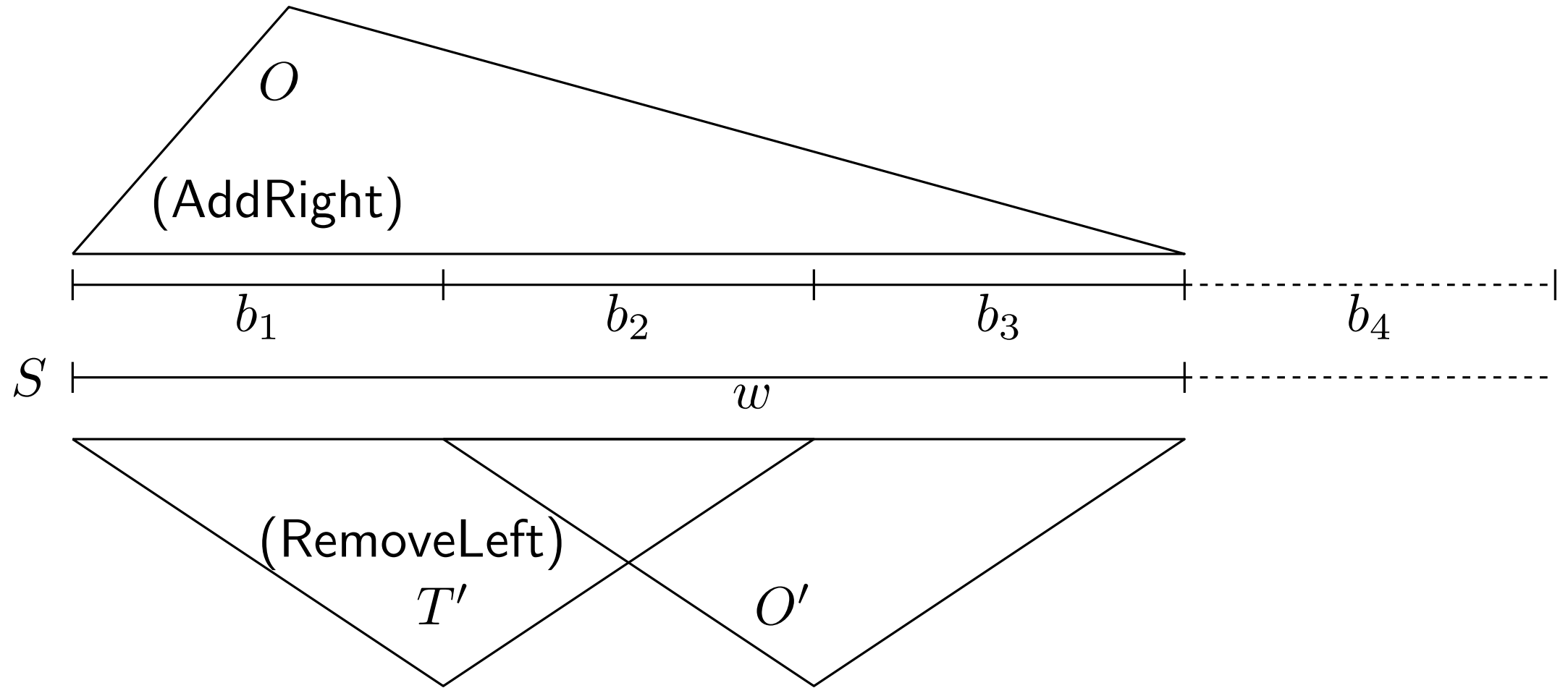
- Blocks of length $B = \lceil w/3 \rceil$

Solution - Combining



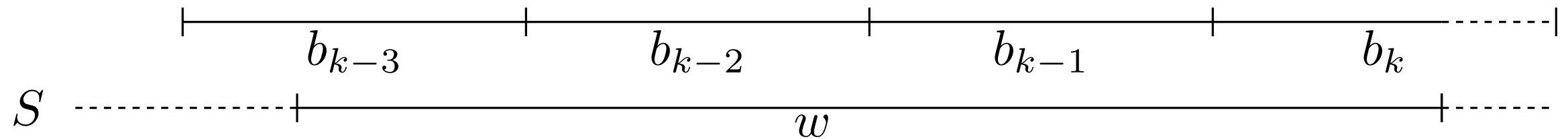
- Blocks of length $B = \lceil w/3 \rceil$
- Rebuild once per block in the background
- T' preprocessing $O(B)$ online suffix tree operations \rightarrow swap after one block

