Lyndon Words, the Three Squares Lemma, and Primitive Squares

Hideo Bannai[1], Takuya Mieno[2], Yuto Nakashima[2]

[1] Tokyo Medical and Dental University

[2] Kyushu University

Lyndon Words [Lyndon, '54]

Definition 1.

A non-empty string w is said to be a ***Lyndon word*** if w is lexicographically smaller than any of its non-empty proper suffixes.

E.g. `awesome` is a Lyndon word. `excellent` is **NOT** a Lyndon word.

`awesome` < `wesome`

`awesome` < `esome`

`awesome` < `some`

`awesome` < `ome`

`awesome` < `me`

`awesome` < `e`

`excellent` > `cellent`

- Applications:
  - Proof of the Runs Theorem [Bannai et al., '17].
  - Designing algorithms for computing all runs in a string [Bannai et al., '17] [Kosolobov, '16] [Gawrychowski et al., '16] [Crochemore at al., '16].

Squares and Primitively Rooted Squares

- A string of the form u^2 $(= uu)$ is a ***square***.
  - u is called the ***root*** of the square.

E.g.

`abaaba` is a square.
The root is `aba`.

`babababa` is a square.
The root is `baba`.

- String w is ***primitive*** if w cannot be written as an integer power of another shorter string.

E.g.

`aba` is primitive.

`baba` is **NOT** primitive.

- Square u^2 is a ***primitively rooted square*** if the root u is primitive.

E.g.

`abaaba` is
a primitively rooted square.

`babababa` is
NOT a primitively rooted square.

Three Squares Lemma [Crochemore & Rytter, '95]

Lemma 1. ([Crochemore & Rytter, '95])

Let u^2, v^2, w^2 be three prefixes of some string such that w is primitive and $|u| > |v| > |w|$. Then, $|u| \geq |v| + |w|$.

- This is a very important lemma in combinatorics on words.
  - E.g., the Three Squares Lemma was used to obtain the upper bound $2n$ of the number of distinct squares [Fraenkel & Simpson' 98].

- The original proof of the Three Squares Lemma was based on the well known "Periodicity Lemma" [Fine & Wilf, '65].

Our First Result.

- Give an alternate proof of the Three Squares Lemma by using arguments based on **Lyndon words**.
- Show a (slightly) stronger variant of the Three Squares Lemma.

Number of Primitively Rooted Squares

- Let $psq(n)$ be the maximum number of occurrences of **primitively rooted squares** in a string of length n.

Given by Fibonacci words
[Fraenkel & Simpson, '99]

Proven by using the Three Squares Lemma
[Crochemore & Rytter, '95] [Fraenkel & Simpson, '99]

$$0.796n\log_2 n + O(n) \leq \quad psq(n) \quad \leq 1.441n\log_2 n$$

Theorem 1. (Our Second Result)

$$psq(n) \leq n\log_2 n.$$

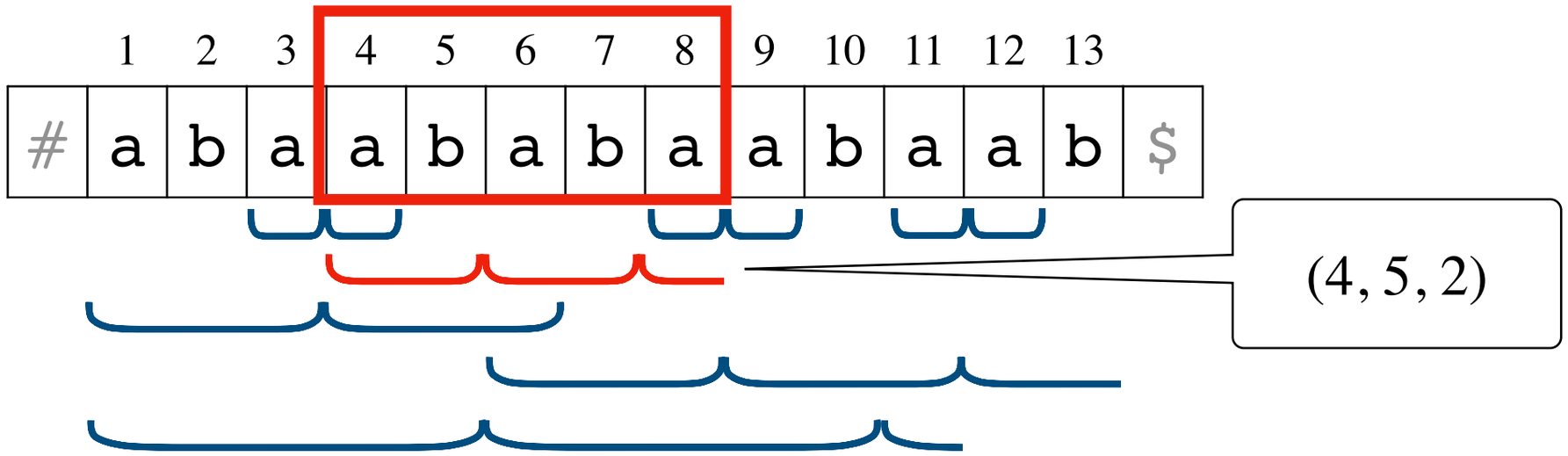
- Our proof again uses arguments based on **Lyndon words**.

