A Role for Chordless Cycles in the Representation and Retrieval of Information

John L. Pfaltz
Dept. of Computer Science
University of Virginia

ABSTRACT

This paper explains how very large network structures can be reduced, or consolidated, to an assemblage of chordless cycles (cyclic structures without cross connecting links), that is called a trace, for storage and later retreival. After developing a basic mathematical framework, it illustrates the reduction process using a most general (with directed and undirected links) network.

A major theme of the paper is that this approach appears to model actual biological memory, as well as offering an attractive digital solution.

Keywords

Closure, biological memory, reduction, consolidation, recall, trace, subgraph matching, directed graphs

1. INTRODUCTION

A central tenet of database theory is that it is the relationships between various entities that constitute real information. It is the foundation of the relational model [7]. But, in those situations where there are millions of entities and the relationships are relatively sparse, the familiar array style representation of data simply won't work. We turn to graph, or network, type representations. In other situations, such as social network analysis, the data is naturally graph structured. In any case, we are concerned with the retreival of specific "chunks" of the stored information.

We will contend that retrieval of information from a storage medium is not a single, unified operation; that there are at least two distinct phases. First the desired information must be identified and located within the storage medium. We call this "information access" and address it in Section 3. Then it must be read out or, in the case of biological memory, reconstituted. This latter step we call "recall" and discuss it in Section 4.

However, this paper is less concerned with actual retrieval than discovering graph structures which facilitate this process. In Section 2.3 we introduce the concept of "chordless cycles" which we believe play a prominent role in the representation of biological information, and which we believe can be exploited in computer applications as well. In Section 2.1 we sketch the mathematical foundations for this notion, which is based on concepts of "closure". In Section 2.2 we present computer code that will reduce any network to its constituent chordless cycles. Then in Section 2.3 we can describe the desirable properties of representing information by these cycles and indicate their role in biological memory.

Our goal in this paper is to use some relatively abstract graph-theoretic concepts to effectively model, and bring together, both biological and computer applications.

2. REPRESENTATION AND STORAGE OF INFORMATION

Our basic understanding of the world, whether visual, oral, or tactile, is neural in nature. The representation of data in regular arrays is, to a large extent, an artifact of computer architecture and the ease of algebraic manipulation. So we begin with the assumption that all "information" is a graph structure of some form.

To our knowledge, there has been no agreement as to what neural configurations correspond to any specific empirical sensations or mental concepts; but there have been many studies documenting that all perception and cognition correlate with neural activity, even detailing its occurrence with specific locations within the brain [11, 16, 15, 18, 21, 39, 40]. However, we are fairly confident that whatever neural networks do correspond to specific sensations, concepts or knowledge, they are very large — probably in the millions of elements, or possibly even billions, as discussed in [43]. We are talking about a very large data space!

The most general mathematical model of neural activity appears to be a directed graph. So the fundamental assumption of this paper is that, at its most basic level, information retrieval involves locating and obtaining rich directed graph structures based on simpler representations which serve as query keys. We are not retrieving array-type information to display on a spread sheet.

For this paper we assume that the network structure is the "information", independent of any edge or node labels.

2.1 Information as Relationships

Information and knowledge are essentially relational. Elements, or neurons, are related to other elements. To our knowledge, neurons are not "labeled". In contrast, most subgraph matching algorithms assume their nodes, or elements, are labeled [8, 24, 43]. These labels can be used to prune

in plain graph search procedures [8], or the basis for hash-based join algorithms [41, 42]. Because we assume unlabeled and unweighted edges, this paper can be regarded as an exercise in pure graph theory. However, we believe it provides insights into both biological and computer retrieval.

