Digraphs, Knowledge Hypernets, and Neurons

by

Hendrik O. van Rooyen – University of South Africa
Franz Stetter – University of Mannheim

E-Mail

"HO van Rooyen" <hopete@intekom.co.za>
"Franz Stetter" <stetter@uni-mannheim.de>

Contents

Keywords

Concept-Relationship Knowledge Structure (CRKS), digraphs, hypernets, image, information flow, insight, interpretation, invariance, Knowledge Hypernet (KH), Menger’s theorem, neural network, neurons, noise, relation net, theory of teaching/learning

Digraphs, Knowledge Hypernets, and Neurons.

Abstract

We deal with a current flow network of switches, with input node I and output node O, represented by a directed graph G. In G we define a model of a neuron, and introduce another model in which neurons are theoretically linked. In this second model, we cover invariance, information flow and noise. We show how this model arises from G, how it can be taught, and how it can be declaratively interpreted. The system is made dynamic due to the closing, from O to I, through the environment of the combined models, of a feedback circuit.

1. Introduction

This paper is not intended to replace, but to support, current models of neural networks. It merely provides an alternative model that incorporates some new notions and some relatively new theoretical work. The paper assumes some prior knowledge of relation nets, hypernets, Concept-Relationship Knowledge Structures and their derivatives, Knowledge Hypernets, and some exposure to the theory of these structures. This background can be found in [RGS 2004].

2. System Models

The model is a dual one; a “hardware” model that is a directed graph G = <U, A> and an “interpretation” model that is a hypernet H = <V, E>. We start with G.

The set of vertices U of G can be regarded as a set of switches that pass or stop current. If a vertex is switched on we say it “fires”. The whole model is timed in successive time units of equal duration: a switch that is off can be switched on, i.e. fire, or not during a time unit. A switch/vertex that fires during a given time unit will be off during the next time unit, and will stay off until switched on/fired during a later time unit. The arrows of A carry current: if a vertex is on during a given time unit then all those vertices incident from it are switched on/fire during the next time unit. Each vertex is incident to at least one other vertex, and G is loop free.

We have a non-empty subset $N \subseteq U$ of “nuclei”. A nucleus together with all the vertices incident to it and all the vertices incident from it constitutes our initial model of a neuron. If all the vertices incident to a nucleus fire in a given time unit then the nucleus fires in the next time unit, and all the vertices incident from the nucleus then fire in the following time unit and we say that the neuron is active during those three time units. The set of vertices incident to the nucleus is called the input to the neuron, and the set of vertices incident from it the output of the neuron.

We now refine our model of a neuron in G. A neuron consists of the following.

- A set of input vertices and a nuclear vertex. Each member of the input set is incident to the nucleus.
- An output vertex incident from the nucleus. Together these vertices, input set, nucleus and output vertex, make up the soma.
- An output path, of length ≥ 1, with a terminal vertex. This path is called the axon of the neuron.
- A set of input paths called the dendrites of the neuron. Each dendrite starts at the terminal vertex of some axon and ends at an input vertex of our neuron.

This constitutes our model of a neuron.

The first arrow of a dendrite is called a synapse. The synaptic capacity of the terminal vertex of an axon is the number of synapses at it. Its synaptic weight during a given time unit is the number of those synapses that are active, i.e. carry a current pulse, during that time unit. A neuron may have feedback inasmuch as some of its own dendrites can have synapses at its own axon terminal vertex, which vertex is in its own soma if the axon length is 1.

If all the input vertices of a neuron are active, i.e. “on”, during a given time unit, then the nucleus is activated during the next time unit, and the output vertex during the following time unit. The length of a dendrite to a neuron determines the duration of the flow of activity along its length. Shortest routes will reach a neuron first: a sort of “least action” principle in simple form. With dendrites of different lengths transmitting activity to a neuron from different axon terminal vertices during different time units, we see that a neuron can process incoming activity through just firing or not.

Given a subset $B \subseteq U$, a cascade from B in G, starting with B in time unit 0 for step 0, is called a sweep in G. We define the cascade in such a way that each step n in the cascade takes place during time unit n, starting with step 0, i.e. B only, during time unit 0. Then all the new vertices “found” in step n are fired during time unit n. (Notice that there is no distinction between fast access and limited access cascades in G.) A neuron will be activated in a sweep when all of its input vertices are found, together, in the same step. Each sweep defines a subgraph of G.

