Logic in Pictures:

An Examination of Diagrammatic Representations, Graph Theory and Logic

by

Derik Hawley

A Thesis
presented to the University of Waterloo
in fulfillment of the thesis requirement for the degree of
Master of Arts
in
Philosophy

Waterloo, Ontario, Canada, 1994

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Derik Hawley

I further authorize the University of Waterloo to reproduce this thesis by means of photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Derik Hawley

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below and give address and date.

Abstract:

This thesis explores the various forms of reasoning that are associated with diagrams. It does this by a logical analysis of diagrammatic symbols. The thesis is divided into three sections dealing with different aspects of diagrammatic logic. They are: (1) The relevance of diagrammatic symbols and their role in logic, (2) Methods of formalizing diagrammatic symbols, such as subway maps and Peirce's Existential Graphs through the means of Graph theory, (3) The conception of inference in diagrammatic logic systems.

For Emil Post

Table Of Contents

Introduction:

This thesis is tripartite. Each part will be dealt with in its own section. The individual parts are:

(1) Diagrammatic reasoning is as much a part of logic as discursive reasoning.

(2) Diagrammatic Reasoning has the potential to be formalized by using graph theory. Graph theory can provide a method of defining formal procedures for making inferences with diagrammatic symbols.

(3) The representation of ideas in Diagrammatic symbols supports a slightly different conception of inference than does the representation of ideas in Discursive Symbols such as First Order Logic.

The first section addresses the issue of whether or not diagrams can form effective symbol systems. The theory of symbolism that I will be using is a modified form of that of espoused by Susanne K. Langer, in her book **Philosophy in a New Key.** In her theory a symbol is "not a proxy for the object, but *a vehicle for its conception*". [1] However, any theory of symbols must include a certain capacity of symbolic systems-- namely the capacity of symbols to serve as platform for calculation or formal inference. Symbols have two uses. They can serve as a vehicle for conception. And they can serve as a platform for calculation. A symbolic system is a prerequisite for calculation. Calculation is nothing more than the application of a set of formal procedures upon a symbolic system. Langer deals mostly with the capacity of symbols to convey conception[2]. This is largely because of her interest in Art and philosophy of Mind. In such a study it is the communicative aspects of symbols which are important. However I will be dealing with a second capability of

[1] **Philosophy in a New Key, pp. 60-61.**
[2] By conveying a conception I mean that the symbol in question induces a mental representation of the entity (real or imagined) depicted or described by the symbol.

symbols, specifrcally *the capacity of diagrammatic symbolisms along with rules of well formation and various formal transformations to serve as a platform for formal inference.* A formal inference is the production of a new symbol by the application of an algorithmic procedure on previously given ones. The calculation of a solution to an equation is a example of a formal inference."

Symbols serve two functions in our collective day to day life, namely communication (facilitating the exchange of conceptions) and calculation (allowing us to do inference on the basis of the formal properties of symbols). The first and most common function is that of communication of conceptions. Symbols, such as those which are found in a textbook, subway map or timetable etc., serve to communicate conceptions of their intended objects. Symbols are also used for calculation. A calculation involves a set of symbols together with a set of operations. These operations involve only the formal properties of the symbols, and do not involve their conveyed or intended conception. The argument of the first section is that diagrammatic symbols have the same capacity to communicate information as do discursive ones and they have the same capacity to facilitate calculation as do discursive symbols.

The second section of this thesis will present graph theory as a method of formalizing diagrammatic symbols. In order to make this point clear I shall provide a brief introduction to graph theory, and show how to go about translating conventional formal notations into graph theory. The resulting system is called GL (Graph Logic). It is based upon a system of Attributed Hypergraphs developed by the Pattern Analysis and Machine Intelligence laboratory, at the University of Waterloo, for the purpose of developing a system of representation for assorted tasks in Robotics and three dimensional scene analysis. However, it is extended by incorporating various methods used by Peirce in his Existential graphs.

In the third section I will discuss various forms of inference that can be performed using a diagrammatic symbol.

---

[3]This is not to say that solving equations is a purely mechanical task. But rather that any insight that is involved in solving an equation is a matter of choosing the method of proceding, not in its application.

An examination of diagrammatic logic systems results in a different conception of inference then that which occurs in the examination of a classical logic system. Classical systems can be divided into two types. Those which use a a finite set of inference operations are often called Principia-like systems after **Principia Mathematica**. Modus Ponens, for example, can serve as the sole inference operation for a deductively complete system, depending on the axioms chosen[4]. The other systems are natural deduction systems. A natural deduction system has a countable set of inference rules. In diagrammatic systems both Principia like and Natural Deduction inference operations can be used. However, most string based systems use at least two premisses for each conclusion. In diagrammatic systems there is only one premiss, which is the initial diagram itself. As a result of this, the conventional conception of inference can not be applied without alteration.

[4]For example the system P of Alonzo Church, in **Introduction to Mathematical Logic.**

1.Diagrammatic Reasoning

I talk in pictures not in words.
Peter Gabriel.

1.0 The thesis of this section is as follows:

> **Diagrammatic reasoning is as much a part of logic as discursive reasoning.**

Several terms must be defined, in order to make this intelligible. By *"diagrammatic"* *I* mean that part of reasoning which is connected with pictures, models and icons[5]. The symbolism conveys the conception by means of diagrams rather than through words or a string based symbolism such as first order logic. By *"discursive"* I mean that part of reasoning which is connected with strings of symbols. They may be part of a formal language (such as First Order Logic) or a rigourized informal one (such as the English language when used in critical thinking textbooks or law courts). By *"logic" I* mean logic qua practical study[6]. I mean *logic* conceived of as a collection of techniques, and knowledge of their limitations. It is those methods which pertain to the practical task of representing information and making inferences solely on the basis of those symbols. This is *logic as the art of symbolizing and inferring on the basis of the symbols.* Logic as an art is not logic interpreted as the science of necessary features of the world. There are arguments that suggest that such a science is impossible. Be that as is it may, logic as an art is an equally important part of life.

The classic logical conceptions of completeness (inferential and referential), consistency, and compactness7 are still applicable to logic as an art. But there

[5]This is a different notion of diagrammatic then C.S. Peirce had. For Peirce regarded even algebraic symbols as being diagrammatic. The easiest way to define my conception of diagrammatic symbolism is to say that the components of that symbol are interelated spatially in more than one direction.
[6]Logic as the "art of thinking" as opposed to the science of thinking. Logic as a *Techne* in Aristotlean terms.
'Compactness is a property of a logical system. A system is compact if and only if any theorem can be derived by a finite number of inference steps.

are other issues that are untouched by traditional treatments. These revolve around tractability. A problem is tractable iff it can be solved in a time that makes the inference worthwhile. The theoretic issue, whether or not a certain inference can be made at all is supplemented with the question of whether or not such an inference can be made in a reasonable length of time. So too the issue of representational completeness is supplemented by the practical issue of ease of understanding, and the degree to which the representation chosen. Even if it is possible to represent a conception in a particular symbolism, there is no point in doing so if that conception remains unconveyed by the symbols chosen.

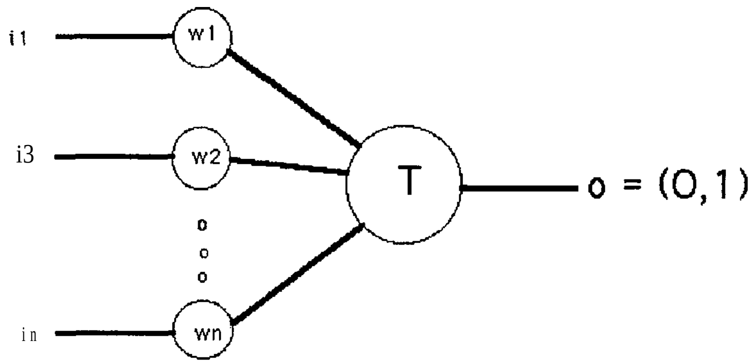
1.1 The Argument

The argument in this section will revolve around a specific example. However the argument is a general argument and not dependent upon the example chosen.

The example that I chose to use is a representation of a single node in a neural network in discursive and diagrammatic form. I will show how the discursive symbol is identical to the diagrammatic symbol by the criteria discussed previously. Two symbols are equivalent if and only if they can convey the same information and they make possible the same sorts of calculations. As the functions of symbols are communication and calculation, these functions form the criteria for equivalence.