There seem to be many different kinds of relationships; however, we abstract them all by a single relational symbol, η. By $y.\eta$ we mean the set of all elements related to y. For set valued operators, such as η, we use a suffix notation. It helps remind us that they are not scalar valued functions. If we represent information as a network $\mathcal{N}$, $y.\eta$ would denote the neighbors of y in $\mathcal{N}$. So one can read $y.\eta$ simply as y's neighbors. (For mathematical convenience we assume η is reflexive, so $y \in y.\eta$.) Visually, η represents the edges of the graph, or relational network. (We use the terms node and element and graph and network interchangeably.) It is unknown whether information networks are directed or undirected. To assume a measure of generality we will assume they are mixed, with some links being directed and some undirected. While Figure 1 is far too small to be a real information network, we will use it to illustrate concepts of this paper. Arrow heads denote directed edges; if there are

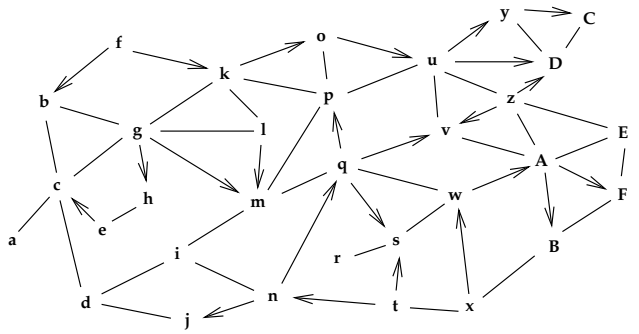

Figure 1: A representative "information network".

none the link is bi-directional, or symmetric.

An important principle for interpreting information networks is the concept of "closure". Closure is well defined in mathematics [6, 26, 22]. There are many different kinds of closure operators; the most intuitive is the convex hull of geometric figures [2]. We will use the fundamental relational operator, η, to define a closure operator, φ. Specifically,

$$y.\varphi = \{z | z.\eta \subseteq y.\eta\} \tag{1}$$

that is, z is an element of y closure, $y.\varphi$, if all elements related to z are also related to y. (φ can be extended to arbitrary sets, Y, by $Y.\varphi = \cup_{y \in Y}\{y.\varphi\}$.) The concept of neighborhood closure underlies the entire development of this paper. In the network of Figure 1, $c.\varphi = \{abc\}$, $p.\varphi = \{op\}$ and $g.\varphi = \{bgl\}$. It is worth convincing yourself why this is so. For a more detailed development see [34, 37, 38], where it is shown that computing closure, φ, is a local operation.

2.2 Consolidation

As defined by (1), an element z in y closure, $y.\varphi$, is relationally connected to no more elements than y itself. Consequently, those elements within a network which are contained in the closure of other elements contribute very little to the information content conveyed by the network. They can be combined with little loss of information. We say that y has *subsumed* z, or equivalently that z *belongs* to y. By $y.\beta$ we mean all nodes, z that belong to y, that is have been combined. Readily $y \in y.\beta$. The pseudo code for a process we call "reduction", and denote by ω, is shown below in Figure 2. The outer loop terminates when every element y is

```
while there exist subsumable nodes
  {
  for_each y in N
     {
     get {y}.nbhd
     for_each z in {y}.nbhd - {y}
        {
        if ({z}.nbhd contained_in {y}.nbhd
           {        //  z is subsumed by y
           combine z with y
           add z to {y}.beta
           }
        }
     }
  }
```

Figure 2: Pseudo code of the reduction, ω, process.

itself a closed set. A network with this property is said to be *irreducible*. A C^{++} version of this code has been applied to a variety of social networks [33, 34]. The actual code is "set based" where constructs such as *y.nbhd* and *y.beta* are implemented as bit strings. Thus operations are effectively O(1). There is no *apriori* constraint on set size, but we have not executed operations with sets of cardinality $> 50,000$. It can be rigorously demonstrated that the reduction of a network, $\mathcal{N}$, which we denote by $\mathcal{T} = \mathcal{N}.\omega$ is *unique* (upto isomorphism). ω is a well-defined function over the space of all networks. We call $\mathcal{T}$ the *trace* of the network $\mathcal{N}$.