This then the “hardware” model G; “hardware” because it can, in principle, be constructed. We now describe the interpretation model. This consists of a hypernet <V, E> in which the vertices are the nuclei of G. <V, E> has, external to it, an input node I and an output node O, and there is a feedback circuit from O to I through the external environment of <V, E>. This constitutes our interpretation model, coupled to the hardware model, so we refer to our models as the G/<V, E> model.

The edges of <V, E > are labelled as follows. Given an edge $\{v_1, v_2\}$, the label consists of v_1 , an “input set” of v_1, i.e. a minimal set of nuclei that must fire to fire v_1, a minimal input set of v_2, and v_2. If the input for v_1 all fire during some time unit, and the input for v_2 all fire during a later time unit, then the labelling is said to be effective and the derivation direction of $\{v_1, v_2\}$ is taken to be from v_1 to v_2 . We consider only effective labellings in <V, E >: no other edges are defined. It is clear that the edge set E of <V, E> changes with time. Note that a vertex of V can be in the input of several different nuclei, and there can be more than one label on an edge at any given time. We can have isolates v in <V, E>: v fires, but there is no edge from v.

3. Primaries, goals, clusters, and invariance

Four of the key facets of learning are trial-and-error, copying, finding invariance in contrast with “noise” by inductive abstraction, and declaration. An invariant is, generally, a characteristic or pattern of characteristics common to a number of otherwise different situations or instances of a situation. We have then, at basic level, play involving trial-and-error and repetition. Satisfaction or reward is paired with invariance while lack of satisfaction or reward is paired with the contrasting noise against which invariance must stand out, so noise is essential.

We now introduce knowledge hypernet (KH) clusters. In what follows we mix vertices of G and vertices of <V, E >, so we must be careful to distinguish them. By a primary vertex/nucleus of V we mean a vertex that is connected to I in the sense that every member of its neuron input is such that there is a path to it, in G, from I. Suppose that we now have an edge $\{v_1 , v_2\}$ from that primary vertex v_1 of <V, E> in <V, E>. We may then be able to define, in <V, E>, a cluster for $\{v_1, v_2\}$, and if such a cluster is repeatedly activated by generally different sweeps from I in G, then we say that that cluster is a primary cluster of <V, E> and that it is invariant in contrast with the noise generated in G, and hence possibly in <V, E>, by the successive sweeps in G. By a goal of V we mean a vertex/nucleus that is connected to O in the sense that its output vertex is such that there is a path from it, in G, to O. If our invariant cluster has a goal in it then it is an invariant subnet of <V, E> that is a KH and is such that any sweep, in G, through it which activates it, induces an information flow in <V, E>. (Notice the difference between a hardware "activation sweep" in G and an interpretational, theoretic, "information flow" in <V, E>.) Clearly non-primary clusters can exist, as can non-"terminal" clusters, i.e. those which are not linked directly to O in G. We can have many invariant clusters in <V, E> with successive sweeps in G, and, as indicated in [RGS 2004], we can associate them, as we will see in the following section, to form KH's that yield information flow from I to O in <V, E>. We should remember that all clusters can be idiosyncratic, as then can any KH that arises by association of them.

4. Inside <V, E>

A KH subnet of <V, E> is called a clustered KH if

Layer 1: the KH has at least one primary cluster as a subnet. Layer 1 consists of a number of primaries of <V, E>, each with its primary cluster, and such that all of those clusters are induced by one initial sweep in G. Layer 1 is complete, and the join of those primary clusters is a KH. Further, layer 1 is the first step in a limited access cascade, in <V, E>, from those primaries. Each non-primary is at deductive distance 1 from the primaries in layer 1.

Layer 2: a cluster in <V, E>, invariantly induced by repeated sweeps in G, is joined to layer 1 if the relevant edge {u, v} of that cluster is such that u is a layer 1 vertex (nucleus) that is "on", and every member of the label of {u, v} is "on" as required, so that the derivation direction of {u, v} is from u to v. Further, all the vertices in the label of {u, v}, all of which are in G and V other than possibly v, must be switched on in all the sweeps that induce layer 1. Layers 1 and 2 together constitute a KH in which every non-primary vertex is at deductive distance 1 or 2 from its primaries: the first two steps in a limited access cascade from its primaries. Again we mention that each cluster induced is, in general, chosen from a number of alternatives, so each is idiosyncratic, as is the resulting KH.