I wish to distinguish two important aspects of any conception, namely its referential and operational aspects. We must show that two conceptions are identical in both operational and referential ways, if we are to claim that the conceptions conveyed by two symbols are identical. The referential similarity of conveyed conceptions is shown by the ability of two symbols to convey the same **indexical** reference. That a diagrammatic system and discursive system can convey the same referential components of a concept is easily demonstrated by the tacit agreement we have that a subway map refers to the same subway system as a verbal description of its stations and routes. This agreement is one that is assumed by almost everyone[8]. The operational aspects are more difficult to show. A diagram must be capable of conveying the operational components of a conception, that is a diagram must convey a conception in a manner such that the concept can be used in practice. The example of a subway map and verbal descriptions illustrates this point but not with sufficient clarity. What is needed is something more formal to illustrate that the capabilities of a diagram to convey the operational aspects of a conception applies equally well to formal as well as informal symbols.

The situation is rather like looking for a Rossetta stone. On the Rossetta stone there are three different symbolic forms. One is Greek, and the other two are forms of hieroglyph@. The hieroglyphs were comprehensible because the three symbolisms conveyed the same conceptions. It is surprising where one can find Rossetta stones. The one I wish to discuss was found in a pattern recognition textbook. The book gives two descriptions of the Linear Weighted Threshold Neuron (it actually gives several). The first is in the form of a diagram.



(Diagram)

[8]It is really hard to imagine someone who would not assent to the referential identity of such things. And even harder to imagine them surviving in the world.
[9] The ancient Egyptians had two Hieroglyphic scripts.

The convention for interpreting this diagram is that the inputs (referred to by the "i"'s) are multiplied by their weights (referred to by the "w"'s). The sum of these values is then compared with the threshold and an output of 1 will occur if that sum is at least as great as the value T, otherwise the output will be zero.

The book then provides an algebraic description:

$$\text{if } \sum_{k=1}^{n} ikwk < T \qquad \text{then } o = 0$$

$$\text{if } \sum_{k=1}^{n} ikwk >= T \qquad \text{then } o = 1$$

(Characteristics) [10]

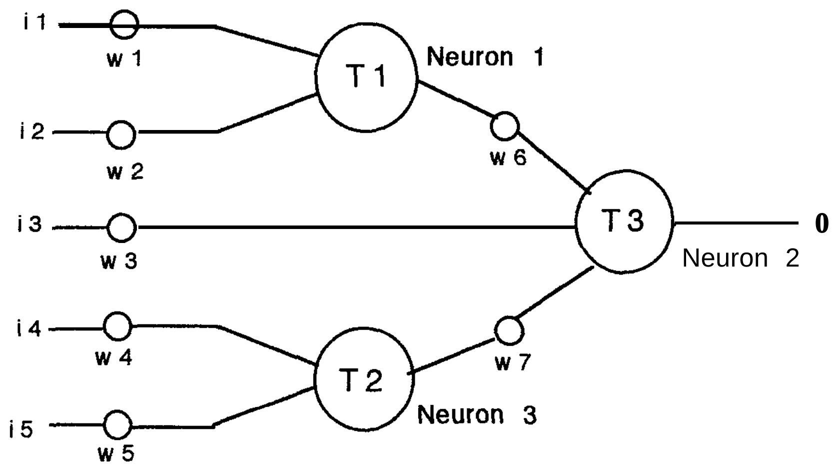
The two notations, the “diagram” and the “characteristics” convey the same conception, when one understands the conventions for their interpretation, Although the algebraic symbol is easier for the average person to interpret, it must be remembered that the algebraic symbol is also interpreted through a set of conventions. Someone unfamiliar with Sigma notation may interpret it as having something to do with a sideways "M". The algebraic expression is one which applies to a Linear Weighted Threshold Neuron, but it also can be used to model multi-criteria parametric decisions. The use of diagram notation allows for the representation of layered networks of Linear Weighted Threshold Neurons, which in the equation form are very complex, and often are unable to convey the conception to a reader. The algebraic notations allow for an understanding of the functioning of a single Linear Weighted Threshold

[10] This is taken from Robert Schalkoff’s **Pattern Recognition: Statistical, Structural and Neural Approaches,** p. 215. I have made two alterations. For clarity I have used "i" for inputs. I have also corrected a typographical mistake in the characteristics.

Neuron. Diagrams are often used to convey more complicated structures, which can not be efficiently symbolized using a discursive notation.

Having the discursive and diagrammatic symbols together can form a Rossetta stone, for the description of the functioning of the perceptron net through the equations allows the reader to gain an understanding of the conventions for interpreting the diagram notation. The real Rossetta stone allowed for the understanding necessary for the interpretation of the two Hieroglyphic scripts. Symbols convey conceptions to individuals only when those individuals have an understanding of the conventions necessary for their interpretation. The point I wish to make is these two symbols differ only in mode of presentation, and not in the conveyed conception. In order to demonstrate this I must show that they convey the same information and both allow for calculation of the output function. In fact one calculates using both symbols in same way, pressing the same buttons on the calculator, performing the same arithmetic calculations, or constructing the same computer program. In either case whether someone is capable of operationalizing a conception is a matter of their understanding of the conventions for interpreting the symbols. Both symbols describe the functioning of the Linear Weighted Threshold Neuron. The neuron will “fire”, i.e. send a signal "1" as output, if and only if the weighted sum of the inputs is equal to, or greater than, the threshold. The two symbols have the same operational meaning to someone who can understand the conventions of the two notations. They also, have the same reference as they both describe the same actual or possible node in a neural network. The two symbolic representations both contain a description about the functioning of the neuron to serve as a vehicle for its conception, to one familiar with its conventions.

The complexity of a discursive (whether verbal or algebraic) description grows with the complexity of the complex being represented, but is lass capable of conveying conceptions of complicated structures. It is very difficult to have a discursive expression of the functioning of a neural network. Diagrams are able to convey a conception of a complicated neural network. For example take the following, diagram:



A discursive description of this network would be something like this:

> The Output will be 1, iff the sum of the inputs into the 3rd neuron is at least the threshold T3. The Inputs into the third neuron are i3*w3 and the output of Neuron 1* w6 and the output of nueron2*w7. Neuron 1 will fire if and only if the sum of the inputs is at least the threshold T1. The inputs to Neuron 1 are i 1*i2 and i2 *w2. Neuron 2 will fire if and only if the sum of inputs is at least the threshold T2. The inputs to Neuron 1 are i4*w4 and i5*w5.

Which is not something that is easily interpreted.

The generality of this argument comes from the fact that there are many cases where diagrams are used to convey conceptions - for example, circuit diagrams, architectural plans, subway maps, and model instructions. Each of these diagrams serve to convey their intended conceptions. They do this not by providing a discursive description representation but by providing a diagram. In each of these cases the diagram provides a method of conveying the conceptions. And in each case the symbol conveys both the representational and operational aspects of the conception.

1.2 Conclusions

Diagrammatic symbol systems are as much a province of logic as are discursive symbol systems. The choice between methods of representation is a matter of practical consideration which depends upon the purpose to which the symbolism is intended to be applied. This follows because the operational and referential aspects of conceptions conveyed by diagrammatic symbols and discursive ones are, at times, identical. Both sorts of systems can convey the referential aspects of the conception and the operational aspects. The two sorts of symbols can serve the same function in that they can convey information about the world and serve as a platform for calculation. Although there is a distinction in the presentational mode they both can satisfy requirements of an effective symbolization strategy.

At the level of logic as a theoretical study the choice of a particular strategy is not important. However, at the level of logic as an art the choice of symbolization strategy, the choice of a symbolization strategy is very important. This is because the choice of symbols used for expression will affect the ease with which they can convey conceptions and the degree to which they facilitate the inference of the desired conclusion. The choice of a particular strategy is determined by the task at hand. Given the many possible formal systems that can be used to convey information and serve as a platform for inference the choice of one strategy over another ought to be defined by convenience rather than by a fixed a priori account.

