ISSN 1512-1232

UDC 512.1

AN ALGORITHM TO COMPUTE THE LOGICAL ENTROPY OF A DIRECTED GRAPH

Guram Kashmadze

Computer science departament, Iv. Javakhishvili Tbilisi State University, Tbilisi, Georgia. gkashmadze@yahoo.com

Abstract

By the concept of logical entropy of a directed graph we mean the logical entropy of graph partition into strongly connected components. Algorithms for working with graphs are fundamental for computer science because many interesting computational problems are formulated in terms of graphs. Many algorithms that work with graphs begin with decomposition a graph into its strongly connected components. After decomposition such algorithms explore separately each one and then combine local solutions into global solution according to the structure of a graph. Partition a directed graph into strongly connected components has many applications in different areas. The notion of logical entropy, that is based on a partition logic, gives us the ability to compare different partitions of a graph into components. In this article we suggest an algorithm to compute the logical entropy of partition of a directed graph into strongly connected components. This algorithm is based on well-known one, which finds strongly connected components by using the Depth-First-Search twice. The algorithm presented here can be regarded as a new application of DFS.

Keywords: *Directed graph, strongly connected components, logical entropy.*
2010 Mathematics Subject Classification: 05C20.

I. INTRODUCTION

David Ellerman developed the dual *logic of partitions* [1], defined the notion of *logical entropy* and describes its properties [2]. Let *U* be the finite non-empty universe. A partition π of *U* is a set of non-empty disjoint subsets of *U* whose union is *U*. A *distinction* or *dit* of a partition π is an ordered pair (u, v) of elements $u, v \in U$ that are in different blocks of the partition. The set of distinctions of a partition π is its dit set $dit(\pi)$.

$$dit(\pi) = \bigcup_{B,B' \in \pi,\ B \neq B'} B \times B' = U \times U - \bigcup_{B \in \pi} B \times B$$

David Ellerman defines the logical entropy $h(\pi)$ of a partition π as follows

$$h(\pi) = \frac{|dit(\pi)|}{|U \times U|}$$

The partitions of *U* are partially ordered by refinement, which is the inclusion of *dit* sets. Let Δ be the diagonal of the $U \times U$. All the possible distinctions $U \times U - \Delta$ are the dits of **1**, that is the top.

$$h(\mathbf{1}) = \frac{|dit(\mathbf{1})|}{|U \times U|} = \frac{|U \times U - \Delta|}{|U \times U|} < 1$$

Thus, if we want $h(\mathbf{1})$ to be 1, we must define the logical entropy of the partition π in the following manner

ISSN 1512-1232

$$h(\pi) = \frac{|dit(\pi)|}{|U \times U - \Delta|} = \frac{|dit(\pi)|}{|U| \cdot (|U| - 1)}$$

II. LITERATURE REVIEW

M. Behara and P. Nath [3] considered countable measurable partitions of a Lebesgue probability space. A partition π is called a refinement of σ, written as $\sigma \subset \pi$, if every element of σ is a disjoint union of elements of π. Behara and Nath defined the entropy of order *a* of the partition that in a particular case is well-known Shannon's entropy.

D. A. Simovici and S. Jaroszewicz [4] link the notion of partition of a finite set to the notion of probability distribution. If $\pi = \{B_1, \ldots, B_n\}$ is a partition of *A*, then the probability distribution attached to π is $(p_1, \ldots, p_n)$, where $p_i = |B_i|/|A|$ for $1 \le i \le n$. They consider the notion of entropy of a partition via the entropy of the corresponding probability distribution and present an axiomatization of a generalization of Shannon's entropy starting from partitions of finite set.

C. Cao, Y. Sui, and Y. Xia [5] define a graph (G, V) which represents the refinement relation between partitions of a non-empty universe *U*. The vertex of a graph is attached to the certain partitions and the edge connects a vertex to its refinement. According to the values of the information entropy of partitions a directed graph $(G, \vec{V})$ is defined on (G, V). It is proved that entropy is non-decreasing along the path from a vertex with the minimal entropy to one with the maximal entropy in $(G, \vec{V})$.