Figure 3 illustrates the trace $\mathcal{T}$ of the network $\mathcal{N}$ of Figure 1. Emboldened solid links, or edges, connect the nodes that

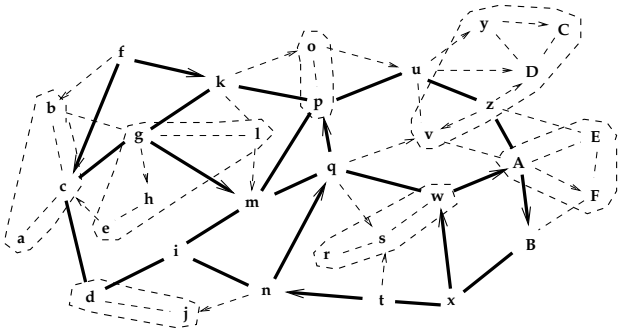

Figure 3: A network $\mathcal{N}$ with mixed relational links. Solid links represent those retained after reduction by ω.

remain in $\mathcal{T}$ after the reduction process; thiner dashed lines connect the nodes that have been combined with others.. Dashed lines enclose the 7 non-trivial β-sets; for example $c.\beta = \{a, b, c\}$ and $z.\beta = \{v, y, z, D, C\}$.

Observe that a new link (f, c) was created to preserve path connectivity through b. We also see that φ and β are distinct operators since although $c.\varphi = c.\beta$, $g.\varphi \subset g.\beta$.

Of the original 32 nodes, only 17 remain after reduction

which represents a considerable storage compaction.

The reduction process, ω, has a number of desirable computational properties. Because the closure operator, φ, is local (it need only interrogate adjacent elements), it can be executed as a parallel process. (The code of Figure 2 that we have actually used is sequential[1] , but it is evident how to convert the outer loop.) This is particularly important for biological information storage and retrieval. The parallel application of a similar closure based process within the visual pathway is described in [36].

We have devoted considerable space to a description of ω because, in addition to preparing information for storage, we believe it plays a prominent role in retrieval as described in the next sections.

In the literature devoted to human "memory", there has been considerable research suggesting that there is a process that converts short-term memory into long-term memory. It is usually called "consolidation" [1, 4, 9, 25, 28]. It appears to be quite similar to the reduction process described above, whence the subsection heading is "consolidation".

2.3 Chordless Cycles

A chordless cycle is most easily visualized as a string of pearls with no cross connections. More precisely, a *chordless cycle* is a sequence $< y_1, y_2, \ldots, y_n, y_1 >$ of elements $y_i, 1 \leq i \leq n$ of length $n \geq 4$ where there exist no links (chords) of the form (y_i, y_k) $k \neq i \pm 1$, $i, k \neq 1, n$.[2] It is better to just think of them as paths with no single edge cross connections. In Figure 3, the sequence, or path, $< c, g, m, i, d, c >$ is a chordless cycle of length 5 The sequence $< n, q, p, u, z, A, B, x, t, n >$ is a chordless cycle of length 9.

If the relation η between elements is symmetric, it can be proven [34, 35] that if $\mathcal{N}$ is irreducible (*i.e.*, every node is a closed set) then every node y is either (1) isolated, (2) an element of a chordless cycle of length ≥ 4, or (3) an element on a path between two such chordless cycles. Thus the trace (consolidation or reduction) of a symmetric relational network is a collection of interlinked chordless cycles. In Granovetter's analysis of social networks [19], these are the "weak ties".

When η is symmetric, ω readily preserves path connectivity within $\mathcal{T}$. It also preserves the distance between elements as measured by shortest paths [35], together with the center of $\mathcal{N}$ as determined by distance which will be found in $\mathcal{T}$, as will those centers as defined by "betweenness" [5, 13, 14].

Symmetry of the relation η is a powerful property. But, to better model real relationships we have relaxed this constraint throughout to allow directed (non-symmetric) connections. Under these conditions the preceding characterization of irreducible networks is no longer valid. For instance, the node f in Figure 3 is not an element of a chordless cycle.

In the case of non-symmetric networks we must ensure path connectivity through subsumed nodes. Let y subsume z, *i.e.* $z.\eta \subseteq y.\eta$. For all x such that $z \in x.\eta$ we ensure that $y \in x.\eta$. This is always true when η is symmetric, but required the creation of the edge (f, c) in Figure 3. In this case, we can show [38] that if $\mathcal{N}$ is irreducible then for all y, if there exists $z \in y.\eta, z \neq y$, there must exist a path through z terminating in a chordless cycle of length ≥ 4. Even when the relation is non-symmetric, chordless cycles dominate the reduced representation.

Are chordless cycles really fundamental in the representation of biological information?

