Graph Compression by BFS

Avinash Amballa
{amballaavinash@gmail.com}

1. Introduction :

This paper introduces a compression scheme for information that combines efficient storage with fast retrieval. The scheme takes advantage of the WebGraph's properties without assuming an ordering of the URLs, whereas others considered lexicographic ordering of URLs.

This method's specific goal is to generate a compressed graph that can support queries like:

1) For two input pages X and Y, does X have a hyperlink to page Y?

2) For input page X, list the neighbours of X

Web graph can be treated as a directed graph, and we represent this using *G = (V, E)* . For any node $v \in V$, $A_v = \{u_1, u_2, ..., u_k : u_i \in V, (v, u_i) \in E\}$ denotes the adjacency list of node v.

2. BFS vs lexicographic :

One way to assign indices from the standpoint of compression is to sort the URLs lexicographically and then assign each page its corresponding rank in the ordering. This results in two properties :

• Locality: Any of the neighbours of a node i might have an index close to i.

• Similarity: Pages with a lexicographically close index are likely to share many neighbours.

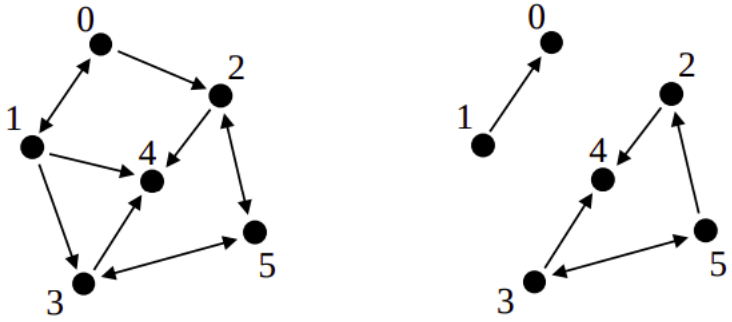
The method described in this paper involves ordering nodes based on the graph's Breadth First Search (BFS) rather than lexicographic order, while also maintaining these characteristics.

3. Encoding :

Step 1 : In this step we perform a BFS of G. When expanding a node $v_i \in V$, we assign consecutive integer indices to its k_i neighbours (not yet expanded) , and store the value of k_i .Once the traversal is completed, all of the links in the breadth-first tree are encoded in the traversal list T : $\{k_1, k_2, ..., k_{|V|}\}$.

This traversal almost removes $|V|$-1 links from the graph.

Note that the compression ratio obtained in this step is affected by the BFS indices assignment.



Here T ={2, 2, 1, 0, 0, 0}

Step 2 : Each compressed chunk C of l nodes (l is compression level) $v_i, ..., v_{i+l-1}$, is prefixed by the sequence $k_i, ..., k_{i+l-1}$ in its compressed representation.

The adjacency list A_i of each node v_i in a chunk C is encoded in increasing order. Each encoding consists of the integer gap between adjacent elements and a type indicator from the set $\{ \alpha, \beta, \chi, \phi \}$. Here, A_i^j denotes the jth element in A_i. The three main cases are as follows:

1. $A_{i-1}^j \leq A_i^{j-1} < A_i^j$: the code is the string $\phi \cdot (A_i^j - A_i^{j-1} - 1)$
2. $A_i^{j-1} < A_{i-1}^j \leq A_i^j$: the code is the string $\beta \cdot (A_i^j - A_{i-1}^j)$
3. $A_i^{j-1} < A_i^j < A_{i-1}^j$: this splits in two sub cases, namely,
   (a) if $A_i^j - A_i^{j-1} - 1 \leq A_{i-1}^j - A_i^j - 1$ then the code is the string $\alpha \cdot (A_i^j - A_i^{j-1} - 1)$
   (b) otherwise the code is the string $\chi \cdot (A_{i-1}^j - A_i^j - 1)$

When A_{i-1}^j does not exist ,replace it with A_k^j , where k ($k < i - 1$ and $v_k \in C$) is the closest index to i for which the degree of v_k is not less than j . If even such a node does not exist use ϕ-type code.

Node	Degree	Links
...	...	...
i	8	13 15 16 17 20 21 23 24
$i+1$	9	13 15 16 17 19 20 25 31 32
$i+2$	0	
$i+3$	2	15 16
...	...	...

Node	Degree	Links
...	...	...
i	8	ϕ 13 ϕ 1 ϕ 0 ϕ 0 ϕ 2 ϕ 0 ϕ 1 ϕ 0
$i+1$	9	β 0 β 0 β 0 β 0 χ 0 α 0 β 2 ϕ 5 ϕ 0
$i+2$	0	
$i+3$	2	β 2 α 0
...	...	...

Step 3 : Because two adjacent nodes in the graph share many neighbours, the adjacency lists may lead to of four types of "redundancies". These can be removed by,

1. Consecutive identical rows are encoded by giving a multiplier to the first row in the sequence.
2. As there are intervals of similar node degrees, the degrees of consecutive nodes are

gap-encoded.

3. Whenever there is a sequence of at least L_{min} identical elements in a row (eg: block of ϕ 1's), then this sequence is run-length encoded.

4. A box of identical rows (eg: biggest block in the table) of at least A_{min} is run-length encoded.

Note : Redundancies are removed in the order in which they are listed above, and we choose the largest one if there is more than one box beginning at the same entry.