Runs (Maximal Repetitions)

Definition 2.

An occurrence $w[s\,..\,e] = v$ is a ***run*** (or a maximal repetition) in w, if its smallest period p satisfies $p \leq |v|/2$, and both $w[s-1..e]$ $w[s\,..\,e+1]$ do not have period p.

E.g.



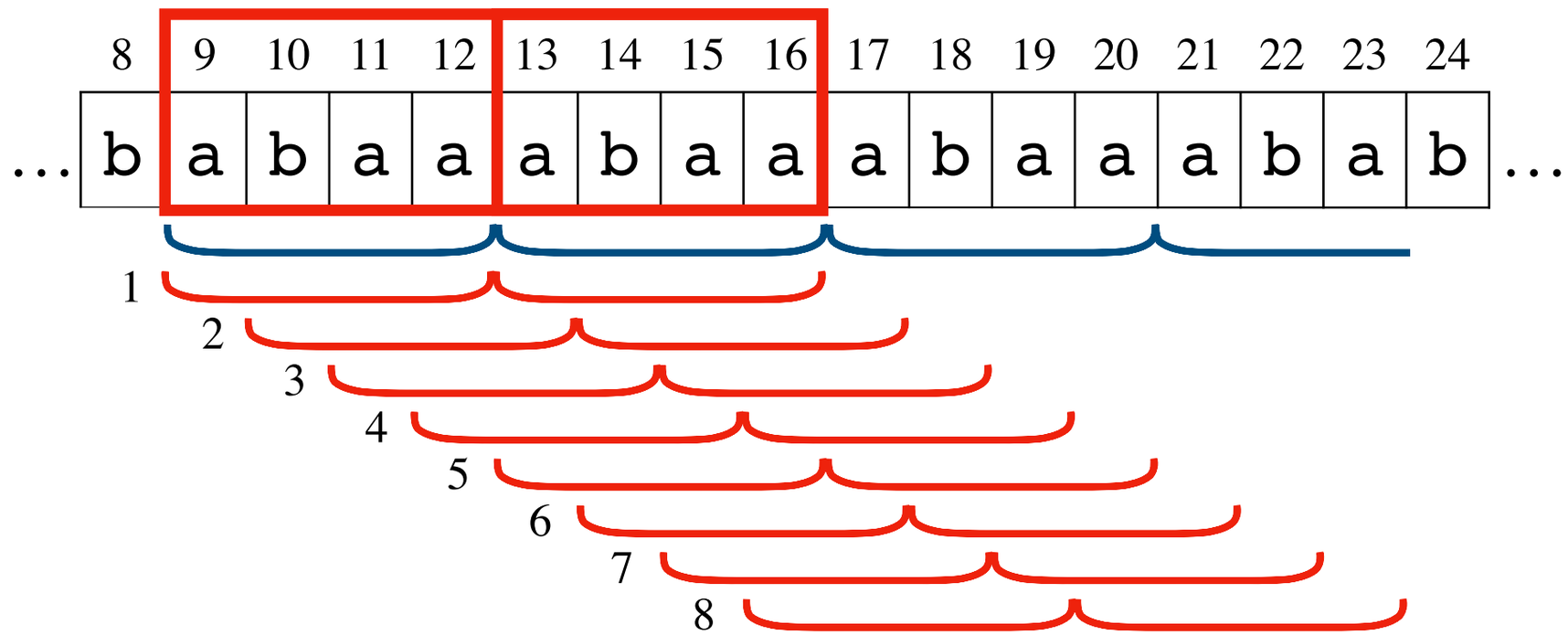
#, $: dummy characters.