We can provide no definitive answer to that question. We can point out that protein polymer molecules composed of chordless cycles exist in every cell of our bodies [45]. One example is a 154 node phenylalaninic-glycine-repeat (nuclear pore protein), $\mathcal{N}$, which is shown in Figure 4. This is not at

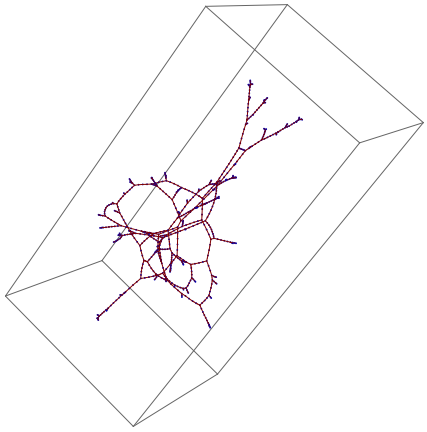

Figure 4: A 3D rendering of a 154 node protein polymer molecule.

all like the dense network of Figure 1. Nevertheless, one can easily see the chordless loops, with various linear tendrils attached to them. When these are removed by ω, there were 107 remaining elements involved in the chordless cycle structure.

If we count [10, 23] the numbers of chordless cycles of length n in the reduction of Figure 4, we obtain the distribution of Figure 5. The average cycle length is 44.9 in this

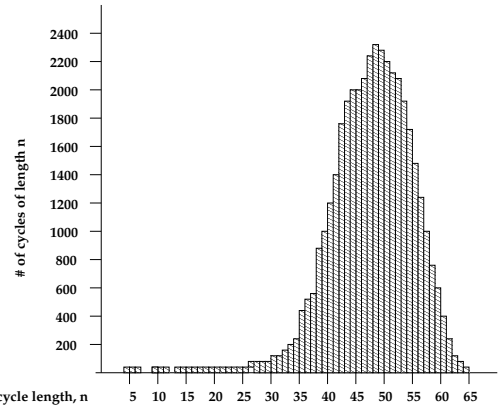

Figure 5: The distribution of chordless cycles in a 107 element network.

network; the modal length is 48; and the 4 longest chordless cycles have length 65. It is clear that the combinatorial possibilities based on cycle lengths alone would permit the representation of a considerable amount of information.

These are only indications that chordless cycle structures might be an important component of relational information

[1]It can be shown that worst case sequential behavior is $O(n^2)$, but actual performance is much better with a maximum of 6 iterations to reduce 1,000 node real networks.

[2]A graph with no chordless cycles is said to be *chordal*. There is an extensive literature on chordal graphs, *e.g.* [27]

representation. The following sections will show how they can be involved with the retrieval process.

3. INFORMATION ACCESS

Information cannot be retrieved unless it is first identified and located within storage. Pointers and URLs identify storage locations. But, lacking these, one must search based on the "content" of the desired information. An early, and still common, mechanism is to attach key words or hash tags to the information, which are then used to create an easily searchable index [3, 17, 30, 31], in the case of hashed storage, to directly control the storage location [12] or to perform hash-joins in relational databases [42].

As access speeds have increased so dramatically, both of the preceding access techniques have often been superceded by simple linear search and comparison.

But, this still presumes that the "content identification" is based on specific terms, or tokens. If important information is based on relationships, as we have asserted in Section 1, then such token based search is rather limited.

Suppose we wish to retrieve within a relational network. We may assume that the search key, $\mathcal{K}$, is itself relational. It might look like Figure 6, but very much larger!

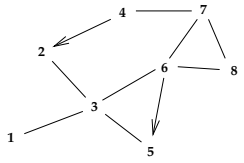

Figure 6: A structural search key, $\mathcal{K}$.

Reduction of $\mathcal{K}$ by ω yields its trace $\mathcal{K}.\omega$, as shown in Figure 7. It is a simple cycle on the nodes $< 4, 3, 6, 7, 4 >$.

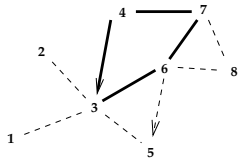

Figure 7: Its reduction, $\mathcal{K}.\omega$.

