plcpcomp

Patrick Dinklage
Jonas Ellert
Johannes Fischer
Dominik Köppl
Manuel Penschuck
<table>
<thead>
<tr>
<th>scheme</th>
<th>method</th>
<th>ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>LZ76</td>
<td>unidirectional</td>
<td>Lempel,Ziv '76</td>
</tr>
<tr>
<td>longest first</td>
<td>grammar</td>
<td>中村 + '09</td>
</tr>
<tr>
<td>Icpcomp</td>
<td>bidirectional</td>
<td>Dinklage+ '17</td>
</tr>
<tr>
<td>LZ-LFS</td>
<td>hybrid</td>
<td>Mauer+ '17</td>
</tr>
<tr>
<td>自己参照なし</td>
<td>without self-ref</td>
<td>藤原 + '19</td>
</tr>
<tr>
<td>unicomp</td>
<td>unidirectional</td>
<td>峰松 + '19</td>
</tr>
</tbody>
</table>
bidirectional compression

- replace substrings by references
- in any direction
- allow self-references
- no cycles
lcpcomp

- replace longest reoccurring substring
- recurse

Starting from position 4, copy 3 symbols.
Starting from position 1, copy 2 symbols.
cycles

- need restriction to prevent cycles
- rule: reference to *lexicographically smaller* suffix
~ LZ77

- LZ77
  - reference to *previous text position*

- lcpcomp
  - reference to *lexicographically smaller* suffix

=> both greedy + reference order constraint
multiple choices
order not clear
plcpcomp: tie break

1 2 3 4 5 6 7 8 9

(7,3) an an an an

take **leftmost** one
how?

find

• longest re-occurring substring $S$
• $S$ = longest common prefix (LCP) of two suffixes
• $|S| = PLCP[i]$
• $S$ starts at argmax $PLCP[i]$
• $PLCP[i] = LCP$ of $i$-th suffix and $\Phi[i]$-th suffix
\( \Phi \) array

\[ \Phi[i] := SA[ISA[i] - 1] \]
main idea

- PLCP: length of factor
- \( \Phi \): reference
computation
discard, since previous position's PLCP is higher
No peak > 5
maximum peak
factor of length 5
PLCP[i]
2 definitions

PLCP[i]

interesting peaks

maximum peaks
recurse

found first maximum peak

apply plcpcomp on this prefix

apply plcpcomp on this suffix

linear time?

only examine interesting peaks!
• new maximum peak is interesting peak
• update interesting PLCP of peaks
left overlap of length 1
shrink peak by overlap length
new highest peak
factor of length 3
right overlap
move peak and shrink to what’s left
PLCP[i]
new maximum peak
factor of length 2
finished!
data structures
data structures & algorithms

- **BGone**: 後藤, 坂内 '14
- **IM-Φ**: Kärkkäinen+ '09

\[ O(n) \text{ time} \]
sparse $\Phi$/PLCP
sparse Φ/PLCP

|sparse Φ| = #runs in BWT =: r

[Kärkkäinen+ '16]
complexity

- time: $O(n)$
- space: PLCP + bit vector

$\max(r \lg n + 3n + o(n) + (O(n \lg n) + 1)^{0.5} \lg n, n \lg n)$

sparse $\Phi$

construct $\Phi$

maintain interesting peaks
external memory

• # of I/Os
  • \( \text{scan}(n) = \Theta(n/B) \)
  • \( \text{sort}(n) = \Theta( (n/B) \log_{M/B} (n/B)) \)
execution

given text, PLCP, Φ

• **scan** PLCP
  • create tuples <j, PLCP[j]>
  • *sort*

• **scan** Φ
  • create tuples <j, PLCP[j], Φ[j]>

• **scan** text
  • create output

3 scan + 1 sort
summary

plcpcomp

- bidirectional compression scheme
- algorithmic improvement to lcpcomp
- linear scan of text, PLCP, Φ
- => works in EM
- **fastest** solution to compress file in EM with ratio ~ LZ77

Outlook

- bounds for #factors?
- bounds for maintaining interesting peaks
- algorithmic optimization
- more variants
  - using Φ⁻¹
  - unidirectional