- A run r is represented by a tuple (s_r, ℓ_r, p_r) where s_r is the starting position of the run, ℓ_r is the length of the run, and p_r is the smallest period of the run.

Runs and Primitively Rooted Squares

- The number of primitively rooted squares inside a run r with the same smallest period p_r is equal to $\ell_r - 2p_r + 1$.

E.g. Consider the run $r = w[9..23]$ represented by $(9, \mathbf{15}, \mathbf{4})$.



There are **15** – 2×**4** + 1 = **8** squares with period **4**.

Observation 1.

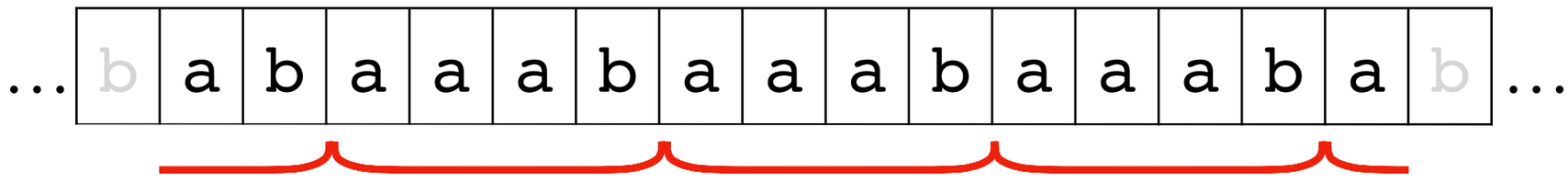
$$psq(n) = \sum_r (\ell_r - 2p_r + 1).$$

L-roots [Crochemore et al., '14]

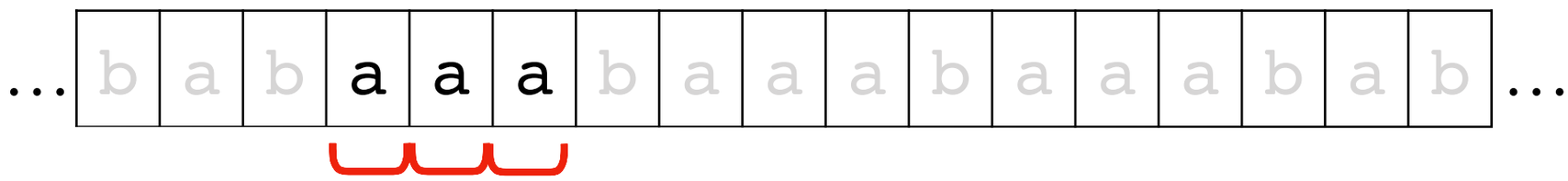
Definition 3.

For any run r, an ***L-root*** λ_r is a substring of r that is a Lyndon word whose length is equal to the smallest period of r.

E.g.



`aaab` is an L-root of the run `abaaabaaaabaaaba`



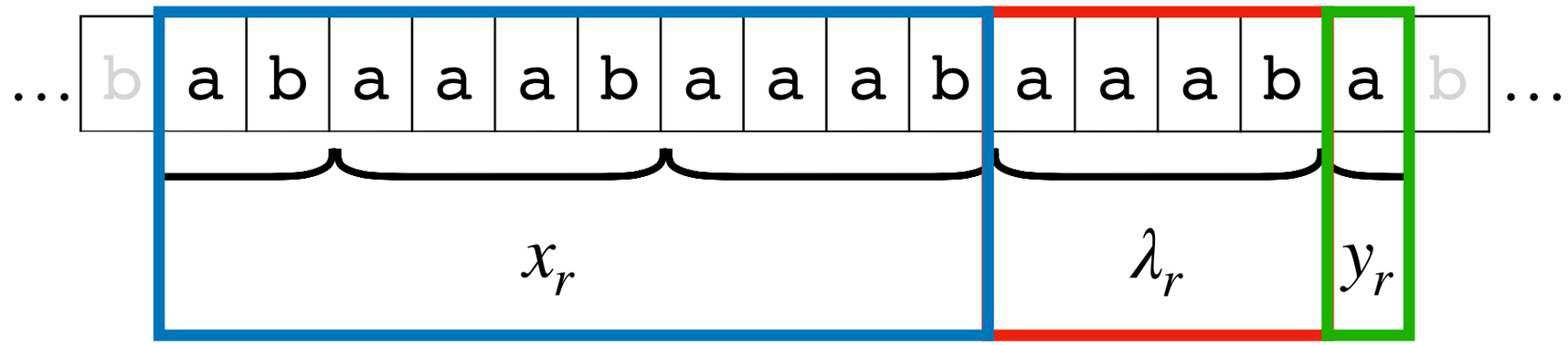
`a` is an L-root of the run `aaa`

- It is easy to see that an L-root of a run always exists.