Comparing Figure 7 with the trace of Figure 3, we see the obvious subgraph matching: $3 \leftrightarrow m$, $4 \leftrightarrow g$, $6 \leftrightarrow p$ and $7 \leftrightarrow k$. In this small example there is only one possible matching chordless cycle. In the kinds of large networks we envision this is not so simple!

Imagine a 1,000 node seach key which reduces to a key trace $\mathcal{K}.\omega$, of say, 600 nodes. To find a matched subgraph in a reduced network of several million nodes is a combinatorialy impossible task. All subgraph matching algorithms, known to the author, assume some form of node and/or edge labeling to initiate the search location and to control the expanding search [8, 24, 43]. Various forms of auxilliary indices provide direct access to graph elements matching those labels. However, while the retrieval of information from a graph structured data store implicitly assumes a labeled graph, we have focused in this paper on the relationships embodied in the network itself. There are no data labels.

Imagine "sliding" an unlabeled 600 irreducible subgraph around over an irreducible million node network to find a match, if one exists.

A method that we are beginning to explore involves the network providing information about its own irreducible structure. To do this we label each edge with the length of the "shortest" chordless cycle of which it is a member. This is not an artificial data label, but an element of the network representation itself. Thus, in the irreducible network of Figure 3, the edge (q, p) would be labeled with a 7 because it is a member of the chordless 7-cycle $< q, p, k, g, c, d, i, m, q >$. This same edge is a member of longer cycles such as the 9-cycle, $< q, p, u, z, A, B, x, t, n, q >$, but we only label with respect to the shortest.

The edge (m, q) would be labeled with a 4, as would the edge (m, p). Each edge in the reduced search key, $\mathcal{K}.\omega$, of Figure 7 must be labeled with a 4.

Each edge label of the search key must be exactly equal to the corresponding edge label in the larger network. So, based solely on the edge labeling, the only possible matches for this reduced key of Figure 7 in the reduced network of Figure 3 are the 4-cycles $< g, m, p, k, g >$, $< n, q, m, i, n >$ and $< w, A, B, x, w >$.

Edge labeling with respect to shortest chordless cycle length eliminates fruitless search expansion, but it can't indicate likely places to begin the comparison. The central retrieval problem is identifying likely nodes within the larger network to initiate the search comparison.

We believe that the solution lies with the imbedded "triangles". Even though they are not chordless cycles of length ≥ 4 of which an irreducible network is comprised, they do exist, provided each edge is a member of a distinct chordless cycle of length ≥ 4. In the reduced network of Figure 3 there is one embedded triangle, that is $< q, p, m, q >$, with associated cycle labels $< 7, 4, 4 >$. There is no embedded triangle in the reduced search key of Figure 7. This is unusual; but it is because the search key is "too small". Our observation is that most reduced networks of 20 nodes, or more, have at least one embedded triangle. The network of Figure 4 has five, with associated cycle lengths of $< 8, 6, 4 >$, $< 12, 5, 5 >$, $< 12, 7, 4 >$, $< 14, 10, 7 >$ and $< 16, 8, 6 >$. If the reduced search key contains any embedded triangle, it must match one of these.

In our JMC approach to information access, we 1) associate with each edge (functional relationship) in η the length of the shortest chordless cycle to which it belongs; and 2) create an external index of shortest length triples corresponding to embedded triangles. Assuming the search key is sufficently well specified to include an embedded triangle, we first search this index of triples to identify one, or more, plausible search regions within the multi-million node reduced network constituting the data store.

This access mechanism is now being implemented, so its actual efficency is still unknown. We believe it has considerable promise. More uncertain is whether the edge labeling and triangle index can be maintained in real time as the network changes dynamically [34]. Our hope is that if the network changes are *continuous* [32], then there exist effective update procedures.

The approach described in this section seems quite appropriate to biological recall as well. We see "Mary" in the market. This is a visual experience involving many relationships. A rapid, almost instantaneous, search through our

memory yields a much larger relational information structure, including her name, her age and the names of her 3 children. This is often called semantic memory in the literature [16].

