Accessing the Suffix Array via $\phi^{-1}$-Forest

Christina Boucher ¹

Dominik Köppl ²

Herman Perera ¹

Massimiliano Rossi ¹

¹Herbert Wertheim College of Engineering, University of Florida, USA

²M&D Data Science Center, Tokyo Medical and Dental University, Japan

extended talk on https://youtu.be/SjAX1Ru2_gE
what’s it all about?

goal
given the $r$-index, can we practically speed up the time for retrieving a suffix array (SA) entry?

why interesting?
- $r$-index is a version of the FM-index with refined SA samples
- while using less space than the FM-index on repetitive texts, the sparse SA samples make random accesses to $SA[i]$ slow
example

index sequences

- GATTACAT
- GATACAT
- GATTAGATA

for that:

- concatenate with $, and
- use # as terminal symbol

input becomes $T = \text{GATTACAT}$\$\text{GATACAT}$\$\text{GATTAGATA}$#
FM-index for text $T := GATTACAT$GATACAT$GATAGATA#

- uses BWT and a wavelet tree for pattern matching
- counting pattern occurrences works out of the box
- for locating the pattern occurrences, we need SA
- FM index samples SA by text position distance
FM-index for text $T := \text{GATTACAT}$\text{$\$$GATACAT}$\text{$\$$GATTAGATA}$

- uses BWT and a wavelet tree for pattern matching
- counting pattern occurrences works out of the box
- for locating the pattern occurrences, we need SA
- FM index samples SA by text position distance

$r$-index
- only stores SA samples at run boundaries
Example

- $\text{SA}[2] = 9$, $\text{SA}[20] = 8$
- $\text{BWT}[2] = \text{BWT}[3] = T$

SA access

how do we get $\text{SA}[i]$ with the $r$-index?

Toehold Lemma, Gagie+’18

$\text{BWT}[i] = \text{BWT}[i + 1] \Rightarrow \text{SA}[i + 1] - \text{SA}[i] = \text{SA}[j + 1] - \text{SA}[j]$ for $\text{SA}[j] := \text{SA}[i] - 1$
Corollary

- let $k \geq 0$ be largest value such that $\text{BWT}[i'] = \text{BWT}[i' + 1]$ for all $i'$ with $\text{SA}[i'] \in [\text{SA}[i] - k + 1..\text{SA}[i]]$
- let $\text{SA}[j] := \text{SA}[i] - k$

Then:

- $\text{BWT}[j] \neq \text{BWT}[j + 1]$ but still $\text{SA}[i + 1] - \text{SA}[i] = \text{SA}[j + 1] - \text{SA}[j]$. 

Corollary

- let \( k \geq 0 \) be largest value such that \( \text{BWT}[i'] = \text{BWT}[i' + 1] \) for all \( i' \) with \( \text{SA}[i'] \in [\text{SA}[i] - k + 1 \ldots \text{SA}[i]] \)
- let \( \text{SA}[j] := \text{SA}[i] - k \)

Then:

- \( \text{BWT}[j] \neq \text{BWT}[j + 1] \) but still
- \( \text{SA}[i + 1] - \text{SA}[i] = \text{SA}[j + 1] - \text{SA}[j] \).
Corollary

- Let $k \geq 0$ be largest value such that $\text{BWT}[i'] = \text{BWT}[i' + 1]$ for all $i'$ with $\text{SA}[i'] \in [\text{SA}[i] - k + 1..\text{SA}[i]]$
- Let $\text{SA}[j] := \text{SA}[i] - k$

Then:

- $\text{BWT}[j] \neq \text{BWT}[j + 1]$ but still
- $\text{SA}[i + 1] - \text{SA}[i] = \text{SA}[j + 1] - \text{SA}[j]$.
Corollary

- let $k \geq 0$ be largest value such that $\text{BWT}[i'] = \text{BWT}[i' + 1]$ for all $i'$ with $\text{SA}[i'] \in [\text{SA}[i] - k + 1 .. \text{SA}[i]]$
- let $\text{SA}[j] := \text{SA}[i] - k$

Then:

- $\text{BWT}[j] \neq \text{BWT}[j+1]$ but still
- $\text{SA}[i + 1] - \text{SA}[i] = \text{SA}[j + 1] - \text{SA}[j]$. 
Corollary

- Let $k \geq 0$ be largest value such that $\text{BWT}[i'] = \text{BWT}[i' + 1]$ for all $i'$ with $\text{SA}[i'] \in [\text{SA}[i] - k + 1..\text{SA}[i]]$
- Let $\text{SA}[j] := \text{SA}[i] - k$

Then:

- $\text{BWT}[j] \neq \text{BWT}[j + 1]$ but still
- $\text{SA}[i + 1] - \text{SA}[i] = \text{SA}[j + 1] - \text{SA}[j]$. 