L-roots and an Upper Bound of $psq(n)$

- We can decompose a run r into $r = x_r \lambda_r y_r$ such that λ_r is the **rightmost** L-root of r, y_r is a proper prefix of λ_r, and x_r is the rest.
- Then, $\ell_r - 2p_r + 1 \leq \ell_r - |\lambda_r y_r| = |x_r|$.

E.g.



Observation 2.

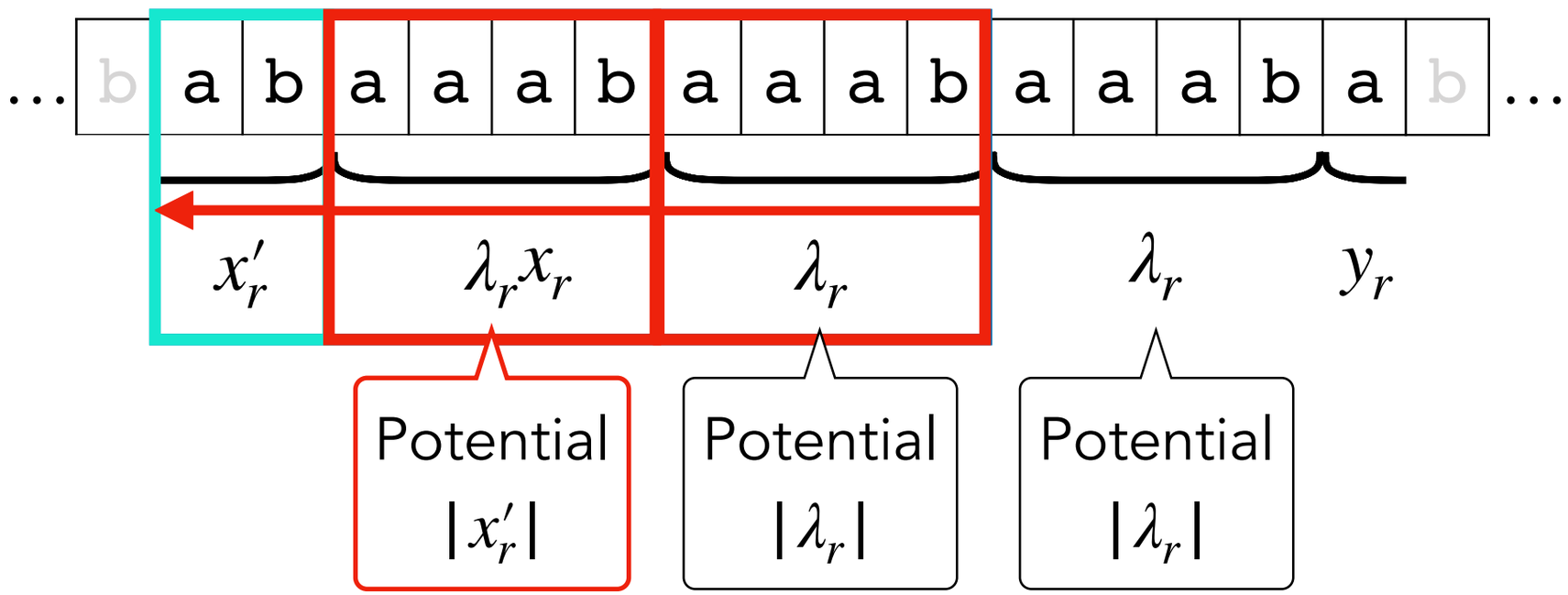
$$psq(n) = \sum_r (\ell_r - 2p_r + 1) \leq \sum_r |x_r|.$$

- Thus, it suffice to show that the upper bound of the total sum of $|x_r|$ for all runs in w is $n \log_2 n$.

Distributing the Length of x_r among L-roots

- We further decompose x_r into $x'_r\lambda_r^k$ such that x'_r is a proper suffix of λ_r and k is a non-negative integer.