Iterating, we may achieve an invariant, idiosyncratic KH with goal vertices called a completed KH. Every edge {u, v} in it is such that every vertex in the label of the edge, other than possibly v, is primary or “derived”. This will contrast against the noise of successive different sweeps, in G, that have the KH and corresponding information flow in common. Every subnet of <V, E> is a potential KH, but some activated subnets will not be KH’s and some will not be invariant even if they are momentarily sweep induced KH’s. Some activated clusters will be pairwise mutually “associated” in the sense of having non-empty intersections of their respective vertex sets – see [RGS 2004]. From these associations we may find a KH, using a limited access cascade as indicated above, and hence possibly an information flow from I to O through <V, E>. Generally the component KH’s in such an association may not all be clusters, but clusters are important as they are closer to observations than are the edges which induce them. Each KH will generally have some noise attached to it, or associated with it, from the generally different sweeps, in G, that generate the KH through repetition, that noise being activated by the sweeps that form subnets that strictly contain the KH. Noise in <V, E>, then, can be regarded as edges that are activated but do not belong to our KH. In a KH we do not have closed non-derivation paths. We may have closed paths in G. If every vertex in a circuit in G fires in the same time unit we have a hardware disaster. A similar problem will occur if we have unlimited “noise flow” as the result of a sweep in G.

A complete sweep is a sweep that leads to the firing of some goals of <V, E>. For any sweep, we define the appropriate set of edges, if any, of <V, E> for that sweep. We look for a complete sweep that defines a subnet, of <V, E>, that contains a completed KH, and some noise, that is a subnet of the sweep subnet. Repeated sweeps may then make that KH invariant during some interval of time. (A sweep will generally induce arbitrary edges in <V, E>.) Given a sweep, the weight of an edge {u, v} induced by it is the number of u–v derivation paths induced by that sweep. The weight of an edge is at least 1, and changes with succeeding sweeps in general. The higher the weight of an edge the more we say we have "learned at that edge", because the higher the weight of the edge the more information is inherently "stored" in it. The weight of a path in <V, E> is the sum of the weights of its edges, and its capacity is the minimal weight of all its edges.

Repeated sweeps may induce the same edge e ∈ E. For every sweep that induces e we add 1 to the "memory gauge" of e, so the memory gauge of e is the number of sweeps that have induced e at a given time.

If we run a limited access cascade in the subnet of <V, E> that is induced by a complete sweep, and this yields a KH with some of the primaries in the sweep and some of the goals in the sweep, then we have information flow from I to O in that KH, and the rest of that subnet is noise. Repeated sweeps can then make that KH invariant if the sweeps start with differing sets of primaries that have an intersection which induces that KH each time. This will generally increase the memory gauge values of the edges of the KH more than those edges, in the sweep subnets, that do not belong to the KH: noise edges generally have smaller memory gauge values than those of invariant edges. Ideally, the meet of the subnets of those sweeps would be the KH.

In [RGS 2004] we dealt with five modes of reasoning. We have looked briefly at the associative and constructive modes above. Also covered is deductive, or inferential, reasoning by limited access cascade. Intuitive reasoning is modelled by fast access cascade, and here fast access cascades can access noise and subnets not directly associated with a given KH. For example, some vertices may belong to the given KH and also to some other subnet which may or may not be another KH, with derivation paths involving such vertices in one or both subnets but no derivation path passing from one to the other either way. A limited access cascade in the given KH will stay in it, but a fast access cascade in it will penetrate the other subnet if there is a path into it from the KH. In this way we can move from a KH to another subnet that may be a KH or just noise. A fast access cascade can link to noise, or to a subnet that provides insight; those “Eureka” moments. Finally, inductive, or analogical reasoning is modelled by means of KH isomorphism. As we saw above, a fast access cascade could provide a link between “overlapping” KH’s, or even between KH’s separated by noise, providing the stimulus for a search for isomorphic association of (sub-KH’s of) the two KH’s. Notice that in G the induced KH’s in question are referent free, but inherent in their representation in <V, E> are distinct interpretations. See later.

To close this section we point out that any given vertex $v \in V$ can belong to the label of several different edges of <V, E> and of a KH. Those edges constitute the context scheme of v in <V, E>, and the KH if relevant, and deletion of v from <V, E> deletes all those edges. For example, a primary firing repeatedly can be in the label of different edges at different times, if it is also adjacent to the relevant nucleus, a facet of the strong vulnerability of <V, E> - see [RGS 2004].