<table>
<thead>
<tr>
<th>i</th>
<th>SA</th>
<th>rotation matrix</th>
<th>BWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27</td>
<td>#GATTACAT$GATA C$GATTAGAT A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>$GATTACAT$GATTAGATAGATAC T</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>$GATTAGATA$GATTACAT$GAC T</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>A#GATTACAT#GATTACAT$GATTAGAT</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>ACAT$GATTACAT$GATTAGATAGATAC T</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>ACAT$GATTAGATA$GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>AGATA#GATTACAT#GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>AT$GATTACAT$GATTAGATA#GATTAC T</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>AT$GATTAGATA$GATTACAT#GATTAC T</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>ATAGATATGAC GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>ATACAT#GATTAGATA#GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>ATTACAT#GATTACAT$GATTAGATAGATAC T</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>19</td>
<td>ATTAGATA#GATTACAT$GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>CAT$GATTACAT$GATTAGATGATAC T</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>CAT$GATTAGATA$GATTACATGATAC T</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>23</td>
<td>GATAGATATGAC GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>GATAGATA#GATTACAT$GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>GATTACAT#GATTACAT$GATTAGATA#</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>18</td>
<td>GATTAGATA#GATTACAT$GATTACATGATAC T</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>T$GATTACAT$GATTAGATA#GATTAC T</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>16</td>
<td>T$GATTAGATA#GATTACAT#GATTAC T</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>25</td>
<td>TA#GATTACAT#GATTACAT$GATTAGA T</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>TACAT$GATTACAT$GATTAGATA#GATAGA T</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>12</td>
<td>TACAT$GATTAGATA#GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>21</td>
<td>TAGATA#GATTACAT$GATTACATGA T</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>3</td>
<td>TTACAT$GATTACAT$GATTAGATA#GATAGA T</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>20</td>
<td>TTAGATA#GATTACAT$GATTACATGA T</td>
<td></td>
</tr>
</tbody>
</table>
### Corollary

- Let $k \geq 0$ be largest value such that $\text{BWT}[i'] = \text{BWT}[i' + 1]$ for all $i'$ with $\text{SA}[i'] \in [\text{SA}[i] - k + 1 \ldots \text{SA}[i]]$
- Let $\text{SA}[j] := \text{SA}[i] - k$

**Then:**

- $\text{BWT}[j] \neq \text{BWT}[j + 1]$ but still $\text{SA}[i + 1] - \text{SA}[i] = \text{SA}[j + 1] - \text{SA}[j]$.

**why it works:**

for each backward step, we move to the preceding character pair in the text.


$r$-index in $\mathcal{O}(r)$ space

- $S[x]$ : sample at start of $x$-th run
- $E[x]$ : sample at end of $x$-th run

where $x \in [1..r]$, and $r$ is the number of character runs in BWT.

interested in following queries on $E$:

- $E$.pred($p$) : $\max\{q \in E : q \leq p\}$
- $E$.succ($p$) : $\min\{q \in E : q > p\}$

for that: build predecessor and successor data structure on $E$
\( \mathcal{E}.\text{pred}(S[x + 1]) \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>( S[x] )</th>
<th>( \mathcal{E}[x] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

\[ \phi^{-1} \text{ graph} \]
- each entry of \( \mathcal{E} \) is a node
- create an arc from \( \mathcal{E}[x] \) to \( \mathcal{E}[y] \) if \( \mathcal{E}[y] = \mathcal{E}.\text{pred}(S[x + 1]) \)

(sorted \( \mathcal{E} = [1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27] \))
\[\mathcal{E}.\text{pred}(S[x + 1])\]

\[x \quad S[x] \quad \mathcal{E}[x]\]

<table>
<thead>
<tr>
<th>(x)</th>
<th>(S[x])</th>
<th>(\mathcal{E}[x])</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

\[\phi^{-1}\] graph

- each entry of \(\mathcal{E}\) is a node
- create an arc from \(\mathcal{E}[x]\) to \(\mathcal{E}[y]\) if \(\mathcal{E}[y] = \mathcal{E}.\text{pred}(S[x + 1])\)