The dominance of diagrammatic representations of neural networks, as opposed to a series of mathematical symbols, is not just a matter of shared personal taste. The reasons that such a symbol is adopted in almost any book or paper on neural networks is that such a symbol of the mechanism facilitates conception. There are several other notations for representing Linear Weighted Threshold Neurons, each with its own conventions for interpretation. In situations where a large complex of neurons is being discussed the algebraic notation ceases to be an effective means of conveying conceptions; the equations become to complicated.

There are, of course, disciplines where discursive symbols are more **advantageous. Algebra and Calculus are far easier to do in a string notation. There needs no justification** of a method of symbolization beyond its effectiveness at performing a task. Once the requirements of completeness (to the degree required[11]) are established, the task ought to determine the choice of symbolism.

[11]For a great many tasks one needs not the capacity to convey quantified propositions. For example, a list of telephone numbers.

1.4 Remark:

A diagrammatic symbol system is a formal symbol system for the same reason that a particular discursive symbol system is formal. For example, First order logic is a formal discursive system. English is an informal discursive system. So too with diagrammatic logics there are formal systems and informal ones. Peirce's existential graphs form a formal diagrammatic system. Peter Bruegel's **Hunters in the Snow,** a print of which hangs on my wall, is part of an informal diagrammatic system of symbols. The same characterizing features which make a symbol system formal hold for both discursive and diagrammatic systems. They are:

(1) The rules of well formation are precisely defined.

(2) Inference is limited to a set of pre-defined algorithmic procedures.

Any formal system is capable of being mechanized into a computer program which implements the inferences automatically. The same has yet to be the case for informal systems.

The remainder of this thesis deals only with formal diagrammatic systems, and the sorts of inference that can be done on the basis of the formal qualities of a diagrammatic symbolism. I will take this opportunity to describe the notion of formal qualities. A formal quality of a symbolism is contained in the symbol itself and has nothing what so ever to do with the concept conveyed by the symbol. A formal inference then, is an inference that is done according to some rule, that is describable on terms of just the symbols themselves.

2. Graph Theoretic Logic

> **"Every picture is at the same time a logical one. (On the other hand, not every picture is, for example, a spatial one.)"**
>
> *L. Wittgenstein.*

The Thesis of this section is as follows:

> **Diagrammatic reasoning has the potential to be formalized by using graph theory. Graph theory can provide a method of defining formal procedures for making inferences with diagrammatic symbols.**

That a system of logic can be constructed from a system of diagrams has been by Charles Sanders Peirce, for his system was later shown to be a complete system of logic (Roberts 1973). This section of the thesis will deal primarily with the potential of graph theory to represent logical symbols of equal expressive power as first order logic such as the Existential Graphs.

Just as "discursive" reasoning has been given formal analysis through the methods of formal logic and formal language theory, graph theory provides a framework by means of which "diagrammatic" reasoning can be given such a formal treatment. Just as formal logic replaces words with abstract symbols, Graph theory replaces icons and arrows in a diagram with abstract entities called vertices and arcs. This allows the formal properties of the symbolism to be examined without the intervention of any aspects of the conception which is conveyed by the symbol affecting the inference. Formal logic treats inference as a set of operations that deal only with the operations on the symbols themselves, and does not examine the inference that involves associations based upon the conceptions conveyed by the symbols. There are certain advantages to such a treatment. Formal logic provides a framework where the examination of the validity of an argument can be examined without the intervening effects of connotations and emotions. The formal examination results in a knowledge of whether or not the premises of an argument are sufficient evidence for the conclusion. This is a useful thing for it provides us with a method of examining our own opinions and those of others.

The study of logic has, with one exception, dealt only with "discursive" symbolisms. It has examined the ways in which the discursive symbolisms can be formalized using different systems. Historically, one can divide the set of logic systems into three categories. First there is the Aristotelian Logic system, secondly there are propositional systems and finally there are the modern quantified, relational logics (which include first and second order logic). Each of these developments attempts to improve the range of arguments that can be given formal treatment. However, there has never been an attempt to deal with "diagrammatic symbols" and provide a framework for the examination of the sorts of inferences that involve the formal aspects of diagrammatic symbols. By using graph theory we can examine diagrammatic symbolisms in a manner such that they can undergo the same formal scrutiny that discursive reasoning can be given in formal logic.

In this chapter I will be giving an introduction to graph theory and a few examples of the manner in which graph theory can be applied to diagrammatic reasoning. I will then sketch a method by which Peirce's existential graphs can be translated into graph theory.

2.1 What is Graph Theory?

Graph Theory is a popular and widely applied mathematical theory. Some of its major applications include:

(1) Circuit Design
(2) Road Network Design
(3) Room, personnel and CPU time scheduling.

As graph theory is a well established mathematical topic, it is rich with theorems. These theorems, under the correct transformation, become theorems of logic.

2.2 A Brief History and Outline of Graph theory

Graph theory is a mathematical theory with a rich and interesting history[12]. Early developments began with Euler's treatment of the Königsberg bridge problem. It also appears in a puzzle presented by Sir William Rowan Hamilton which is now known as the Hamiltonian Circuit problem. With the introduction of the Four colour map problem and its erroneous solution by Alfred Bray Kempe, graph theory became an object of mathematical study. Kempe's article "Memoirs of Mathematical Form" greatly influenced the American philosopher Charles Sanders Peirce in the development of his existential graphs (see Roberts 1971, sec 2.3). Peirce altered Kempe's notation and used branching lines (which are capable of expansion), rather than vertices, to represent individuals. Throughout the twentieth century graph theory has been used as a formal tool for describing molecules, circuit diagrams, flow diagrams, road topology and Game Theory .

Two of the best text books in the field are Springer Verlag's **Graph Theory** (Bollabas 1979) and North-Holland's **Graph Theory with Applications** (Bondy, Murty 1976) [13].

2.3 Peirce's Existential Graphs and Graph Theory.

Charles Sanders Peirce developed his diagrammatic system of logic (i.e. the Existential Graphs) under the influence Alfred Bray Kempe's "Memoirs of Mathematical Form". However, Peirce chose to alter the representation. He did this in order to exploit the topological features of the plane (i.e. a piece of paper or a chalkboard). In order to do this he made objects correspond to entities which can expand and contract on the printed page. This method has its advantages but also its disadvantages when one attempts to give it a computer implementation. For computer memory is discrete, and data objects cannot

---

[12] See N.L.Biggs, E.K. Lloyd, R.J. Wilson., **Graph Theory 1736-1936,** for a history of graph theory.
[13] Another useful text by L.R. Foulds, **Graph Theory Applications,** part of Springer-Verlag's ***Universitext*** series, suffers from several terminological errors causing a confusion between the concepts of NP-hard and NP-Complete when applied to graph algorithms. This is more than a trivial error, for it results in the fact that addition is in P implying the equivalence of P and NP.

undergo the same expansion as Peirce’s spots. I have chosen to develop a graph theoretic model for diagrammatic systems. This system is bases jointly on Peirce’s Existential Graphs and the Attributed Hypergraph Representation system used in Robotics.

The reason for the reformulation was hinted at above. Graph theory is a rich an well established mathematical topic, and has been shown to be a useful technique for solving many practical problems. Moreover, from a point of view of implementation there are various graph algorithm packages available which support and allow for the implementation of graph algorithms. This is a great advantage when one want to operationalize a mode of representation. As logic as art involves the practical aspects of symbolization, one has to build on what has already been accomplished.

There are other advantages to the development of a formal representation based on graph theory. Graph theory has been shown to be an effective symbol strategy in Artificial Intelligence-- especially in computer vision. If the study of logic is to remain of service to humanity an understanding of the developments in neighboring disciplines is a requirement. Otherwise logic will fall the way of Latin as an academic discipline. The humanist movement during the Renaissance with its insistence that Latin remain an unchanging and static language, is largely responsible for Latin’s death.[14] First order logic will suffer the same fate, for its inflexibility is legislated by canonical injunction. Logic as a study will perish with it unless it begins to examine the features and potentials of alternative formalisms.

I shall now proceed to provide a brief introduction to Graph Theory, and formulate a graph theoretic language (CL) within the bounds of Graph Theory.

2.4 The basic concepts of Graph Theory.

In this section of the I shall outline the basic concepts of graph theory. I shall do this semi-formally in order to present the concepts without a great deal of set theoretic baggage.

---

[14] For example there are no words for “car”, “keyboard” etc. in Latin.