5. Through <V, E>

In [RGS 2004] we have dealt with Menger’s theorem in a KH with regard to “flow” through a KH. Here we could extend our treatment since we now have capacities for the derivation paths in a KH. Given that only the KH’s in <V, E> can have that flow of activity that we call information flow, from I to O in the critical cases, we can give some meaning to Menger’s theorem in max flow – min cut form for the current KH’s, or a subset of them, in <V, E>. Without pursuing this here, we can claim that one facet of learning consists of associating more invariant clusters into a current KH, or of course of constructing a new KH. Further, it entails increasing flows through invariant KH’s by “finding”, i.e. activating, more paths so as to increase weights, capacities and memory gauge values. Both vertex flow and edge flow can be considered, where the latter is perhaps more relevant here.

In all this, vertex and edge vulnerability in <V, E> play a central role in finding critical affects on the linking of clusters and on information flow. Associated work on complexities, concept schemas and predecessor schemas is also relevant to analysis of the KH’s in <V, E> with regard to learning – see [VGS 2004]. Perhaps the less complex KH’s will be associated with higher memory gauges and “higher” flows in the sense of Menger’s theorem in max flow – min cut form.

The question is one of how all this activity and the associated learning is to be achieved. The answer lies with the notion of a feedback circuit, from I through G/<V, E> to O, and then from O to I through the environment of G/<V, E>. In the following sections we set out some of the criteria for that circuit and environment, and point out some of their consequences.

6. Feedback circuit

To make the G/<V, E> model dynamic we need at least the following characteristics of the feedback circuit through the environment of G/<V, E>, including I and O as conduits that are part of that circuit. We denote the system model by G/<V, E>/F.

- from I through G/<V, E> to O.
- from O, trial-and-error (primitive procedure) and copying/mimicry (primitive declaration), both of which are idiosyncratic, can lead to repeated sweeps and
- invariance of a cluster, or noise, which, in the case of invariance produces
- satisfaction via indirect and direct “rewards”, leading to
- repetition, with some inevitable changes through I producing noise and reinforcing invariance – the cluster loops.
- Iteration links clusters to produce idiosyncratic KH’s, and then information flow from I to O – the completed KH loops, which constitutes

- learning and "understanding" by virtue of declaration of interpretation of those KH's – the learning loops.

All three loops return to I for the next "perceptual stimulus" of G/<V, E>, and then through G/<V, E> to O for the next feedback round.

Learning to recall implies some control of our loops to emphasise invariance through repeated sweeps of G. Instinctive behaviour is genetically built in to G/<V, E>/F, particularly of course in G and F, by natural selection, and may include some completed KH's for it.

Activation flow from I in G can induce KH's in <V, E>, through which that flow is called information flow. If that KH is completed, we speak of complete information flow. Information flow occurs only in KH's that are invariant under successive sweeps of activation flow, and information flow is necessarily contained in activation flow in order that the contrasting noise exists against which invariance is gauged. Information flow can function as one of the gauges of learning.

Information flow conforms with the progress of a limited access cascade through a KH, other activation flows from sweeps being noise flows. Teaching tries to eliminate idiosyncrasies; to establish invariant, repeatable KH's that are currently "correct" according to some opinion. This entails that the teacher guides the feedback from O to I. Induction and analogy are governed by KH isomorphism, while intuition is covered by fast access cascade – see [VGS 2004]. Interpretations of KH's in <V, E> vary with context, and context and understanding within that context depend on "linguistic" declaration, i.e. communication. Thus we may treat interpretation, understanding and communication as different facets of learning and teaching.

7. Some comments on the model system G/<V, E>/F

Clearly some of our real world situation is either not represented or only partially represented in the G/<V, E>/F model. This is always the case; it is scientific method. Only some invariant observations are chosen as the observables to be represented, and only some of the relationships among those observables are represented, in the model. Precisely specified reasoning is then applied to the model, model and reasoning constituting a theory. This is intended to produce some predictions, about the real world situation, from the model. At least some of these should be empirically confirmed, more or less, to support the model and theory. At least one prediction must be empirically falsifiable, so that we can modify or reject the model and theory, thereby learning in the process. The choice of observables and relationships must be "objective", i.e. repeatable.

The G/<V, E>/F model, with the theory developed in [VGS 2004], generates no immediate predictions to test, but it has certain consequences that maybe testable, and a certain "philosophy" to criticise. Its "time unit" supposition is introduced to synchronise with the steps in cascades, and if there really is a time unit in sweeps then it would vary in duration in practice. Memory gauges should be coupled with "least resistance" paths in G/<V, E>/F, using the inverse of the current memory gauge of an edge for example, so that we have, again, a "least action" kind of principle functioning. Investigation of vulnerability, and vulnerability of flow using Menger's theorem, as well as complexity gauges – see [VGS 2004] – should be investigated.