(sorted \(\mathcal{E} = [1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27]\))
(label the arc of \( x \)-th run with cost \( c_x \) and limit \( \ell_x \))

\[
\begin{align*}
\mathcal{E}.pred(S[x+1]) \\
\hline
x & S[x] & \mathcal{E}[x] & c_x & \ell_x \\
1 & 27 & 27 & 5 & 1 \\
2 & 4 & 9 & 22 & 3 \\
3 & 4 & 7 & 15 & 1 \\
4 & 23 & 24 & 19 & 2 \\
5 & 4 & 6 & 23 & 0 \\
6 & 10 & 10 & 10 & 0 \\
7 & 1 & 1 & 1 & 0 \\
8 & 18 & 18 & 4 & 1 \\
9 & 4 & 8 & 25 & 0 \\
10 & 4 & 4 & 4 & 0 \\
11 & 12 & 12 & 12 & 0 \\
12 & 21 & 21 & 21 & 0 \\
13 & 1 & 3 & 20 & -2
\end{align*}
\]

\( c_x := S[x + 1] - \mathcal{E}.pred(S[x + 1]) \)

\( \ell_x := \mathcal{E}.succ(\mathcal{E}[x]) - \mathcal{E}[x] \)

(sorted \( \mathcal{E} = [1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27] \))
to compute $SA[i + 1]$ from $SA[i]$

1. start node is $p := \mathcal{E}.\text{pred}(SA[i])$
2. initial cost $c_0$ is $SA[i] - p$
3. stop if accumulated cost $c$ is above limit of arc
4. add cost of arc to $c$ and move to next node $v$
5. report $SA$ value $v + c$
6. goto 3.

example

- $SA[1] = 27, p = 27, c \leftarrow c_0 = 0$
- arc $(27, 4)$ has limit $1 > c \Rightarrow$ move to $4$
- add cost $5$ of arc $(27, 4)$ to $c \leftarrow c_0 + 5 = 5$
- $SA[2] = 9$ is node label $4$ plus cost $5$
to compute $SA[i + 1]$ from $SA[i]$

1. start node is $p := E.\text{pred}(SA[i])$
2. initial cost $c_0$ is $SA[i] - p$
3. stop if accumulated cost $c$ is above limit of arc
4. add cost of arc to $c$ and move to next node $v$
5. report $SA$ value $v + c$
6. goto 3.

example
- are at node 4 with accumulated cost $c = 5$
- arc (4, 12) has limit 6 > $c \Rightarrow$ move to 12
- $SA[3] = 17$ is node label 12 plus cost 5
to compute $SA[i + 1]$ from $SA[i]$

1. start node is 
   $p := \mathcal{E}.\text{pred}(SA[i])$
2. initial cost $c_0$ is 
   $SA[i] - p$
3. stop if accumulated cost $c$ is above limit of arc
4. add cost of arc to $c$ and move to next node $v$
5. report $SA$ value 
   $v + c$
6. goto 3.

example
- are at node 12 with accumulated cost $c = 5$
- arc (12, 21) has limit $3 < c \Rightarrow$ stop
- continue with $SA[3] = 17$
- $p = \mathcal{E}.\text{pred}(17) = 15$, $c_0 = 2$. 

to compute $SA[i + 1]$ from $SA[i]$

1. start node is $p := E.\text{pred}(SA[i])$

2. initial cost $c_0$ is $SA[i] - p$

3. stop if accumulated cost $c$ is above limit of arc

4. add cost of arc to $c$ and move to next node $v$

5. report $SA$ value $v + c$

6. goto 3.

example

- $p = E.\text{pred}(17) = 15, c \leftarrow c_0 = 2$
- arc $(15, 23)$ has limit $3 > c \Rightarrow$ continue
- add cost 1 of arc $(15, 23)$ to $c \leftarrow c_0 + 1 = 3$
- $SA[4] = 26$ is node label 23 plus cost 3
to compute $SA[i + 1]$ from $SA[i]$

1. start node is $p := E.p\text{red}(SA[i])$
2. initial cost $c_0$ is $SA[i] - p$
3. stop if accumulated cost $c$ is above limit of arc
4. add cost of arc to $c$ and move to next node $v$
5. report $SA$ value $v + c$
6. goto 3.