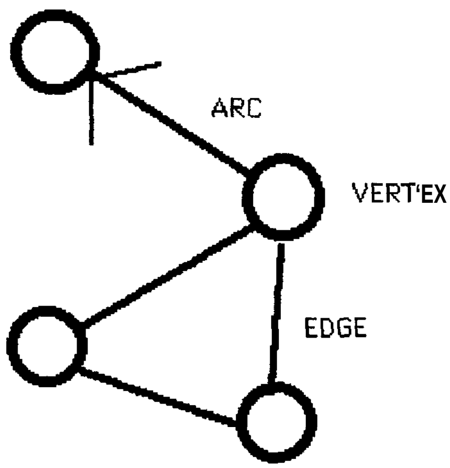
2.4.1 **The Primary Graph Elements**

The Primary elements of a graph are two sets of objects, vertices and arcs. All other graph elements are complexes of these primitive elements.

A vertex is a graph representation of an entity.

An arc (or directed edge) is a representation of a non-symmetric relation between entities represented in a graph. An Edge is representation of a symmetric relation between entities represented in a graph.

The following diagram should clarify the notions of edge and arc.



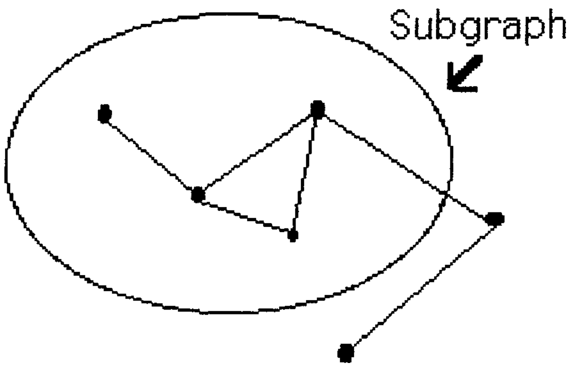
ELEMENTARY GRAPH COMPONENTS

The conventions of graph theory uses arrows to show arcs, and lines to show edges. This is merely a convention for representing a graph.

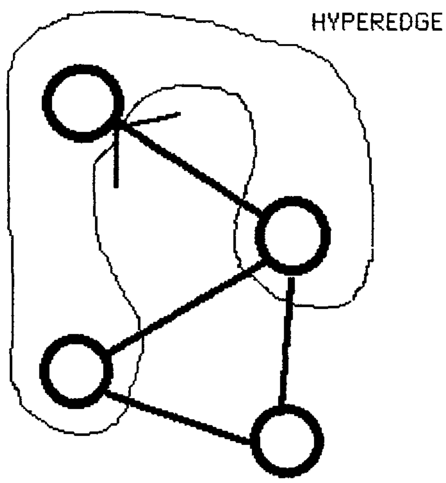
2.4.2 **Secondary Elements**

From the primary elements of a graph there can be constructed secondary elements of a graph. The secondary elements are complexes of the primary elements.

A **Subgraph** is a graph theoretic representation of a set entities and the way in which they are related. Subgraphs refer to complexes of entities rather than to single entities, which are referred to by vertices.



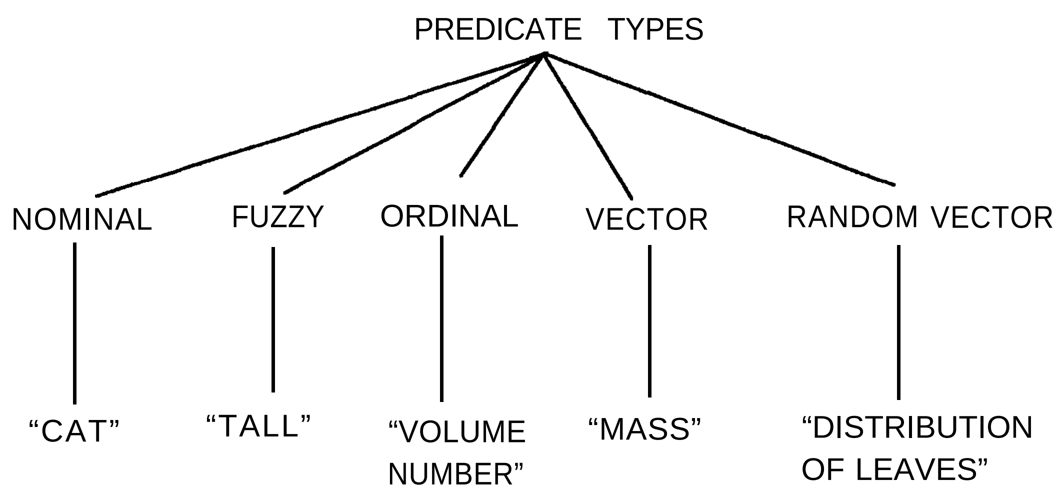
A hyperedge is a special sort of subgraph. It is one that contains only vertices as members.



2.4.3 Element Labeling and Predicates

A graph which has not been labeled (a predicate attached to the various elements which have some semantic import), is an abstract structure like the natural numbers. It describes abstract relations between abstract objects. The labeling of an element is an association between a Predicate (something which can be said of an entity) and a graph element. Predicates come in several varieties depending on what sort of property is being attributed to the graph element. I am expanding the range of predication beyond the limits of First Order Logic, in order to demonstrate the capabilities of graph based formalisms.

The following diagram gives examples of sorts of predicates that can be used:



As predicate is anything that can be said of a graph element, predicates can come in a several classes that are distinguished by the means by which they can be represented in a knowledge or data base.

A predicate can be applied to any graph element, be it an arc, edge, vertex or subgraph.

I shall now proceed to discuss briefly the various sorts of predicates that can be used in a graph theoretic representation. This list goes beyond the sorts of predicates that are commonly studied in logic. This is because they involve the predication of fuzzy, cardinal, etc. properties to objects.

2.4.3.1 Nominal Predicates

A nominal predicate is any property or relation that can be represented by a name. The term “nominal” merely refers to the fact that these sorts of predications can be described completely by means of an object or class name.

examples of nominal predicates are “on”, “red” etc.

A nominal predicate is an ordered pair <e,p> where e is a graph element and p is a property name.

Nominal predicates come in two varieties general and specific. A general is something that can attributed to more than one element. A specific can predicated to only one element in the graph.

2.4.3.2 Fuzzy Predicates

A fuzzy predicate is a special case of a nominal predicate. It is a predicate that ascribes a property to an element which that element can have more or less. An example of a fuzzy predicate is a “tall” or “old”.

A Fuzzy predicate is an order triple <e, p, d> where e is a graph element, p is a property and d is a value on the interval [0,1] indicating the degree to which the element has that property.

Example: <Wilt Chamberlain, Tall, 0.98>
<Bilbo Baggins, Tall, 0.03>

In practice the degree is represented by a floating point variable.

2.4.3.3 Ordinal Predicates

Ordinal Predicates represent quantities that are discrete, and at least partially ordered.

An Ordinal Predicate is an ordered triple <e,p,n> where e is a graph element p is a property name and n is a Natural Number.

An example of an Ordinal Predicate:

<Encylcopedia_Britanica, volume, 4>

2.4.3.4 **Vector Predicates**

Vector Predicates are a means of representing a vector quantity. A vector quantity requires a definition of the vector space and of the vector itself.

A vector predicate is a ordered n-tuple <e,s,a,b,c....> where e is a graph element, s is a designator of a vector space and a,b,c.... is the description of the vector in that vector space. The values for a,b,c... are best represented by floating point variables.