To indicate 3rd or 4th redundancy, we use a special character Σ , followed by a Σ_F denoting whether the redundancy starting with this element is of type 3 ($\Sigma_F = 2$), type 4 ($\Sigma_F = 3$), or both ($\Sigma_F = 1$).

Degree	Links									
...	...									
0										
9	β 7	ϕ 1	ϕ 1	ϕ 1	ϕ 0	ϕ 1	ϕ 1	ϕ 1	ϕ 1	
9	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 2	
10	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 1	ϕ 903
10	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 223	ϕ 900
10	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 1	α 0
10	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 1	β 0
10	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 1	β 0
10	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 1	β 0
10	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 1	β 0
10	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	α 76	α 232
9	β 0	β 1	β 0	β 0	β 0	β 0	β 0	β 0	β 0	
...	...									

Lines	Degree	Links			
...	...	...			
0	0				
0	9	β 7	$\phi\Sigma$ 2 1 1	ϕ 0	$\phi\Sigma$ 2 1 2
0	0	$\beta\Sigma$ 3 0 7 5	β 1	$\beta\Sigma$ 2 0 4	β 2
0	1	β 1	ϕ 903		
0	0	β 223	ϕ 900		
0	0	β 1	α 0		
3	0	β 1	β 0		
0	0	α 76	α 232		
0	-1	β 0			
...	...	...			

Type 3 : "ϕ Σ 2 1 1", ϕ : ϕ-type encoding, 2 : value of Σ_F , 1 : the gap, and 1 : the no.of times the element appears - L_{min} (= 2).

Type 4 : "β Σ 3 0 7 5", β - β-type encoding, 3 : Σ_F , 0 : gap, 7 : width - 1, and 5 : height - 2.

Step 4 : We do not need to explicitly write ϕ characters, which are implicit in $A^j_{i-1} \leq A^{j-1}_i$. The characters α, β and χ as well as Σ_F are encoded by Huffman code. Gaps, and remaining integers are encoded using the *pi*-code. Also gap g could be negative (as with degrees), so we encode 2g if g is positive, and $2|g| - 1$ when $g < 0$.

Say, n is a positive integer with binary representation b and h = 1 + $\lfloor log_2(n) \rfloor$. We represent n using k+h+$\lceil \frac{h}{2^k} \rceil$-1 bits, k is a positive integer. Here, h = 2^kl-c ($l > 0$ and $0 \leq c < 2^k$), then the π_k -encoding of n is obtained by writing the unary representation of l, followed by the k bits required to encode c, and finally by the rightmost h-1 bits of b.

Example, the π_2 - encoding of n=43 :

binary representation b of 43 : 101011, h = 6 = 2^2·2-2 (l=2 and c=2), unary representation of l = 2 : 01, Value of c = 2 (k-bits): 10, h-1 least significant digits of b : 01011. Thus, the encoding of 43 is 01 10 01011.

The expected length of n is:

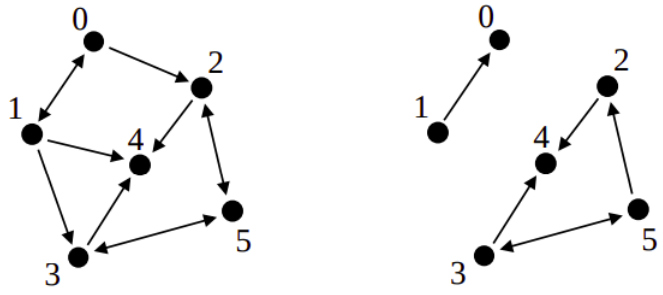
$$\begin{aligned} E_P(L_\pi) &\leq E_P(k+h+\lceil \tfrac{h}{2^k} \rceil - 1) \\ &\leq E_P(k+h+\tfrac{h}{2^k}) \\ &\leq E_P(k+log_2 n+1+\tfrac{log_2 n+1}{2^k}) \\ &\leq 1+k+\tfrac{1}{2^k}+(1+\tfrac{1}{2^k})E_P(log_2 n) \\ &\leq 1+k+\tfrac{1}{2^k}+(1+\tfrac{1}{2^k})H_P \end{aligned}$$

$$\frac{E_P(L_\pi)}{H_P} \leq \frac{k}{H_P}+\frac{(1+H_P)}{H_P}(1+\tfrac{1}{2^k})$$

$\lim_{k\to\infty}\lim_{H_P\to\infty}\frac{E_P(L_\pi)}{H_P} = \lim_{k\to\infty}(1+\frac{1}{2^k})$ =1

4. Code to check if the directed link (v_i, v_j) exists :

a=$1+\sum_{h=0}^{h=i-1} k_h$
if $a \leq j < a+k_i$:
    return true
if $j \geq a+k_i$:
    return false
A_i = the adjacency list of v_i
if $v_j \in A_i$:
    return true
else :
    return false



Here T ={2, 2, 1, 0, 0, 0}

We can enforce a quick query because of the underlying BFS in Step 1 to check whether or not a link (v_i, v_j) exists.

References: Apostolico, Alberto, and Guido Drovandi. "Graph compression by BFS." Algorithms 2.3 (2009): 1031-1044.