E.g. $k = 2$ in this example.



- Let the ***potential*** of an L-root of r be the minimum of two values; $|\lambda_r|$ and the maximal left-extension of the periodicity of the L-root.
- Then, the length $|x_r| = |x'_r| + k|\lambda_r|$ can be regarded as the total **sum of the potentials of all L-roots** of r.

Lyndon Trees [Barcelo, '90] [Bannai et al., '17]

Definition 4.

The ***Lyndon tree*** of a Lyndon word w, denoted by LTree(w) is an ordered binary tree defined recursively as follows:

- If $|w| = 1$, then the Lyndon tree of w is a single node labeled w.
- if $|w| \geq 2$, then the root is labeled w, and the left and right children of w are respectively the Lyndon trees of u and v, where $w = uv$ and v is the lexicographically smallest proper suffix of w.

u, v are guaranteed to be Lyndon words.

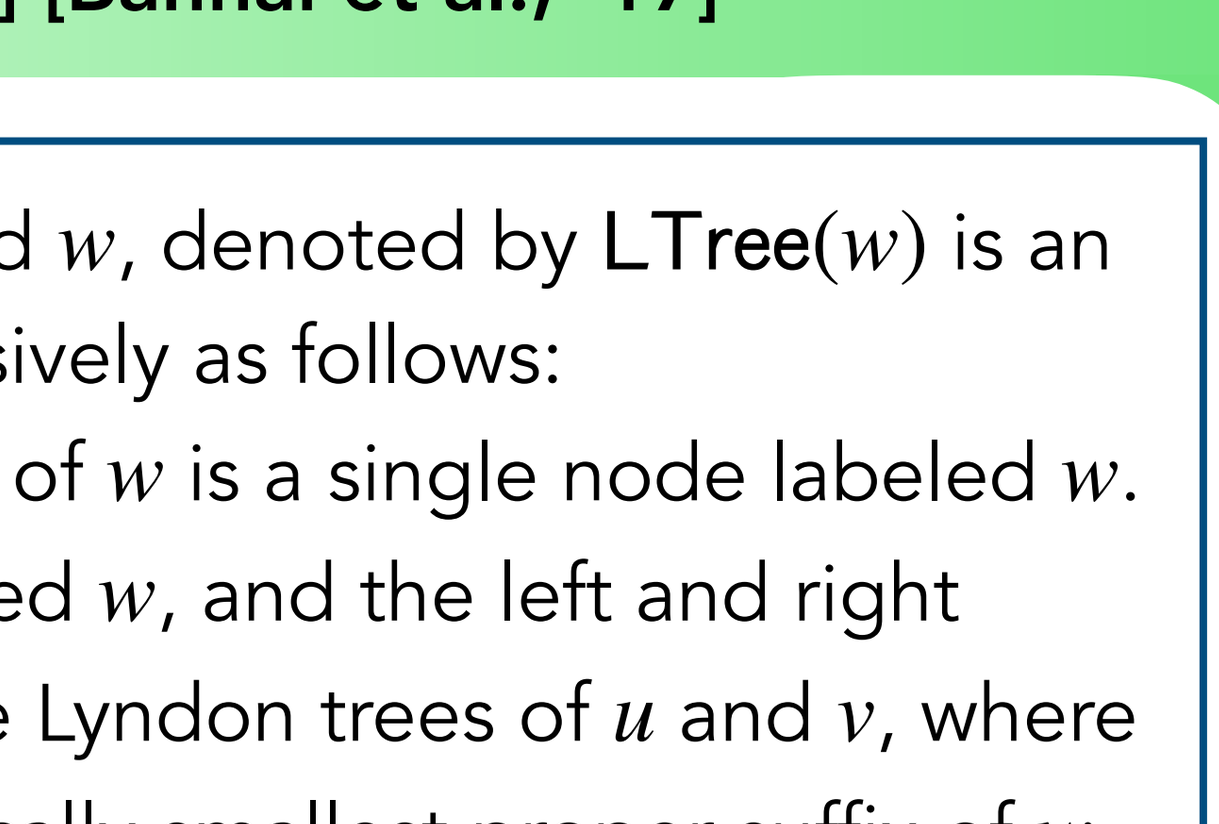
E.g.

● is labeled by #abbbabb.

The **right** child is LTree(abb), and

the **left** child is LTree(#abbb).

Note: $\# < c$ for any $c \in \Sigma$.