M. Dehmer and A. Mowshowitz [6] describe methods for measuring the entropy of graphs and demonstrate the wide applicability of entropy measures. It is mentioned that entropy measures play an important role in a variety of problem areas, including biology, chemistry, and sociology.

III. AN ALGORITHM TO COMPUTE THE LOGICAL ENTROPY OF THE STRONGLY CONNECTED COMPONENTS

We assume that the vertices of a directed graph G are $V = \{1, 2, \ldots, n\}$. *color*, π, *d*, and *f* are vectors of length *n* with components *color*[*i*], $\pi[i]$, *d*[*i*], *f*[*i*], *i* = 1, 2, … , *n* .
In the CLRS [7] is given the algorithm to find strongly connected components of a directed graph:

Strongly-Connected-Components(*G*)
1. call DFS(*G*) to compute finishing times *f*[*i*] for each vertex $i \in V$.
2. compute G^T.
3. call DFS(G^T), but in the main loop of DFS, consider the vertices in order of decreasing *f*[*i*] (as computed in line 1).

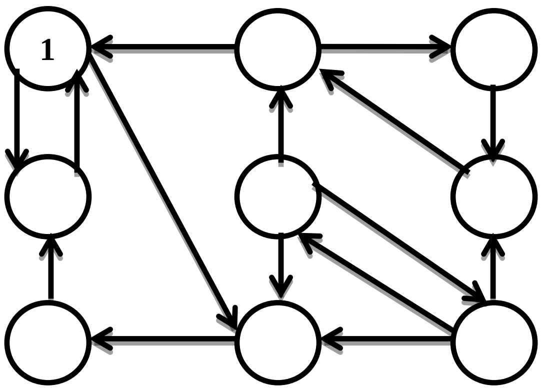
In order to compute the logical entropy of strongly connected components we continue the steps 1 – 3 given above, as follows. Note that the values of vectors π, *d*, and *f* are considered as computed by the second call of DFS.

Let csour be a new vector of strongly connected components sources
m=0
***for** i=1 to n*
    ***if** $\pi[i] == NIL$*
        m = m+1
        csour[m] = i
Let csiz be a new vector of strongly connected components sizes
    ***for** i = 1 to m*

ISSN 1512-1232

j = *csour*[i]
csiz[i] = (f[j] – d[j] + 1)/2
sum = 0
for i = 1 *to* m − 1
for j = i+1 *to* m
sum = *sum* + *csiz*[i] · *csiz*[j]
logicalEntropy = 2 · *sum* / (n ·(n −1))

As an example, consider the folowing graph.



A partirion π of the set of vertices $V = \{1,2,3,4,5,6,7,8,9\}$ into strongly connected components is

$$\pi = \{\{5,9\},\{2,3,6\},\{1,4,7,8\}\}. \ |dit(\pi)| = 2 \cdot (2 \cdot 3 + 2 \cdot 4 + 3 \cdot 4) = 52.$$

Logical entropy:

$$h(\pi) = \frac{|dit(\pi)|}{|V|(|V|-1)} = \frac{52}{9 \cdot 8} \cong 0{,}72.$$

IV. ANALYSIS OF THE SUPPLEMENTARY STEPS

After execution of the steps 1 – 3 the depth-first forest is formed. According to the theorem 22.16 in CLRS the Strongly-Connected-Components procedure correctly computes the strongly connected components of the directed graph G provided as its input. The vertices of the depth-first tree in G^T that are rooted at i (i.e. at the vertex that has the property $\pi[i] == NIL$) form exactly one strongly connected component. Thus, at the end of the first ***for*** loop the value of the variable m gives the number of the strongly connected components of a graph G and the vector *csour* is filled with indices of the components sources in increasing order.

During processing any depth-first tree DFS starts at source vertex and moves “deeper” in the graph whenever possible. In each step DFS explores edges out of the most recently discovered vertex. Once all of vertex i’s edges have been explored, search “backtracks” to explore edges heaving the vertex from whch i was reached. This process ends at source vetex. Every vertex j has been assigned a discovery time d[j] and a finishing time f[j]. The initial value of a time in DFS is 0. Before any assignment the value of a time increases by 1. At each vertex j the time increases twice: before assignment to d[j] and before assignment to f[j]. Therefore, the number of vertices in given tree is equal to
(f[j] − (d[j] − 1))/2 assuming that j is a source vertex. Thus, at the end of the second ***for*** loop the vector *csiz* is fiiled with sizes of components in order of increasing indices.

