Appendix to preprint entitled “Hierarchical Overlap Graph”

Bastien Cazaux, Eric Rivals

March 5, 2018

Contents

1 Difference in number of nodes between EHOG and HOG

Consider a finite alphabet, say $\Sigma = \{a, c, g, t\}$. Let s be any word formed by a permutation of Σ. Here, s could be $s := acgt$. The length of s is the cardinality of Σ.

Let z be a positive integer. We build the following instance P_z by taking :

- the word w made of the concatenation of z copies of word s,
- and $(|\Sigma| - 1)$ other words, which are $(|\Sigma| - 1)$ cyclic shifts of w, denoted $w_1, \ldots, w_{(|\Sigma|-1)}$.

For a letter α and a word v, the (first) cyclic shift of the word αv is $v\alpha$. In our construction,

- w_1 is the cyclic shift of w,
- w_2 is the cyclic shift of w_1,

  $\ldots$
- w is the cyclic shift of $w_{(|\Sigma|-1)}$.

Note that $|P_z| = |\Sigma| = |s|$ and that $||P_z|| = |P_z| \times |w| = |P_z| \times z \times |s| = |P_z|^2 \times z$

For an instance P_z,

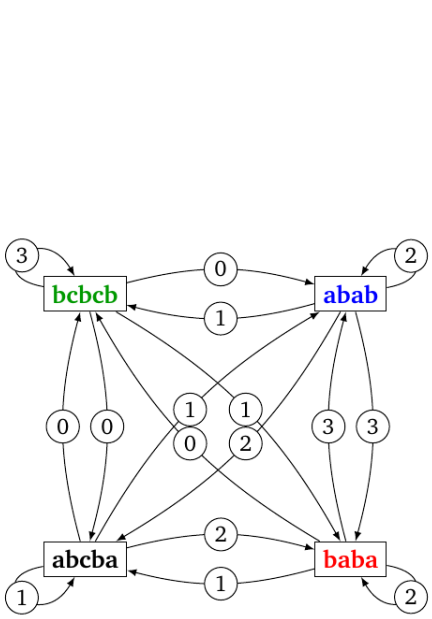
- the $EHOG(P_z)$ contains exactly as many nodes as the Aho-Corasick automaton contains states, that is $||P_z||$ nodes (leaves included). Indeed, all internal nodes are overlaps.
- The $HOG(P_z)$ contains $|P_z|^2$ internal nodes and $|P_z|$ leaves.

Let us denote the number of nodes of the EHOG and of the HOG respectively by $|EHOG(P_z)|$ and $|HOG(P_z)|$. Then the ratio

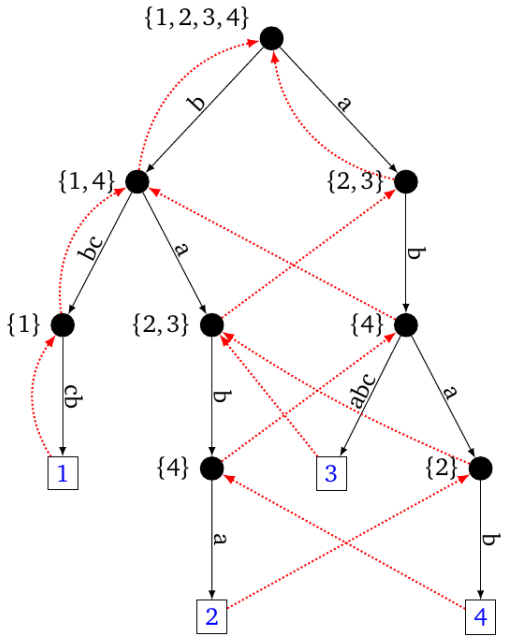
$$\begin{aligned}
\frac{|EHOG(P_z)|}{|HOG(P_z)|} &= \frac{||P_z||}{|P_z|^2 + |P_z|} && (1)\\
&= \frac{z \times |P_z|^2}{|P_z|^2 + |P_z|} && (2)\\
&= \frac{z}{1 + \frac{1}{|P_z|}} && (3)\\
&\rightarrow_{z\rightarrow\infty} +\infty. && (4)
\end{aligned}$$

2 An example of HOG construction

Consider the instance $P := \{bcbcb, baba, abcba, abab\}$. Its OG and EHOG are given in the figure below.The HOG is exactly the EHOG, since all internal nodes of the EHOG are in fact maximal overlaps for some pairs of words of P.



(a) Overlap Graph (OG) of P



(b) Both $EHOG(P)$ and $HOG(P)$

node	R_ℓ	C(before)	C(after)	Specific pairs	bHog
bcb	{1}	0000	1000	(1,1)	1
bab	{4}	0000	0001	(4,2)	1
ba	{2,3}	0001	0111	(2,2) (3,2)	1
b	{1, 4}	1000 ^ 0111			
b	{1, 4}	0000	1001	(4,1) (1,2)	1
aba	{2}	0000	0100	(2,4)	1
ab	{4}	0000 ^ 0100			
ab	{4}	0000	0001	(4,3) (4,4)	1
a	{2,3}	0001	0111	(2,3) (3,3) (3,4)	1
root	{1,2,3,4}	1001 ^ 0111			
root	{1,2,3,4}	0001	1111	(1,3) (1,4) (2,1) (3,1)	1

(c)

Figure 1: (a) and (b) OG and EHOG for instance $P := \{bcbcb, baba, abcba, abab\}$. In the EHOG, `goto` transitions appear in black arcs, `Failure Links` in dotted red arcs. For each internal node, the list R_L is given between bracket. (c) Trace of Algorithm MarkHOG. For each internal node are shown: the word it represents, R_l, the bit vector C when before and after the node is processed, bHog, and the pairs for which it is a maximum overlap. All nodes of the EHOG also belong to the HOG (as shown by the values of bHog being 1).