Two Lexicographic Orders

- Let $<_0$ be a total order over Σ.
- Also let $<_1$ be the ***opposite*** order of $<_0$.
  - Namely, for any $a, b \in \Sigma$, $a <_0 b$ iff $b <_1 a$.

- We call $<_0$ the *standard* order, and $<_1$ the *opposite* order:

$$\texttt{a} <_0 \texttt{b} <_0 \texttt{c} <_0 \cdots <_0 \texttt{y} <_0 \texttt{z}$$

$$\texttt{z} <_1 \texttt{y} <_1 \texttt{x} <_1 \cdots <_1 \texttt{b} <_1 \texttt{a}$$

- We consider two lexicographic orders induced by $<_0$ and $<_1$.

E.g.

$\texttt{abaa} <_0 \texttt{baa}$ $\qquad$ $\texttt{baa} <_1 \texttt{abaa}$

$\texttt{ba} <_0 \texttt{baaa}$ $\qquad$ $\texttt{ba} <_1 \texttt{baaa}$

`awesome` is **NOT** a Lyndon word w.r.t. the opposite order $<_1$.

$\texttt{wesome} <_1 \texttt{awesome}$

L-roots and Longest Lyndon Words

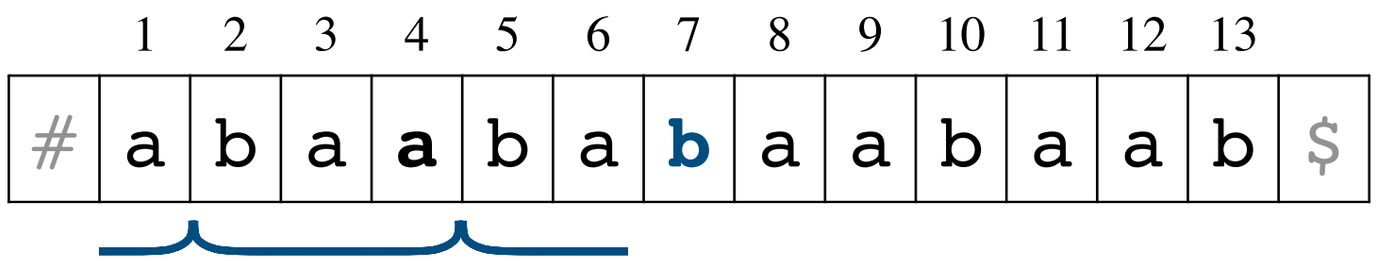
Lemma 2. ([Bannai et al., '17])

For any run $w[s\,..\,t]$ with period p, consider the lexicographic order $< \in \{ <_0 , <_1 \}$ such that $w[t+1] < w[t+1-p]$.

Then, any occurrence of the L-root of the run $w[s\,..\,t]$ is the longest Lyndon word starting at that position.

- For any run in a string, we will refer to the lexicographic order considered in Lemma 2 as ***the*** lexicographic order of the run.

E.g.



$\mathtt{a} <_0 \mathtt{b}$

$\mathtt{b} <_1 \mathtt{a}$

Consider run $w[1..6] = \mathtt{abaaba}$ and the *opposite* order $<_1$.

Then L-root $w[2..4] = \mathtt{baa}$ of the run is the longest Lyndon word starting at position 2 w.r.t. $<_1$.