We must remember that paths, in G, from I to primaries in <V, E> and from goals in <V, E> to O, can be of different lengths. Thus KH's can occur anywhere in G/<V, E>. More work needs to be carried out on the details of I, O and the feedback circuit, and on different details of different kinds of neurons.

The dual nature of G/<V, E> seems appropriate. Further the notion of feedback is essential in G/<V, E>/F; without it one cannot have a realistic model. Finally, consider that directed graphs like G are appropriate models for a host of situations in differing fields – see [HNC 1965] for a basic, classic example. If we have a complex loop free digraph we might approach it as follows.

- The vertex base constitutes I.
- The vertex contra-base constitutes O.
- Define neurons and synapses as they occur in it, if any, and the edges of a <V, E> hypergraph.
- Consider invariances, information flow and interpretation in this light.

We may get a fascinating new view of the digraph!

Using <V, E> as an interpretation – see later – and programming tool that incorporates the feedback, one can visualise G as a potentially different kind of "computer" aimed perhaps at structural analysis and learning. The user/programmer would be the environment and control the sweeps through G in accordance with what happens at O and what is desired in <V, E>. Such a "hybrid" invites parallel processing of simultaneous KH's, for example, through fast access cascades, also drawing in "useful noise" against which to judge invariance of a solution.

It is clear that standard computing is required to work with G/<V, E> - see the constructional schemes in [VGS 2004]. Standard computing together with neural network computing and G/<V, E>/F computing would provide a broadening of our range of computing and the domain of computable and simulatable problems.

The real complexity of the model begins with the subgraphs of G, generated during the same time interval by various sweeps, when these "overlap".

To be useful a model must be common to, applicable in, a number of different situations: here different kinds of brain for example. Our model suggests some possible “mechanisms” in learning, which involves G/<V, E>, and teaching, which involves partial control or directing of the feedback from O to I – see [VGS 2004].

8. G/<V, E>/F and images

Consider a given image, or pattern. Controlled sweeps through G build up an overall impression, and sections of the image, as invariant KH’s in an idiosyncratic order, as dictated by feedback and satisfaction, against contrasting noise in the sweeps. Gradually more detail is incorporated, myriads of relationships and their edges are defined. Idiosyncratic edge weights and memory gauge values build up, dictating what is noticed and what is recalled, both in the relevant invariant KH’s and in the “noise” subnets.

The image, or rather generally only parts of it, and some noise with a relatively high memory gauge value, are built up. They are then forgotten later through non-stimulation of the required primaries in later sweeps. The image, or parts of it, or some of the noise could then be recalled if the appropriate subset of primaries is stimulated, perhaps fortuitously or deliberately, in feedback to I.

9. G/<V, E>/F and training/teaching of it

While there may be “genetically programmed” inherent KH’s in <V, E>, the model system needs initial training/teaching. Starting with trial-and-error and examples, this can potentially be accomplished. The examples should be chosen to be different while each having the same relationship structure, and should each have different noise as contrast with that invariant relational structure. For instance, well chosen examples can result in the formation of invariants such as primitive concept – names like “red” (from a predicate) as an invariant from different sweeps, number (from a collection of collections, of objects, that are in pairwise one – one correspondence), relation (from several different statements of a relationship that reduce to the same tuple set and label an edge in <V, E>), and a referent free KH (from a collection of KH's that are pairwise isomorphic) – see [VGS 2004]. Thus <V, E> can be taught to "recognise" certain invariants, in contrast with the ever present noise.

As input through I to subsets of primaries varies due to controlled or uncontrolled feedback through the environment from O, <V, E> will not only recognise previously established higher memory gauge value, more "used", invariant structures and edges, but it will continue to create new ones and change previously established ones. <V, E> can thus be taught, and it can learn, due to the existence of a dynamic and partially controllable feedback circuit in G/<V, E>/F.

The G/<V, E>/F system can establish invariant structures given appropriate trial-and-error experiences and examples. These can be interpreted using <V, E>, bearing in mind that the edges of a referent free KH can each be associated with a number of statements of relationship – see the definition of a KH in [VGS 2004].