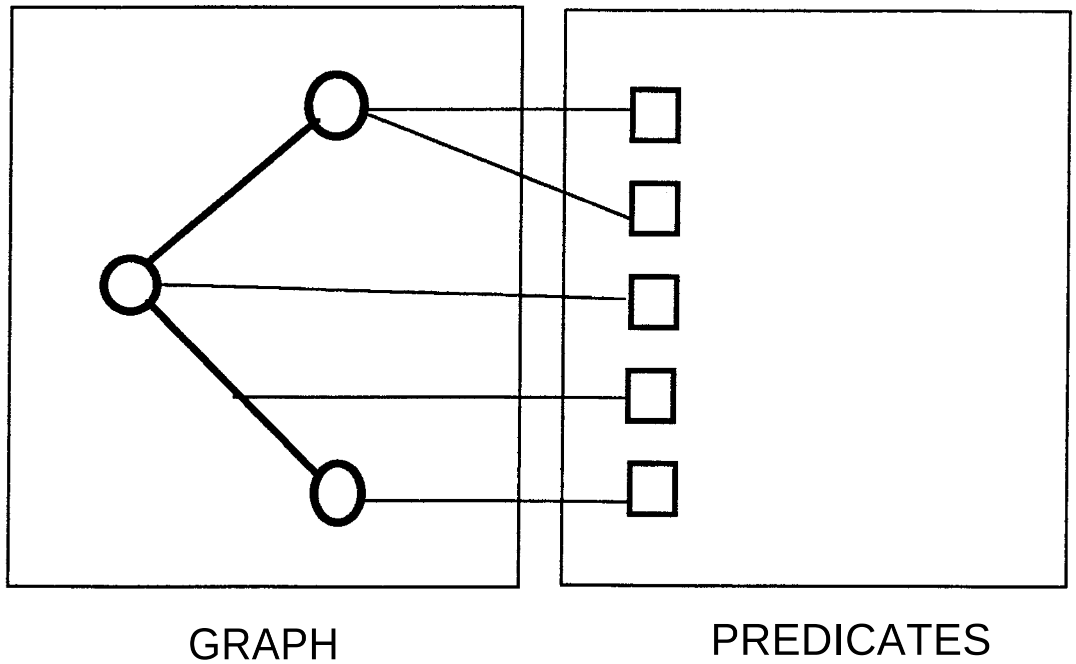
Examples of vector quantities are mass, area, shapes of rectangles.

<object_a, mass(kg), 2.4>
<surface_b, area(meters), 2.5
<rectangle_c, shape(cm*cm), 20, 30>[15]

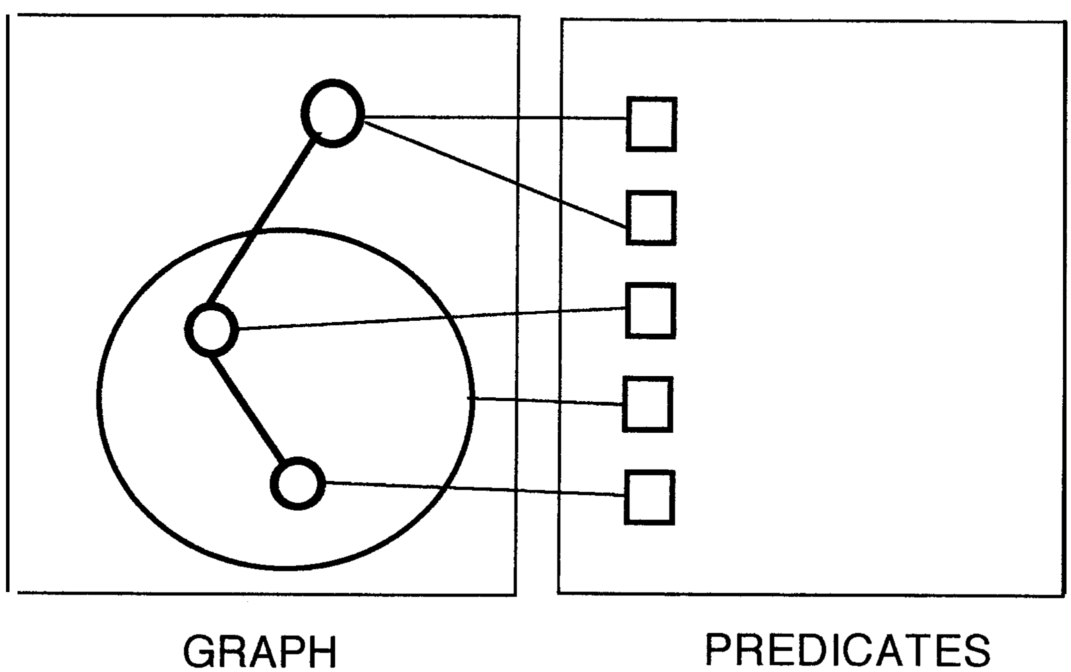
2.4.3.5 PREDICATION

When we label a graph element we form an association between a predicate and the graph element:

[15] The rectangle "c" is 20 cm by 30 cm.



A Predicate can be applied to any graph element including Subgraphs:



2.5 GL the vivid component

In this section I will outline the construction of a vivid GL symbol. A vivid GL representation allows only the assertion of positive facts and their conjunction. It does not allow the expression of negative facts or the disjunction of facts. In section 2.6 I shall outline the implementation of quantification and other logical operators into the symbolism.

2.5.1 Vivid and Iconic Symbols

The first conception I wish to deal with is the idea of vivid, or iconic symbols. Vivid representations, defined by Hector Levesque as having the following properties:

> **(1)** For every object in the world that is of interest there is a corresponding data object instantiated in the knowledge base.
> (2) For every relation of interest in the world there corresponds a relation instantiated in the knowledge base.

Vivid is the computational equivalent of Iconic. Iconicity is defined, by Peirce, Langer, and others, as:

> A representation is *Iconic* if the components of the symbol stand in the same logical relationship as the components of the object represented.

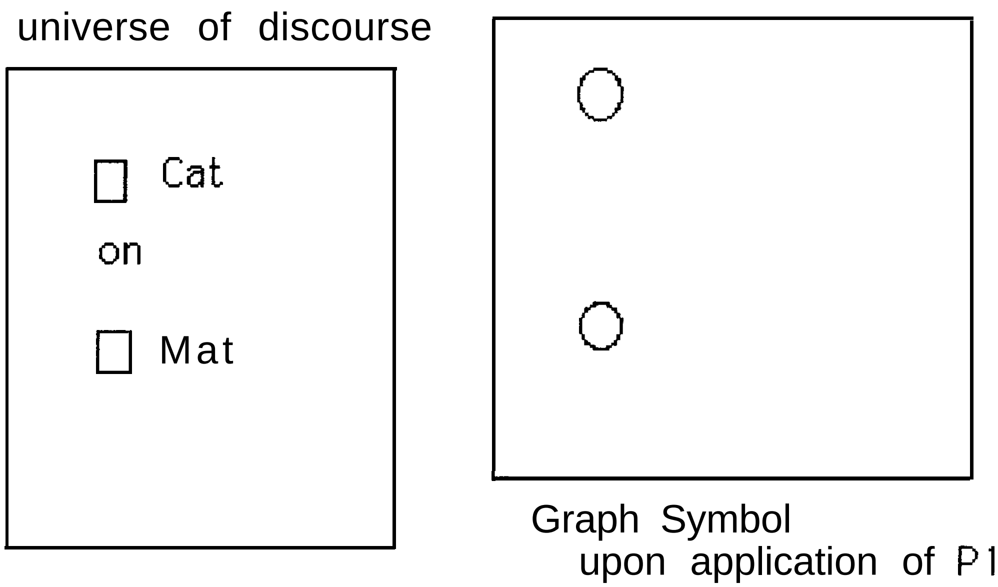
An iconic symbol is not necessarily a direct or subjective image of the object, for the relationships may be defined temporally in one domain and spatially in another, such as in a project flow chart. It is, nonetheless, an image in the mathematical sense since it is isomorphic to what is represented.

Although diagrammatic and graph theoretical representations are an image of what they represent, they are an image in the mathematical sense rather than the psychological sense. Nothing of what I shall say in this thesis pertains directly to the study of mental imagery. The graph symbols are not images in the conventional sense, for a psychological image is more analogous to a photograph and is arguably a continuous entity rather than a discrete one.