4. RETRIEVAL OF RELATIONAL INFORMATION

We suppose that a reduced trace, $\mathcal{T} = \mathcal{N}.\omega$, or a portion of it, has been identified as in the preceding section. During the reduction process, ω, used to consolidate the network information we need not store only the resulting trace. We can also represent various steps in the reduction process. Our own code, for example, records $y.\beta$ for each retained node y. Thus, if the trace of Figure 3 was accessed given the key of Figure 7 then we would know that $c.\beta = \{a, b, c\}$. Moreover, we know the order in which a and b were subsumed by c.

From this, a very close approximation of the original network of Figure 1 can be recreated. But, there will still be some loss of information. For instance, the connection between f and b would be lost. More complete information could be stored with each subsuming node, though at some increased storage expense. Alternatively, the complete information network, $\mathcal{N}$, could be stored, with the trace network, $\mathcal{T}$, separately stored as an easily searchable index.

It is our belief that with information structures of this size, retrieval of close approximations, *i.e.* the trace itself, will be sufficient for most applications.

4.1 Biological Recall

Biological recall is likely to be somewhat different from a largely deterministic computer recall. There is some evidence that biological organisms only store an abbreviated trace and that the memory, as we experience it, is somehow "recreated" [20]. This helps explain why human memories can be distorted in specific detail, yet correct in their overall structure.

In [38], the author describes a semi-random expansion process, ε, that fills in subsumed nodes to create a richer network $\mathcal{N}'$. In many respects, ε is an inverse operator to ω. Given a trace, $\mathcal{T}$, $\mathcal{T}.\omega^{-1}$ defines the collection of all networks $\{\mathcal{N}^*\}$ such that $\mathcal{N}^*.\omega = \mathcal{T}$. ε constructs a single network, $\mathcal{N}'$, within this collection. Thus, given an initial network, $\mathcal{N}$, ε will "retrieve" $\mathcal{N}'$ which may, or may not, be (isomorphic to) $\mathcal{N}$. However, we are assured that $\mathcal{N}'.\omega = \mathcal{T} = \mathcal{N}.\omega$, or $\mathcal{N}.\omega.\varepsilon.\omega = \mathcal{N}.\omega = \mathcal{T}$.

Such a semi-random "retrieval" process may be completely inappropriate in computer applications, but it seems to model biological recall rather well. Our memories often are confused with respect to detail, even when they they are generally correct. It also supports the notion of "re-consolidation" which asserts than long-term memories are repeatedly rewritten, unless deliberately distorted in our (semi)conscious mind [28, 29, 44].

5. SUMMARY

The representation of complex networks by a trace comprised of chordless cycles has a firm, well defined basis. We know that there exists an effective procedure ω to reduce a network $\mathcal{N}$ to its trace $\mathcal{T}$ by local and easily parallelizable code, such as might be found in the visual pathway. It is also known that such chordless cycle assemblages exist as protein polymers in all the cells of the body, including synapses.

Clearly, there are significant advantages, in terms of many fewer nodes and links, to representing network data in this fashion. Readily, real information stores, whether computer generated or biological, may have differentiated (*i.e.* labeled) nodes and links. Nevertheless, this first treatment of the problem as a purely abstract, graph theoretic issue has value. It provides a theoretical basis for more applied work.

We have suggested that subsets of the stored information can be identified and retreived by similarly reducing the search key. Then, in Section 3, using graph-theoretic properties arising from the fact that irreducible networks consisting of chordless cycles, we have sketched an access method that might support retrieval based solely on relational structure. In biological organisms, this or an associative memory approach may be used.

The next to last paragraph of Section 4.1 mentions a reconstruction process ε which functions as a kind of inverse operator to ω. This may be the most important observation of the paper. We are just beginning to explore it. If there is a way, given a trace $\mathcal{T}$, to reconstruct a network $\mathcal{N}'$ which closely approximates the original network $\mathcal{N}$, this could have profound implications for both computed and biological applications.

6. REFERENCES

[1] Sir Fredrick Bartlett. *Remembering: A Study in Experimental and Social Psychology*. Cmbridge Univ. Press, Cambridge, UK, 1932. Paperback version, 1967.

[2] Jon L. Bentley. Approximation Algorithms for Convex Hulls. *Comm. ACM*, 25(1), Jan. 1982.