Solution - Combining



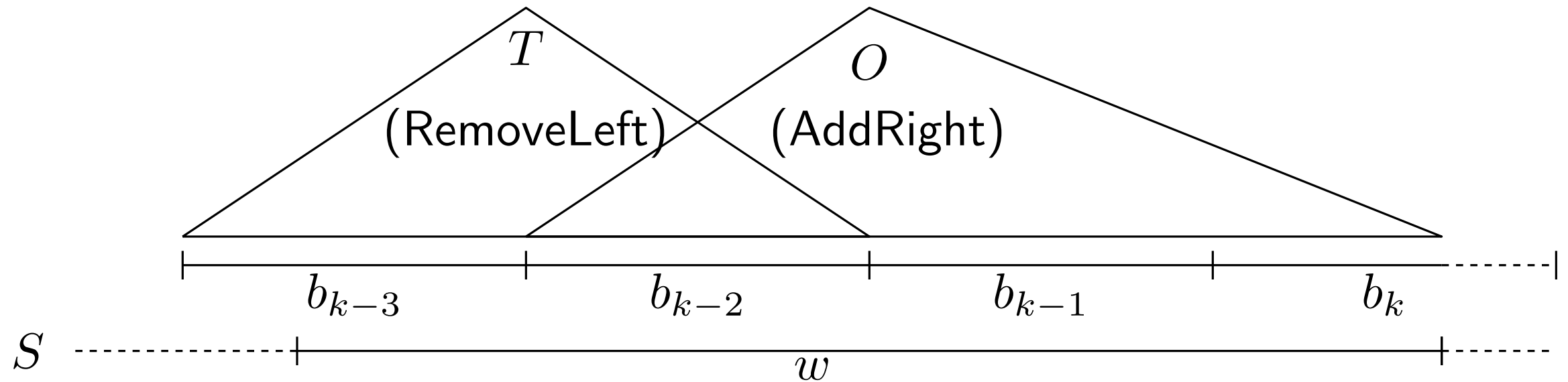
- Blocks of length $B = \lceil w/3 \rceil$
- Rebuild once per block in the background
- T' preprocessing $O(B)$ online suffix tree operations \rightarrow swap after one block

Solution - Combining



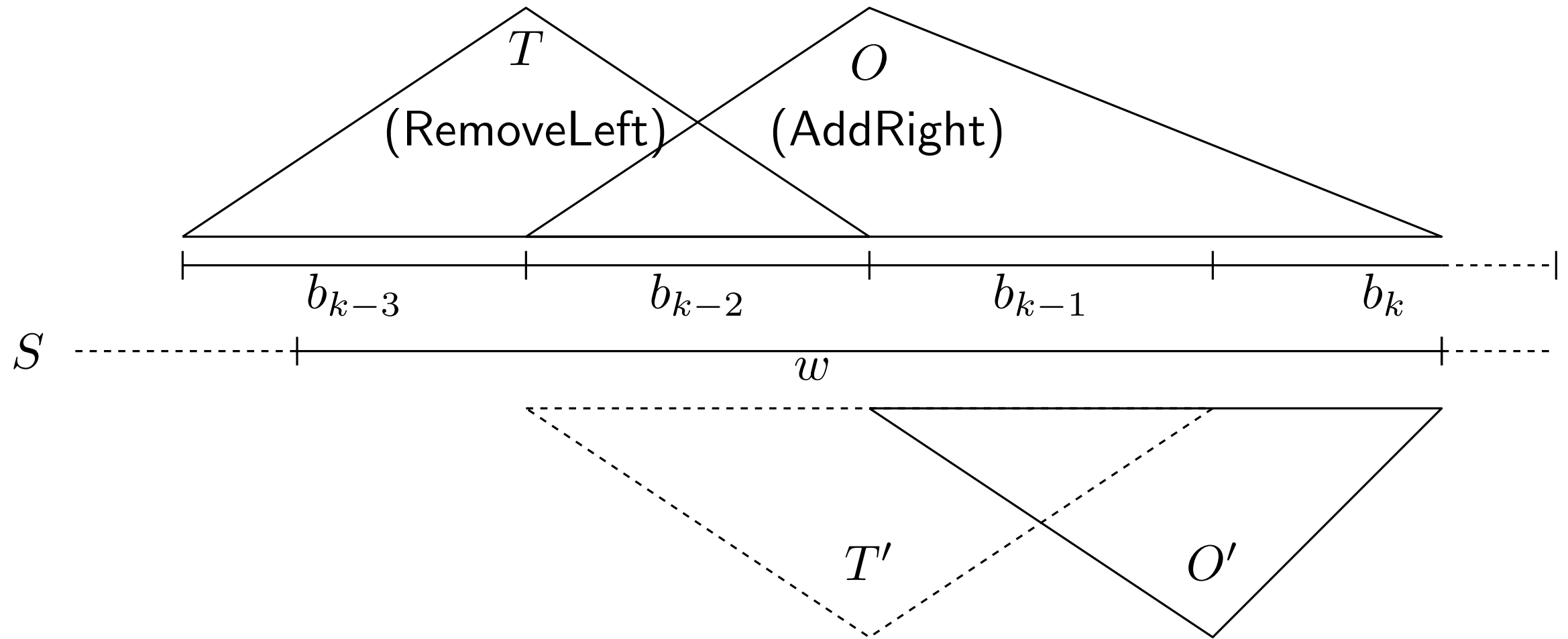
- Blocks of length $B = \lceil w/3 \rceil$
- Rebuild once per block in the background
- T' preprocessing $O(B)$ online suffix tree operations \rightarrow swap after one block