2.5.2 Construction of GL representations

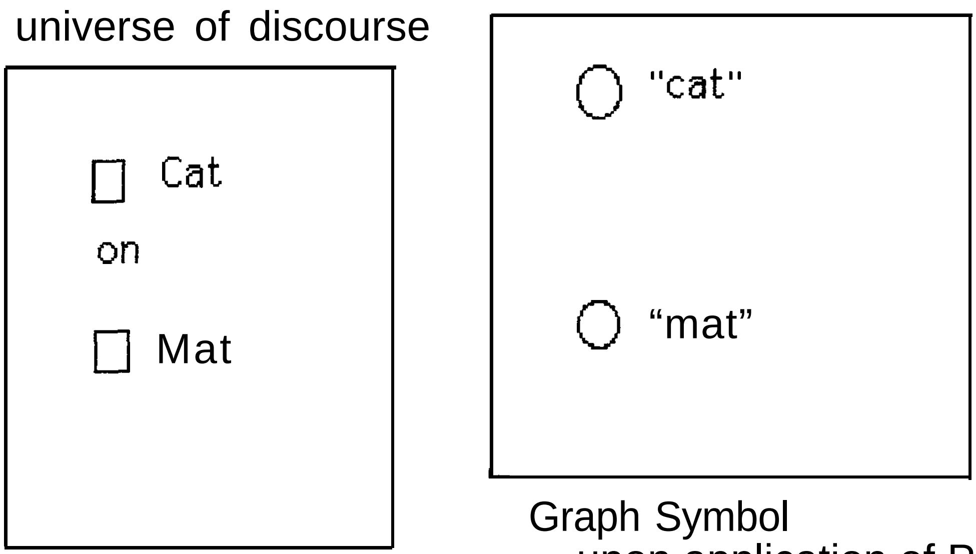
The following principles guide the construction of a Vivid GL representation. I shall follow each principle with an application.

(Pl) For every object in the universe of discourse instantiate a vertex.[16]



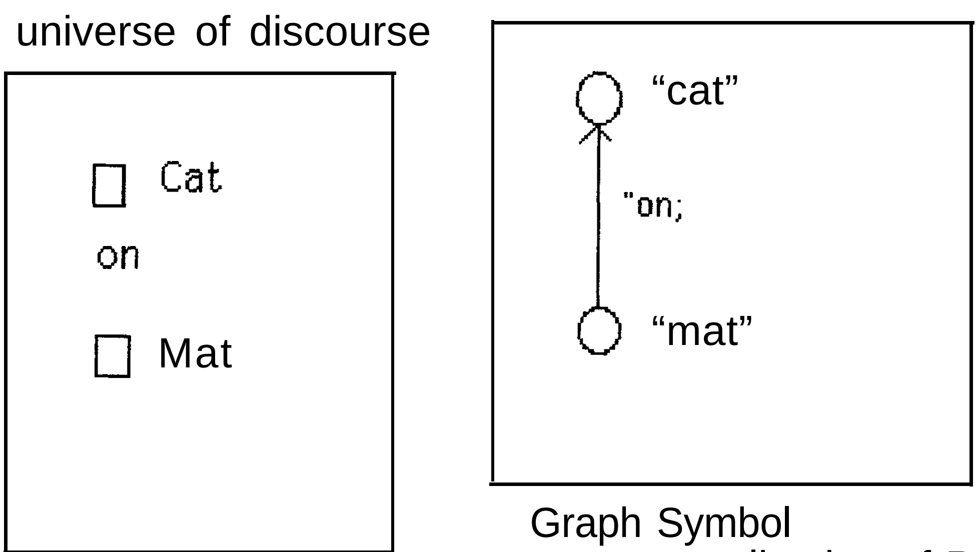
(P2) For every monadic predicate on object x instantiate an attribute to the vertex with a symbol corresponding to the predicate.

[16]The resulting GL graph representation will have an order (the cardinality of the vertex set) that is equal to the cardinality of the universe of discourse. I **bring** it up here because it is an obvious consequence of P1.



Graph Symbol
upon application of P2

(P3) For every dyadic relation R between x and y instantiate an arc with a tail at vertex x' and a head at vertex y'. Attribute the edge with the appropriate symbol. If the relation is symmetrical (such as "is-beside") then instantiate a second arc, in the opposite direction and attribute accordingly.

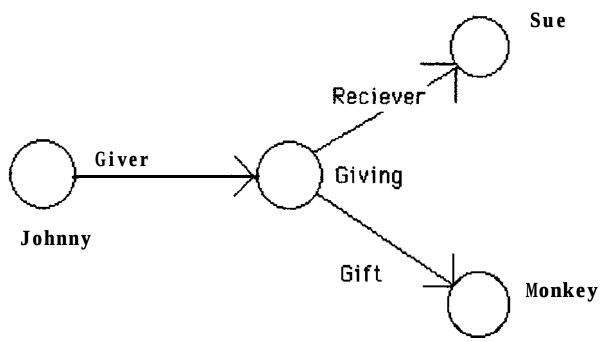


Graph Symbol
upon application of P 3

(P4) Triadic and higher relations can be reduced to dyadic relations by the instantiation of additional entities.

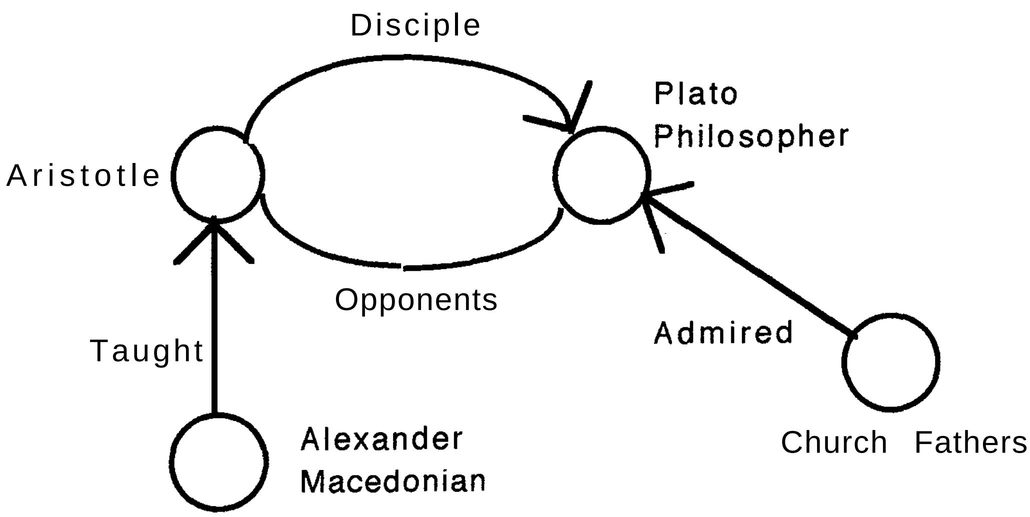
Peirce maintains that all relations can be reduced to triads. However, dyadic entities such as arcs or edges work just as well. For a triad can be made by adding a new entity and instantiating three arcs.

For example, “Johnny gives Sue a monkey”, can be represented as:



Whether one has a preference for dyads (represented by edges and arcs) or triads is ultimately a matter of personal choice, or practical consideration. Peirce’s fascination with Triads, and the number three, is not strongly supported by the logic of relatives. One can argue that the “giving” is a triadic sign. However, that triad can become a Quadrad, if one makes an arc which represents “occurred upon” and a vertex represents a date. Peirce’s theory of signs is developed around triads. But reducing signs to triads produces the same notational complications as does reduction to dyads. Although I have briefly left the field of logic for the theory of signs (semiotics), the semiotic point is important. If a graph theoretic conception of logic is to allow for a rich enough representation for a system of signs then it must be capable of representing an n-adic relation for any **n.** I do not claim the point that this is necessarily the best way to deal with the representation of triadic relations. But neither is the reduction of n-ads to triads the ideal way of coming to understand a quadratic or higher degree relation.

To show an example of the result of the application of these principles we get the following vivid GL symbol of the universe of discourse in the following sentence: "Aristotle who taught a Macedonian named Alexander, and was a disciple and an opponent of Plato who is a philosopher and greatly admired by the Church Fathers" The application of principles defined above would allow a person to construct the following symbol:



2.5.3 Vivid Components of the Existential Graphs

Any Existential Graph which does not contain a cut is a vivid symbol. I will not formally prove this but give an example which should clarify my point. The above GL symbol has an equivalent Existential Graph. The vivid components of a symbol correspond to entities that are postulated to have a reality of being actually present. The non-vivid components of the GL symbol, which will be defined briefly in the next section, do not correspond to entities that are represented in the same way. They can become represented vividly only by the application of an inference procedure. They are conceivable, but do not correspond to any actually present conception.