[3] Christian Böhm. A Cost Model to Query Processing in High Dimensional Spaces. *ACM Trans. on Database Sys.*, 25(2):129–178, June 2000.

[4] Clive R. Braham and Eloucine Messaoudi. BDNF function in adult synaptic plasticity; The synaptic consolidation hypothesis. *Progress in Neurobiology*, 76(2):99–125, 2005.

[5] Ulrik Brandes. A Faster Algorithm for Betweeness Centrality. *J.Mathematical Sociology*, 25(2):163–177, 2001.

[6] G. Burosch, J. Demetrovics, G.O.H. Katona, D.J. Kleitman, and A. Sapozhenko. On the Number of Databases and Closure Operations. *Theoret. Comput. Sci.*, 78(2):377–381, Jan. 1991.

[7] E. F. Codd. A Relational Model for Large Shared Data Banks. *Comm. ACM*, 13(6):377–387, June 1970.

[8] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Analysis and Machine Intell.*, 26(10):1367–1372, 2004.

[9] Robert G. Crowder. *Principles of Learning and Memory*. Psychology Press, New York, 2015.

[10] Elisângela Silva Dias, Diane Castonguay, Humberto Longo, and Walid A. R. Jradi. Efficient enumeration of chordless cycles. *arXiv:1309.1051v4*, 24 Nov. 2014.

[11] Yadin Dudai. *The Neurobiology of Memory: Concepts, Findings, Trends*. Oxford University Press, Oxford, 1989.

[12] R. J. Enbody and H. C. Du. Dynamic hashing schemes. *COMPSUR*, 20(2):85–113, June 1988.

[13] Linton C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41, 1977.

[14] Linton C. Freeman. Centrality in Social Networks, Conceptual Clarification. *Social Networks*, 1:215–239, 1978/79.

[15] J.D.E. Gabrieli. Cognitive Neuroscience of Human Memory. *Annual Review of Psychology*, 49:87–115, 1998.

[16] John D.E. Gabrieli, John E. Desmond, Jonathan B. Demb, Anthony D. Wagner, Maria V. Stone, Chandan J. Vaidya, and Gary H. Glover. Functional Magnetic Resonance Imaging of Semantic Memory Processes in the Frontal

Lobes. *Psychological Science*, 7(5):278–283, Seot, 1996.

[17] Volker Gaede and Oliver Günther. Multidimensional Access Methods. *Computer Surveys*, 30(2):170–231, June 1998.

[18] Michael S. Gazzaniga, Richard B. Ivry, George R. Mangun, and Megan S. Steven. *Cognitive Neuroscience, The Biology of the Mind*. W.W. Norton, New York, 2009.

[19] Mark S. Granovetter. The Strength of Weak Ties. *Amer. J. of Sociology*, 78(6):1360–1380, 1973.

[20] Machael E. Hasselmo. *How we remember: brain mechanisms of episodic memory*. MIT Press, Cambridge, MA, 2012.

[21] Michael E. Hasselmo. Encoding: Models linking neural mechanisms to behavior . In H.L. Roediger III, Y. Dudai, and S.M. Fitzpatrick, editors, *Science of Memory: Concepts*, pages 123–127. 2007.

[22] Yannis Ioannidis, Raghu Ramakrishan, and Linda Winger. Transitive closure algorithms based on graph traversal. *ACM Trans. on Database Sys.*, 18(3):512–576, Sept. 1993.

[23] Walid A. R. Jradi, Elisângela S. Dias, Diane Castonguay, Humberto Longo, and Hugo A. D. do Nascimento. A GPU-based parallel algorithm for enumerating all chordless cycles in graphs. *arXiv:submit*, 1288731:1–15, June 2015.

[24] Michihiro Kuramochi and George Karypis. Frequent Subgraph Discovery. In *Proc.IEEE Conf. on Data Mining, ICDM 2001*, pages 313–320, San Jose, CA, 2001.

[25] Joseph E. LeDoux. Consolidation: Challenging the traditional view. In H.L. Roediger III, Y. Dudai, and S.M. Fitzpatrick, editors, *Science of Memory: Concepts*, pages 171–175. 2007.