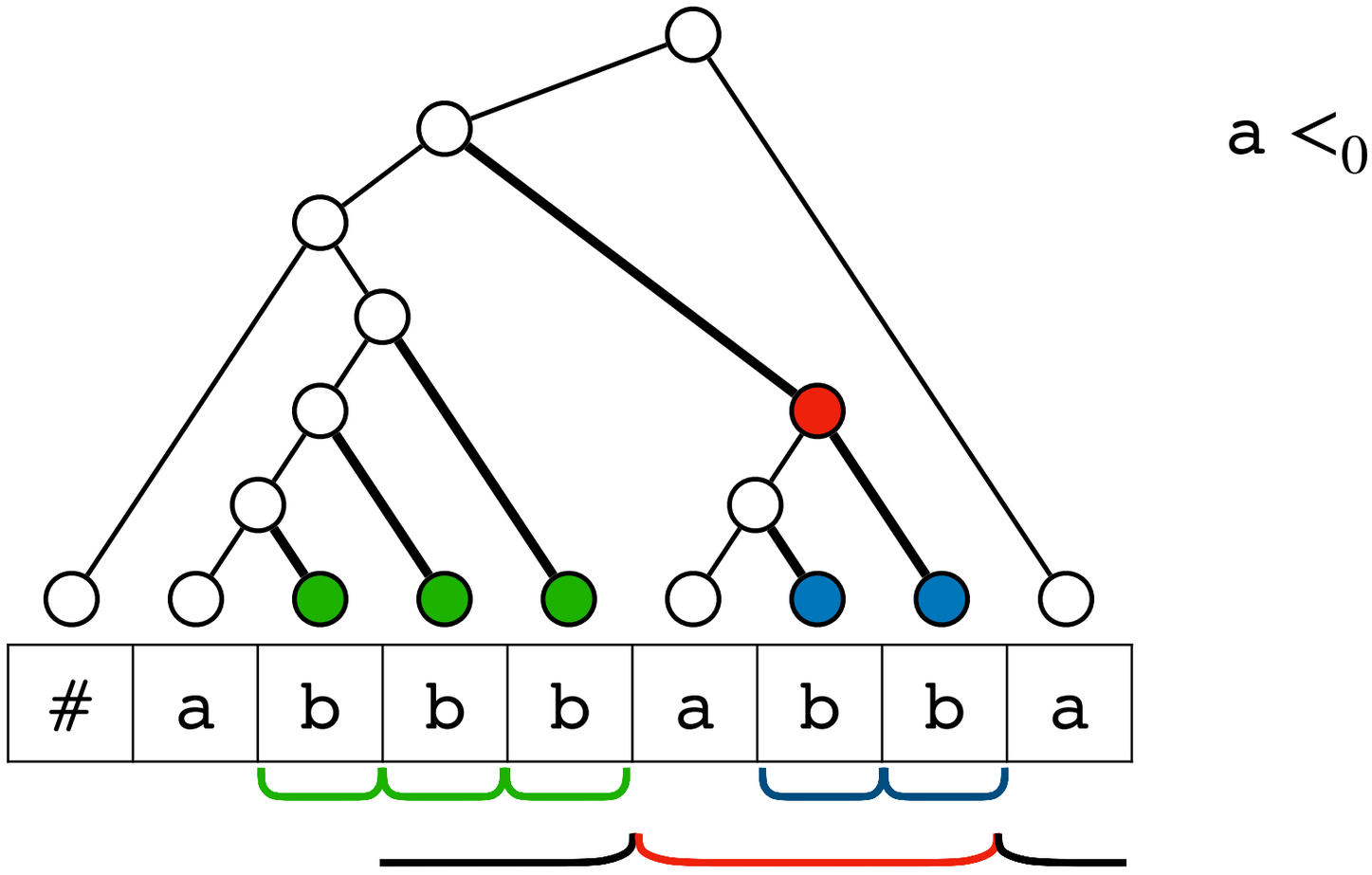
Thus, **the** lexicographic order of run $w[1..6]$ is $<_1$.

L-roots and Lyndon Trees

Property. ([Bannai et al., '17])

For each run r in a string, each L-root of r corresponds to a **right node** of the Lyndon tree w.r.t. the lexicographic order of r.

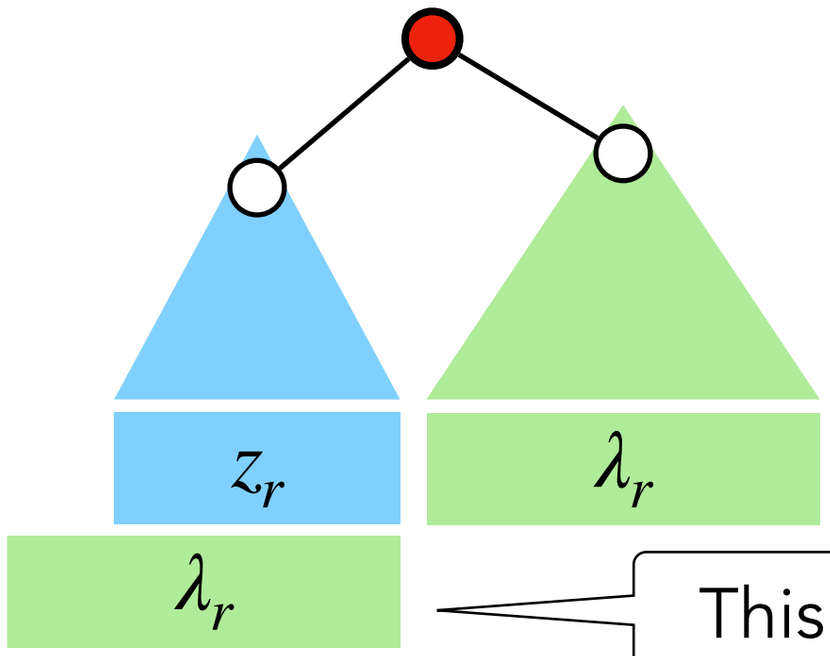
E.g.