Further speedup
- for each traversed arc, we can omit a predecessor query on $E$
- if such traversable paths are long, we add shortcuts
  $\Rightarrow$ build $\phi^{-1}$ trees on long paths
\[ \phi^{-1} \text{ tree} \]

The diagram illustrates a perfect binary tree with labels indicated on the nodes and arrows showing the direction of the tree structure. Each node is labeled with an ordered pair, indicating its position in the tree. The labels are as follows:

- Node 27: (5, 1)
- Node 4: (0, 6)
- Node 12: (0, 3)
- Node 21: (2, 1)
- Node 1: (0, 3)
- Node 18: (4, 1)

The arrows show the connections between the nodes, indicating the path from the root to the leaves.
$\phi^{-1}$ tree

- Identify label of out-going arc with node itself.

![Diagram of a tree with labels showing the relationships between nodes.](image-url)
$\phi^{-1}$ tree

- identify label of out-going arc with node itself
- fill up path with dummy arcs having label 0, $-1$ (=untraversable)
$\phi^{-1}$ tree

- identify label of out-going arc with node itself
- fill up path with dummy arcs having label 0, $-1$ (=untraversable)
- build perfect binary tree on path by partitioning it
\( \phi^{-1} \) tree

- Identify label of out-going arc with node itself.
- Fill up path with dummy arcs having label 0, \(-1\) (=untraversable).
- Build perfect binary tree on path by partitioning it.
- Label of internal node is based on the label of its children as below:

\[
c_1 + c_2, \min(\ell_1, \ell_2 - c_1)
\]

\[
c_1, \ell_1 \quad c_2, \ell_2
\]
querying $\phi^{-1}$ tree

1. climb up until exceeding the limit at a node
2. climb down to leaf at which we exceed the limit the first time
querying $\phi^{-1}$ tree

1. climb up until exceeding the limit at a node
2. climb down to leaf at which we exceed the limit the first time

example

1. start at 27 with cost 0
2. climb up to 7, $-4$ with limit $-4 < 0$
3. take cost of left child and descend to 0, 3
4. return 12 with cost 5, having skipped 4
experiments

dataset
- Chromosome 19 sequences from the 1000 Genomes Project

machine
- AMD EPYC 75F3 32-core processor
- 512 GB of RAM
- 64-bit Linux

alternative solutions
- standard $r$-index, Gagie+’18
- $sr$-index, Cobas+’21
- RLCSA, Mäkinen+’10
Chromosome 19

ϕ⁻¹ forest has no clear advantage
introduced $\phi^{-1}$ forest

- $O(r)$ space data structure on top of $r$-index
- provides random access to $SA$
- query time depends on
  - length of a run
  - values of costs and limits

open problems

- only trivial bound known
  - $O(\log r)$ time per $\phi^{-1}$ tree traversal
  - $O(\log \log_w(n/r))$ time for a predecessor query

- need theoretical analysis of number of predecessor calls

Thank you for listening. Any questions are welcome!
how to compute $SA[i + 1]$ from $SA[i]$ with the $r$-index:

- if $SA[i] = E[x]$ for an $x \in [1..r]$
  \[ \Rightarrow SA[i + 1] = S[x + 1] \]

- otherwise, take $p = E[pred(SA[i])]$, and let $j$ be such that $SA[j] = p$

- apply toehold lemma:

- we obtain $SA[j + 1]$ from the above case ($SA[j] \in E$)
computing $SA[i + 1]$ from $SA[i]$ is actually an application of

$$\phi^{-1}[SA[i]] := \begin{cases} 
1 & \text{if } i = n, \\
SA[i + 1] & \text{otherwise}
\end{cases}$$

then:

- $\phi^{-1}(E[x]) = S[x + 1]$
- take $p = E\cdot\text{pred}(SA[i])$, and let $j$ be such that $SA[j] = p$

$$\phi^{-1}(SA[i]) = SA[i + 1] = \phi^{-1}(p) + SA[i] - p$$
from $SA[i]$ to $SA[i + 1]$

- $\phi^{-1}(E[x]) = S[x + 1]$
- take $p = E.p\text{red}(SA[i])$, and let $j$ be such that $SA[j] = p$
- $\phi^{-1}(SA[i]) = \phi^{-1}(p) + SA[i] - p$

example

- $i = 2$, $SA[2] = 9$ is known
- $4 = E.p\text{red}(9)$
- $\phi^{-1}(4) = 12$
  $$\Rightarrow SA[3] = 12 + 9 - 4 = 17$$