A possibility: perhaps one side of the brain is more adept at linking clusters (vertical/deductive/constructive/inferential reasoning), while the other is better at finding links, such as isomorphisms, (lateral/analogical reasoning) between referent free invariant structures without concern for any particular interpretation of those structures via <V, E> - see [VGS 2004].

10. Interpretation of a KH

We only "notice" and interpret invariant, completed (sub-)KH's. If a sweep in G induces isolates, or effective edges that form loops or circuits, we must regard those as noise.

An interpretation of a KH is any one of the CRKS's from which that KH is abstracted – see [VGS 2004]. CRKS isomorphisms can exist between two or more of these interpretations, of the same KH, or between two or more interpretations of isomorphic (sub-) KH's. Every CRKS isomorphism is defined via abstraction to a KH, then KH isomorphism, and then interpretation – see [VGS 2004]. Of course any KH is isomorphic with itself.

We must emphasise that "activation flow", i.e. a sweep, occurs in G and is a "hardware reality". On the other hand, information flow occurs in the edges of <V, E>, and in particular, in the completed invariant KH subnets of <V, E>. While induced by activation flow, information flow is an entirely theoretical notion. Information flow becomes "interpreted flow" when we have any one of the interpretations of the relevant KH. Interpreted flow can mean a variety of things; for example knowledge flow in the interpretation CRKS or simply derivation and the progress of a limited access cascade, from the primaries, in that CRKS. Note that for every edge {u, v} of <V, E>, the members of its label are all unnamed vertices of G, so our KH's are all referent free.

In the case of the KH for an image, the interpretation CRKS would be very complex but idiosyncratic, and would be noticed and recalled only in relatively small sections each of which is both idiosyncratic and relatively simple, i.e. incomplete, with respect to the full detail of that section. The same would apply to noticing and recalling the image as a whole: much of the detail would be missing from such an overall experience of the image, and it would be idiosyncratic. An expert in the type of image involved would notice and recall more detail than a non-expert, and have a different and more detailed interpretation.

We have dealt briefly with trial-and-error, examples, invariance and insight. We have pointed out how (rare) insights can occur. The other basic facet of learning is declaration, and thus, here, CRKS interpretations of a KH. Teaching tries to establish an interpretation, that is regarded as “correct” according to some opinion, by controlling feedback. When a teacher says “pay attention”, s(he) is trying to direct feedback from O to I.

11. Concluding Comment

The “measure” and nourishment of curiosity, imagination and learning is speculation. This speculative model is intended to further stimulate discussion by the curious and imaginative. To set up our model we need

- To specify a loop free diagraph G = <U, A>.
- Find a vertex base B for G, introduce I, and let I be incident to each member of B.
- Find a vertex contra-base C for G, introduce O, and let O be incident from each member of C.
- Find the neurons, and hence the set of nuclei $V \subseteq U$, in G.
- Define, at each time, the (effective) edges of <V, E>.
- Let the teacher/programmer function as the feedback from O to I.

If it does nothing else, our model clarifies the approach to teaching and learning indicated in [VGS 2004] and Part 1 of [GVS 99]. Learning is modeled in the G/<V, E> part of the system, while teaching involves partial attempted control of the feedback from O to I. The G/<V, E>/F model provides some possible theoretical mechanisms for achieving learning and teaching, for example the building of invariant subnets of <V, E> with contrasting noise, limited access cascades and information flow, fast access cascades and insight and intuition, isomorphic subnets, and partial control of feedback from O to I.

Notice that the fundamental notion of a model hinges upon inductive abstraction, which is characterised by structural isomorphisms – see [VGS 2004] and Part 1 of [GVS 99].

Acknowledgements

Sue Fenton and Petro van Rooyen for typing the drafts. The many colleagues and friends who have influenced our work. Our families for constant support.

References

[VGS 2004] Van Rooyen Hendrik O., Geldenhuys Aletta E., Stetter Franz: Modelling knowledge systems using relation nets and hypernets. Technical Report TR-04-009, Dept. of Mathematics and Computer Science, Univ. of Mannheim, 2004.
http://madoc.bib.uni-mannheim.de/madoc/volltexte/2004/850/pdf/TR200409.pdf

[HNC 1965] Harary, F., Norman, R.Z., and Cartwright, D.,: Structural models: An introduction to the theory of directed graphs. John Wiley, 1965

[GVS 99] Geldenhuys, Aletta E., van Rooyen, Hendrik O., and Stetter, Franz: Knowledge Representation and Relation Nets. Kluwer Academic Publishers, 1999.