- In this example, for the lexicographic order $<_0$, all L-roots of runs correspond to **right nodes** of the Lyndon tree w.r.t. $<_0$.

How Long Can the Periodicity Extend?

- Consider potentials of L-roots by looking at Lyndon Trees.

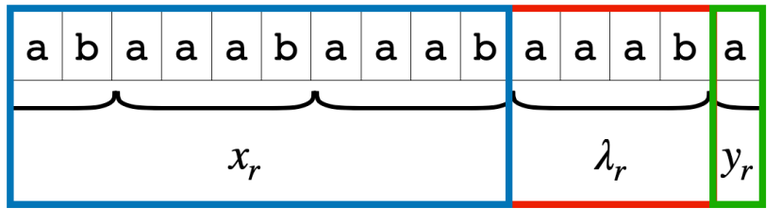


We assume that $|z_r| < |\lambda_r|$ where z_r is the left sibling of λ_r.
If λ_r can be extended to the left beyond z_r, then $\lambda_r < z_r$.
On the other hand, $z_r\lambda_r$ is a Lyndon word, and thus, $z_r\lambda_r < \lambda_r$.
These imply that $z_r\lambda_r < \lambda_r < z_r$, a contradiction.

- Therefore, the potential of the L-root λ_r is at most $\min\{|z_r|, |\lambda_r|\}$ where z_r is the string corresponding to the left sibling.

The Total Sum of Potentials

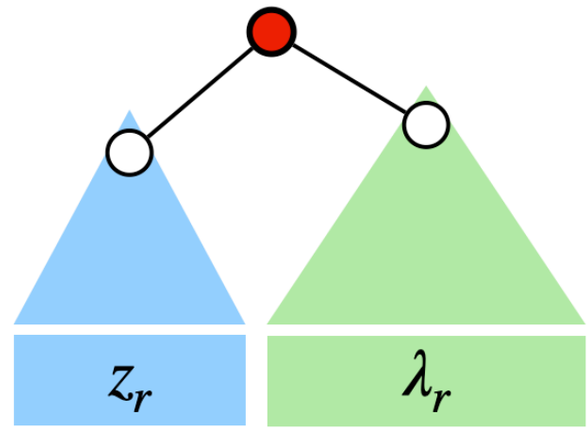
- Now, we are ready to analyze an upper bound of $psq(n)$.



$$psq(n) \le \sum_r |x_r|$$

$$= \sum_r \sum_{\text{all occurrences of } \lambda_r} (\text{potential of L-root } \lambda_r)$$

$$\le \sum_r \sum_{\text{all occurrences of } \lambda_r} \min\{|z_r|, |\lambda_r|\}$$



$$\le 2 \times S(n)$$

$$\le n \log_2 n$$

math

$S(n)$ denotes the maximum number of the total sum of $\min\{|z_r|, |\lambda_r|\}$ for all nodes in any Lyndon tree of a string of length n.

$$S(n) = \begin{cases} 0 & \text{if } n = 1, \\ \max\{S(n_1) + S(n_2) + \min\{n_1, n_2\} \mid n_1, n_2 > 0 \text{ and } n_1 + n_2 = n\} & \text{otherwise}. \end{cases}$$

Conclusions and Open Questions

1. We gave an alternate proof of the "**Three Squares Lemma**" by using arguments based on Lyndon words, and showed a (slightly) stronger variant of the lemma.

2. We gave a new upper bound of the maximum number $psq(n)$ of **primitively rooted squares** in a string of length n:

$$psq(n) \leq n \log_2 n.$$

Open questions:

- Can we prove or generalize other known results on string repetitiveness by using arguments based on Lyndon words?
  - E.g., the "New Periodicity Lemma" [Fan et al., '06].
- Is our upper bound for $psq(n)$ tight?
  - The best known lower bound is $0.796n \log_2 n + O(n)$ for Fibonacci words [Fraenkel & Simpson, '99].