There is an alternate way of viewing inference which I shall be discussing in the next section. It involves the use of a stricter separation of the vivid and non-vivid components of a graph. The non-vivid components are a removed from the graph symbol and made into formal rules of inference. In Peircean terms the *habits* are programmed into the inference mechanism of the expert system and treated as a different category than are the vivid components.

2.5.4 Negation and Identity

The four principles described above do not allow for the representation of negation and identity. The addition of the following principles of representation allow for the representation of negated and quantified statements. Their use will be clarified in the next section, and terms defined.

(P5) If two vertices are asserted to refer to identical entities instantiate an equivalence arc between them.

(P5) If something is denied then instantiate the graph elements and label them in accordance with the first four principles, and embed them in a layer subgraph.

(P6) Principle (P5) can be applied repeatedly.

These three principles allow GL to utilize the same features and methods for providing complete system of logic as the Existential Graphs. The layers serve the same role as the cuts and the **equivelence** arcs allow for idenity to be asserted on more than one layer. Thus allowing graph theoretic representations to implement the ability of the lines of identity to pass through the cuts.

2.6 **The Formal Specification of GL.**

In this section I will proceed to define formally the rules of well formation for GL graphs.

A GL representation has two important components, namely the underlying directed graph which contains the structural information. And a set of labels for the graph. A directed graph is an ordered triple:

G = **<V,A,f1>**

where

V is a set of vertices.

A is a set of arcs

f1 is a mapping

***f1*(VxV) -> A**

which keep track of which vertex goes with which arc.

A GL representation is an ordered **4-tuple**

GL = <G,S,L,*f1*>

G is a directed graph.

S is a set of Subgraphs of G

L is a set of labels (or predicates)

f2 is a mapping

***f2*(L) -> {V U A U S}**

such **that** *f2*(l)->**x** iff the element x is "labeled" with 1 (this function is a way of clearly defining the notion of labeling).

There are two special sorts elements of the set L. There is a set of "layers" and that of an "equivalence arc". Let the equivalence arcs stand for the identity relation. It asserts not a relation between entities but rather it asserts that the two vertices represent the same object.

Let the layer predicate be an order pair **(layer,d)** where d is a ordinal number representing the depth of a layer.

The layer labels can only be a subgraph with the following properties:

(1) If x and y are layers then either x is a proper superset of y or x is a subset of y or x intersect y is a null set.

(3) If x, is a layer, and d is the depth of the layer x, then there exist d layers that are supersets of x within the graph symbol.

The layers will then form a tree structure. This tree structure is isomorphic to the allowable arrangements of cuts on Peirce's existential graphs.

The ability of the "lines of identity" to cross over cuts is important in the existential graphs, for it allows the implementation of quantification and negation. By defining equivalence edges this capacity can be implemented in graph theory.

The following are rules of well-formedness.

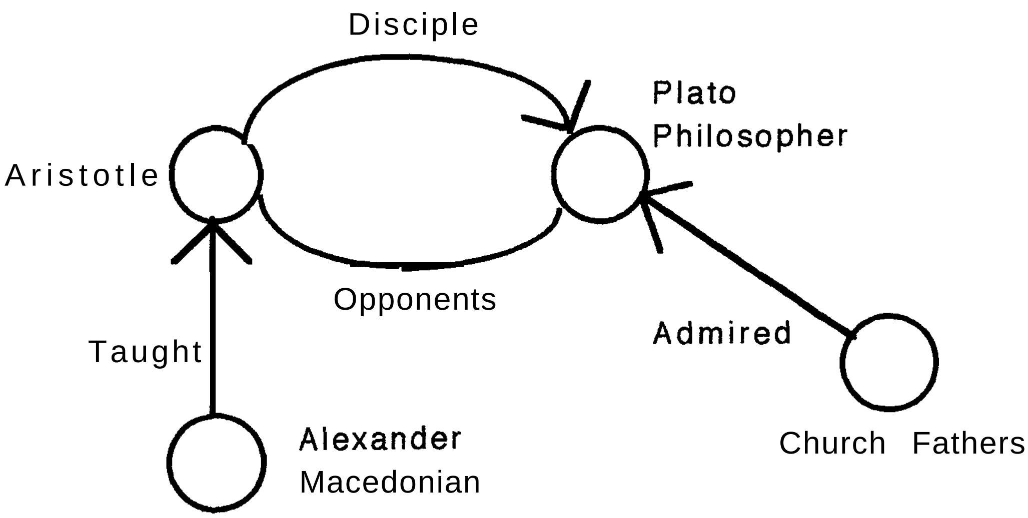
R1: Any graph constructed labeled or unlabeled vertices and labeled arcs (including equivalence arcs) is a wfg.

R2: Any wfg that has a set of subgraphs labeled with a set of layer predicate is a wfg.

I shall now proceed to describe the way in which the GL graphs should be interpreted.

Convention (1)
Any GL graph which contains no labeled subgraphs, nor equivalence arcs, can be interpreted to assert the existence of objects described by the labeled vertices standing in the relations described by the labeled arcs.

For example:



Is equivalent to the following First Order Logic propositions.
(Ew,x,y,z)(Alexander(w) & Macedonian & Aristotle(x) & Plato(y) & Philosopher(y) & Church-fathers(z) & taught(x,w) & disciple_of(x,y) & oppenent_of(x,y) and admired(z,y))

Convention (2)
The equivalence arc asserts the identity of the two entities described by the vertices at the endpoints of the arc. Any two vertices that are connected by a path of equivalence arcs can be considered connected by an equivalence arc.

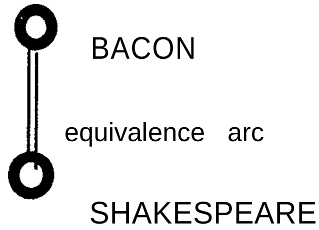
For example:

0 BACON

0 SHAKESPEARE

Asserts
(Ex,y)(Bacon(x) & Shakespeare(y))

However, the following



Asserts:
(Ex,y)(Bacon(x) & Shakespeare(y) and Identical(x,y)
or alternatively:
(Ex)(Bacon(x) & Shakespeare(x))

To be expressively complete a language must be capable of expressing the following:
(1) Any n-ray predicate.
(2) At least one Quantifier **(Ex) or (x)**
(3) Any of a number of sets of logical connectives. For example:

~, &	not, and
~, v	not, or
~, ->	not, **implies**[17]

The first two conventions allows the expression of the Existential quantifier, and the logical connective "&".

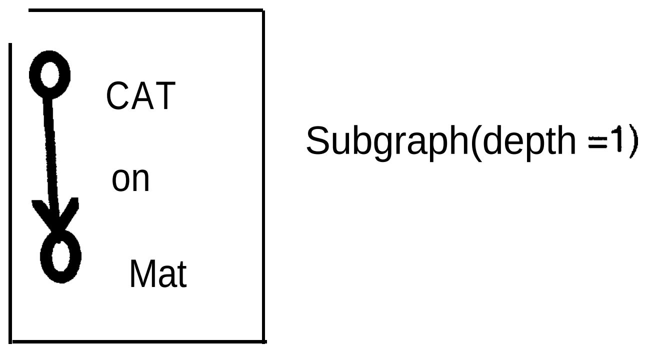
This system does not by itself form a complete system of logic, for it is unable to express not. And without the capacity to express "~", the logic is unable to deal with universal quantification either. If there is a "~" in the symbolism then any universally quantified statement can be expressed because: (x)p is identical to ~(Ex)~p[18]. All we need now is to add a second convention which uses the layers to express negation.

Convention(3)
If a subgraph is labeled a then its contents are denied.