(sorted

$E = [1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27]$)
from \( SA[i] \) to \( SA[i + 1] \)

- \( \phi^{-1}(\mathcal{E}[x]) = S[x + 1] \)
- take \( p = \mathcal{E}.\text{pred}(SA[i]) \), and let \( j \) be such that \( SA[j] = p \)
- \( \phi^{-1}(SA[i]) = \phi^{-1}(p) + SA[i] - p \)

**example**

- \( i = 3, \ SA[3] = 17 \) is known
- \( 15 = \mathcal{E}.\text{pred}(17) \)
- \( \phi^{-1}(15) = 24 \)

\[ \Rightarrow \ SA[4] = 24 + 17 - 15 = 26 \]

(sorted

\( \mathcal{E} = [1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27] \))
Observation
- # predecessor queries is bounded by length of run
⇒ practically slow for long runs

Can we compute $m$ iterations of $\phi^{-1}$ faster?
suppose $\text{SA}[i] = \mathcal{E}[x]$, and let
\begin{itemize}
  \item $\mathcal{E}[x] := \mathcal{E}.\text{pred(SA}[i])$
  \item $\mathcal{E}[y] := \mathcal{E}.\text{pred(SA}[i + 1])$
\end{itemize}
then: $\phi^{-1}(\text{SA}[i]) = \text{SA}[i + 1] = \mathcal{S}[x + 1]$
\[= \mathcal{E}[y] + \mathcal{S}[x + 1] - \mathcal{E}[y]\]
\[= \mathcal{E}[y] + c_x\]
where $c_x := \mathcal{S}[x + 1] - \mathcal{E}.\text{pred(}\mathcal{S}[x + 1])$ is the cost of $x$-th run.

definition
\begin{itemize}
  \item $i = 1$, $\text{SA}[i] = \mathcal{E}[1] = 27$,
  \item $\mathcal{S}[2] = 9$, $\mathcal{E}[y] = \mathcal{E}.\text{pred(}\mathcal{S}[2]) = 4$
  \item $c_1 = \mathcal{S}[2] - \mathcal{E}[y] = 5$; $\phi^{-1}(27) = 4 + 5 = 9$
\end{itemize}
suppose $SA[i + 1] \notin \mathcal{E}$.

Then:

$$\phi^{-1}(SA[i + 1]) = \phi^{-1}(\phi^{-1}(SA[i]))$$
$$= \phi^{-1}(S[x + 1])$$
$$= \phi^{-1}(\mathcal{E}[y] + c_x)$$

apply toehold lemma for $\mathcal{E}.\text{pred}(S[x + 1]) = \mathcal{E}[y]$:

$$\Rightarrow \phi^{-1}(\mathcal{E}[y] + c_x) = \phi^{-1}(\mathcal{E}[y]) + c_x$$

finally, $\phi^{-1}(\mathcal{E}[y]) = S[y + 1] = \mathcal{E}[z] + c_y$,

where $\mathcal{E}[z] := \mathcal{E}.\text{pred}(S[y + 1])$

total: $\phi^{-1}(SA[i + 1]) = \mathcal{E}[z] + c_x + c_y$

given $\mathcal{E}[w] := \mathcal{E}.\text{succ}(\mathcal{E}[y])$, we assumed that

$$c_x = S[x + 1] - \mathcal{E}[y] < \mathcal{E}[w] - \mathcal{E}[y] = \ell_y,$$

where $\ell_y := \mathcal{E}.\text{succ}(\mathcal{E}[y]) - \mathcal{E}[y]$ is the limit of the $y$-th run.
recursive application

- given $SA[i]$ with $E.$pred($SA[i]$) = $E[x_1]$,
- let $c_0 := SA[i] - E[x_1]$ and
- $x_1, x_2, \ldots, x_m$ be the indices of the runs we visit, such that
  $E.$pred($E[x_j] + \sum_{k=1}^{j-1} c_{x_k} + c_0$) = $E[x_{j+1}]$ for all $j \in [2..m - 1]$

then:

$$\phi^{-m}(SA[i]) = S[x_m + 1] = E[x_m] + \sum_{k=1}^{m-1} c_{x_k} + c_0$$

conclusion:

- just need to sum up $\sum_k c_{x_k}$
- but check that sum does not exceed the limit
- can translate this to a path problem on a directed labeled graph
Pizza&Chili experiments

\( \phi^{-1} \) forest faster with negligible memory overhead