[26] Norman M. Martin and Stephen Pollard. *Closure Spaces and Logic*. Kluwer Academic, Boston, 1996.

[27] Terry A. McKee. How Chordal Graphs Work. *Bulletin of the ICA*, 9:27–39, 1993.

[28] Lynn Nadel. Consolidation: The demise of the fixed trace . In H.L. Roediger III, Y. Dudai, and S.M. Fitzpatrick, editors, *Science of Memory: Concepts*, pages 177–181. 2007.

[29] Karim Nader and Oliver Hardt. Asingle standard for memory: the case for reconsolidation. *Nature Reviews Neuroscience*, 10:224–234, 2009. doi:10.1038/nrn2590.

[30] Ratko Orlandic and John L. Pfaltz. Compact 0-complete trees: A new method for searching large files. Technical Report IPC TR-88-001, Institute for Parallel Computation, Univ. of Virginia, Jan. 1988.

[31] Ratko Orlandic and Byunggu Yu. A Retrieval Technique for High-Dimensional Data and Partially Specified Queries. *IEEE Trans. on Data and Knowledge Engineering*, 42(1):1–21, Jan. 2002.

[32] John Pfaltz and Josef Šlapal. Transformations of discrete closure systems. *Acta Math. Hungar.*, 138(4):386–405, 2013.

[33] John L. Pfaltz. Finding the Mule in the Network. In Reda Alhajj and Bob Werner, editors, *Intern. Conf on Advances in Social Network Analysis and Mining, ASONAM 2012*, pages 667–672, Istanbul, Turkey, August 2012.

[34] John L. Pfaltz. Mathematical Continuity in Dynamic Social Networks. *Social Network Analysis and Mining (SNAM)*, 3(4):863–872, Dec. 2013.

[35] John L. Pfaltz. The Irreducible Spine(s) of Discrete Networks. In Xuemin Li, Yannis Manolopoulos, Divesh Srivastava, and Guangyan Huang, editors, *Web Information Systems Engineering - WISE 2013*, volume LNCS # 6984, Part 2, pages 104–117), Nanjing, PRC, Oct. 2013.

[36] John L. Pfaltz. The Role of Continuous Processes in Cognitive Development. *Mathematics for Applications*, 4(4):129–152, 2015. DOI:10.13164/ma.2015.11.

[37] John L. Pfaltz. Using Closed Sets to Model Cognitive Behavior. In Tapabrata Ray, Ruhul Sarker, and Xiaodong Li, editors, *Proc. Australian Conf. on Artificial Life and Computational Intelligence (ACALCI 2016)*, volume LNCS 9592, pages 13–26, Canberra, ACT, 2016.

[38] John L. Pfaltz. Two Network Transformations. *Mathematics for Applications*, 6(1):(to appear), 2017.

[39] Dale Purves, George J. Augustine, David Fitzpatrick, and *et al. Neuroscience*. Sinauer Assoc., Sunderland, MA, 2008.

[40] Edmund T. Rolls and Alessandro Treves. *Neural Networks and Brain Function*. Oxford Univ. Press, Oxford, 1998.

[41] Donovan A. Schneider and David J. DeWitt. A performance evaluation of four parallel join algorithms in a shared-nothing multiprocessor environment. In *Proc. 1990 ACM SIGMOD Conf.*, pages 110–121, Atlantic City, NJ, May 1990.

[42] Dennis Shasha and Tsong-Li Wang. Optimizing equijoin queries in distributed databases where relations are hash partitioned. *ACM Trans. on Database Sys.*, 16(2):279–308, June 1991.

[43] Zhao Sun, Hongzhi Wang, Haixun Wang, Bin Shao, and Jianzhong Li. Efficient Subgraph Matching on Billion Node Graphs. *Proceedings of the VLDB Endowment*, 5(9):788–799, 2012. Originally presented at VLDB Conf., Istanbul, Turkey, 2012.

[44] Natalie C. Tronson and Jane R. Taylor. Molecular mechanisms of memory reconsolidation. *Nature Review Neuroscience*, 8(4):262–275, 2007.

[45] Karstan Weis. The Nuclear Pore Complex: Oily Spaghetti or Gummy Bear? *Cell*, 130:405–407, August 10 2007.