Solution - Combining



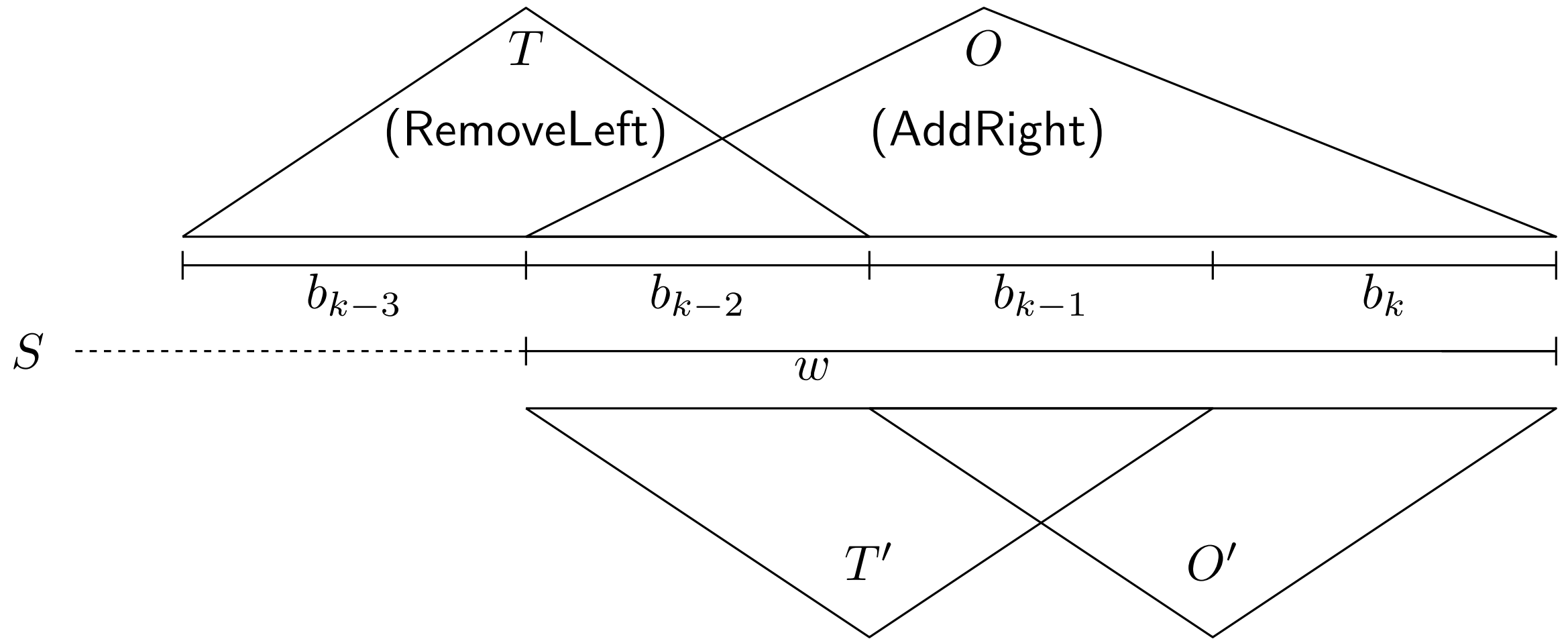
- Blocks of length $B = \lceil w/3 \rceil$
- Rebuild once per block in the background
- T' preprocessing $O(B)$ online suffix tree operations \rightarrow swap after one block