In order to compute the number of dits we start with first component and multiply the size of component by size of all succeeding components. At the end of the third ***for*** loop the value of variable *sum* gives a total sum of all such products. To get the correct number of dits we must double this sum for the sake of symmetry (if a pair (u, v) is a dit, then a pair (v, u) is a dit also).

ISSN 1512-1232

In last line we compute the logical entropy.

V. COMPLEXITY OF SUPPLEMENTARY STEPS

In the worst case every strongly connected component is a singleton. Therefore a value of the variable *m* is equal to *n* = $|V|$. The first and the second ***for*** loops get executed *n*-times each. The total number of times the third and the fourth nested ***for*** loops get executed equals $n-1+n-2+\cdots+1=n(n-1)/2$. Thus the complexity of the supplementary steps in the worst case is $O(n^2)$.

VI. CONCLUSION.

The different notions of a graph entropy reflect different properties of graphs. Let us consider non-probabilistic notions of graph entropy.

The automorhism group of a graph *G* = (*V*, *E*), denoted by *Aut* (*G*), is the set of all adjacency preserving bijections of *V* (see[8]). Let $\{V_i|\ 1 \leq i \leq k\}$ be the collection of orbits of *Aut* (*G*), suppose $|V_i| = n_i,\ 1 \leq i \leq k$. The entropy or information content of *G* is given by the expression below:

$$I_a(G) = -\sum_{i=1}^{k} \frac{n_i}{n} log \frac{n_i}{n}.$$

As mentioned in [8], this measure captures the symmetry structure of a graph. The lower the information content, the greater the symmetry.

The logical entropy of the strongly connected components of a directed graph characterize the distingvishability of graph vertices. Zero logical entropy means that the graph is strongly connected, all vertices belong to the same component and are not distinguishable. If each vertex forms a distinct component, and hence every pair (*u*, *v*), *u*≠*v* is distinguishable, then the logical entropy equals 1.

M. Dehmer and A. Mowshowitz [6] write that identification and classification of structural configurations in networks pose challenging problems for which entropy measures have proven useful. Further development of the theory of entropy measures and progress in designing efficient algotithms for computing entropy are needed to meet this challenge.

Given above algorithm can be considered as a step in this direction. It makes possible to compare directed graphs by means of distinguishability of vertices.

ISSN 1512-1232

References

1. David Ellerman,"The logic of partitions: Introduction to the dual logic of subsets." *Review of Symbolic Logic,* 3(2), pp. 287-350, June 2010.
2. David Ellerman, "An introduction to logical entropy and its relation to Shannon entropy." *International Journal of Semantic Computing,* 7 (2), pp. 121-145, 2013.
3. M. Behara, P.Nath. "Information and entropy of countable measurable Partitions." *Kybernetika.* Vol.10, No. 6, pp. 491-503.
4. D. A. Simovici, S.Jaroszewicz. "An axiomatization of partition entropy." *IEEE Transaction on Information Theory,* 48 (7) pp. 2138-2142.
5. C. Cao, Y. Sui, and Y. Xia, "The graph–theoretical properties of partitions and information entropy." In *Proc. 10th International Conference, RSFDGrC 2005,* Vol. 3641 of the series *Lecture Notes in Computer Science.* pp. 561-570.
6. M. Dehmer, A. Mowshowitz. "A history of graph entropy measures." *International Journal of Information Sciences,* Vol. 181, Issue 1, pp. 57-78, 2011.
7. T.H. Cormen, C.E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms.* 3rd ed., Massachusets, USA, The MIT Press, 2009.
8. A. Mowshowitz, V. Mitsou, "Entropy, orbits, and spectra of graphs." in Analysis of complex networks: from biology to linguistics. (Eds. M. Dehmer and F. Emmert) ch.1. Wiley–VCH Verlag, Weinheim. Germany. 2009.

Article received: 2016-06-11