Solution - Combining



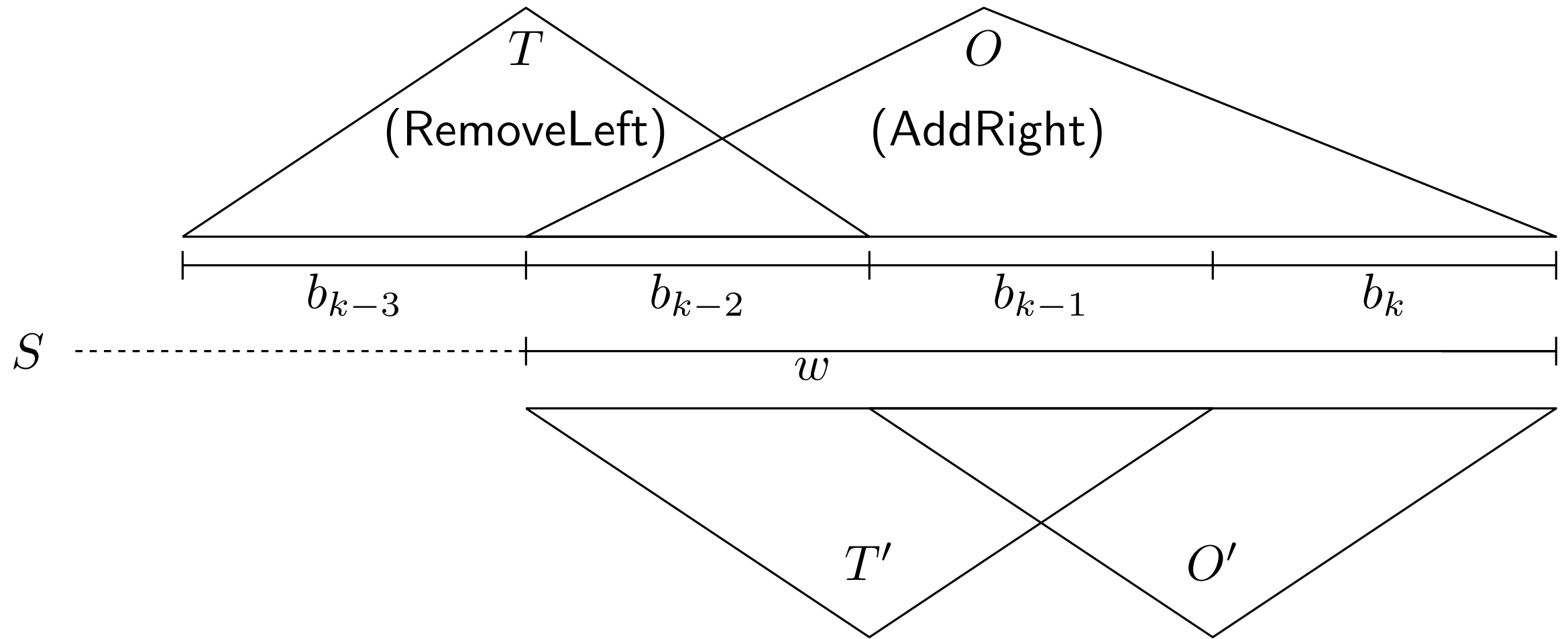
- Blocks of length $B = \lceil w/3 \rceil$
- Rebuild once per block in the background
- T' preprocessing $O(B)$ online suffix tree operations \rightarrow swap after one block

Solution - Combining



- Blocks of length $B = \lceil w/3 \rceil$
- Rebuild once per block in the background
- T' preprocessing $O(B)$ online suffix tree operations \rightarrow swap after one block

Solution - Combining



- Reports all occurrences when $|P| \leq B$
- Observation: $|P| > B \rightarrow$ occurrences are $O(1)$ chains
- Find chains when $\frac{1}{4}B$ characters of P has arrived and extend them naively

Thank you!