---

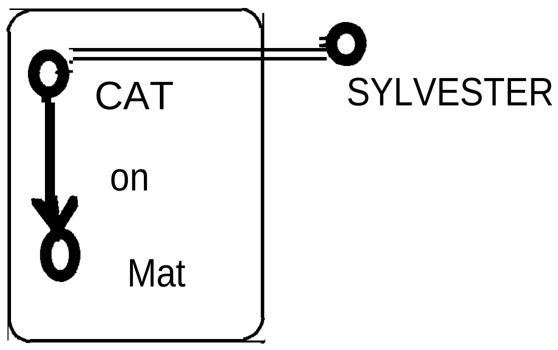
[17]See Howard Delong, **A Profile of Mathematical Logic,** p.138.
[18]Several logic systems exploit this feature, in particular Irving Copi's system RS_1, presented in **Symbolic Logic, see** p. **244.**

For example the following graph



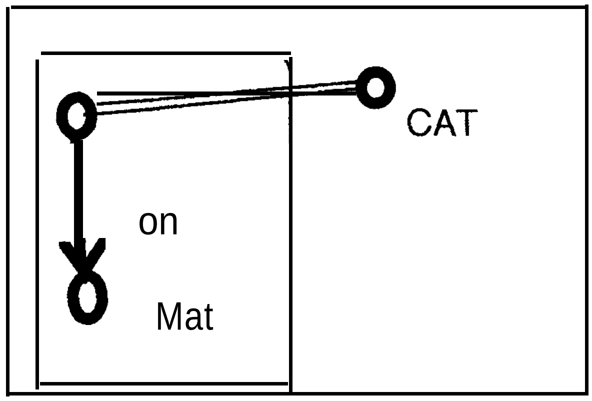
Expresses the following First Order Logic proposition:
~(Ex,y)(Cat(x) & Mat(y) & On(x,y))

This GL graph:



Asserts:
(Ex)(Sylvester(x) &~(Ez,y)(Cat(z) & Mat(y) & On(z,y) & identical(x,z))

The following graph asserts a universally quantified statement:



Translated directly by the convention it is:
~(Ex)[(Cat(x)) & ~(Ey,z)(Identical(x,y)& on(y,z) & mat(z))]

Which is equivalent to the statement
(x)(Cat(x) -> (Ey)(Mat(y) & On(x,y))

which in plain English means “if there is a cat then it is on the mat”.

2.7 The Existential Graphs and GL.

Unlike the Existential Graphs which are able to exploit the topological features of the page and for the Lines of identity, which assert the existence of a thing (analogous to a vertex in GL) to expand on the page, Graph theory is composed of discrete entities.

The way in which this is resolved is by using the equivalence arcs to serve of function of lines of identity thereby allowing the identity of individuals to be asserted across the boundaries of layers. The equivalence arcs allow for the identity of individuals to be asserted on more than one layer of the graph. The layers of GL serve the same formal role as the “cuts” in Peirce’s system. For the purpose of clarity I will now give a translation of the relevant rules of inference for the existential graphs and GL, these rules make it possible for GL to be a complete system of logic. Though limiting the inference that are allowable, to those inferences is very limiting. just as limiting the inference that are legitimate in First Order Logic to modus ponens make certain sorts of inferences intractable.

RULE FOR THE EXISTENTIAL GRAPHS	RULE FOR GL
R1: The rule of Erasure. Any evenly enclosed graph on an evenly enclosed portion of a line of identity may be erased.	R1: Any subset of the representation that is on layer of even depth can be removed from the GL symbol.
R2: The Rule of Insertion. Any graph may be scribed on an area oddly enclosed and two lines of identity on an oddly enclosed area may be joined.	R2: Any graph my be instantiated into an layer of odd depth. And any two vertices in a layer of odd depth may be associated with an equivalence arc.
R3: Rule of Iteration. If a graph P occurs on SA or in a nest of cuts it may be scribed on any area not part of P, which is contained by P. Consequently (a) a branch with a loose end may be added to any line of identity, provided that no crossing of cuts results from this addition., (b) any 1oose end of a line of identity may be extended inward through the nest cuts.	R3: Rule of iteration: Any subset of a graph symbol may be instantiated on a layer that is below and a subset the layer that it originally occurred upon. And any vertex may be instantiated into lower layers and connected with a an equivalence arc to a unique element. in the upper layer.
R4: The Rule of Deiteration, Any Graph whose occurrence could be the result of the rule of iteration may be erased.	R4: The Rule of Deiteration, Any subset of a graph symbol whose occurrence could be the result of the rule of iteration may be removed from the graph symbol.
R5: Rule of the Double Cut: A double cut may be inserted around or removed (where it occurs) from any graph on any area. And these transformation will not be prevented by the presence of lines of identity passing from the outside of the outer cut to the inside.	R5: Rule of the double layer: A double layer may be instantiated or removed upon any subgraph. (A double layer is a labeling of a subgraph with two layer labels of depth n, and n+l)
The Remaining Rules of Transformations apply to Peirce's Gamma graphs which contain modal operators.	

2.8 Completeness and Inference

Howard Delong defines two notions of completeness?

> For the notion of completeness we need to define two different senses of the word: *expressive completeness* and *deductive completeness*. The general idea of expressive completeness is that a formal system be able to 'express' all statements whether true of false. To define the concept we

"Howard Delong, **A Profile of Mathematical Logic,** p. 132.

> must make reference to the an interpretation. A formal system is expressively complete if under the intended interpretation it is possible to express all the sentences, (both true and false) of the informal theory. Expressive completeness is related to primitive symbols and formation rules, and is a semantical property.

I have sketched a method by which propositions of First Order Logic can be translated into GL representations. I have done this by using Peirce’s existential graphs as a prototype. If this is holds true, which I believe it does, for all possible First Order Logic propositions then it can be assumed that:

> Anything that can be represented in First Order Logic can be represented in GL.

Or more precisely:

> For any finite set of propositions in First Order Logic there exists a corresponding GL symbol that has a finite number of elements.

What I have not (apparently) shown is that GL can form the basis of a complete system of Logic. For I have not even begun to describe rules of inference that can act upon the GL symbols. Although the rules of inference defined for the Existential Graphs can be easily translated into rules for inference for GL, it is not necessary to produce a complete system of inference procedures to show that one exists.

I have however, laid the groundwork for showing that it is possible to specify a system of inferential rules that can make GL an inferentially complete logic. The question remains as to the validity of the thesis that if a symbol system is expressively complete whether or not there exists a system of inference rules which would make that language an inferentially complete logic system . Put another way the question as to the validity of this claim rests on following question:

Can one specify an equivalent rule of inference in a Graph theoretic langauge that has the same inferential role as a rule specified for First Order Logic?

Or put more simply:

Is there a limitation to the way in which production rules can be expressed for graphs, a limitation which does not affect production rules for strings (such as First Order Logic expressions)?

If there is such a limit then there may not be a set of graph transformations that are specifiable which form a set of inference rules which would make the language of GL expressively complete. If there is no such limit than any rule definable for a string based langauge such as first order logic is definable analogously for a graph language such as GL.

I am going to examine inference from the perspective of formal language theory. Formal Language theory was first developed by Emil Post. He examined logical inference as a system of rewriting. His approach has been summarized as follows:

> Emil Post studied formal systems of logic, as did most of the pioneers in the area of computability. He abstracted formal systems as rewriting systems, where rewriting corresponds to forming a conclusion from a set of premises. A *Post System* [also called a production system or a cannonical system] consists of a finite set P of inference rules in the form
>
> $\alpha_1. . . , \alpha_m \rightarrow \alpha$
>
> together with a finite axiom set S. The α_i are the premises, from which we infer the conclusion α.[20]

Noam Chomsky in his study of formal grammars examined the ways in which rewrite rules can be specified. A grammar is a method of specifying the ways in which productions can be specified. Different Grammars have different capabilities with regard to their capacity to generate certain strings of symbols.

---

[20]Derick Woods, **Theory of Computation,** p. **82.**

There is a particular class of grammars called Context Free Grammars. This set of Grammars is *Turing Complete* in that any possible transformation of symbols that can be accomplished by a computer or a human being using only the formal rules, in a finite amount of time, can be specified by the productions that are specifiable in Context Free Grammars.21 This is the meaning of the concept Turing Complete. A *Turing Complete* language, is way is specifying productions, such that all possible sets of allowable transformations can be defined within it.

The important question is whether or not there exists a Turing compete class of Graph Grammars, If there is not a Turing Complete class of Graph grammars then there are productions that can be specified for strings for which no analog exists for Graphs. However, there are examples of Turing Complete Graph Grammars which are used in practice.22

Although I have not defined a set of logical rules which make GL a complete system of inference, the *Turing Completeness*, of graph grammars (the set of specifiable graph algorithms) does suggest that one exists. In any case any algorithm, or series of inference steps, that can allow one to reach a conclusion can be specified for a graph theoretic language, with the same amount of detail as it can be specified for a string based langauge.

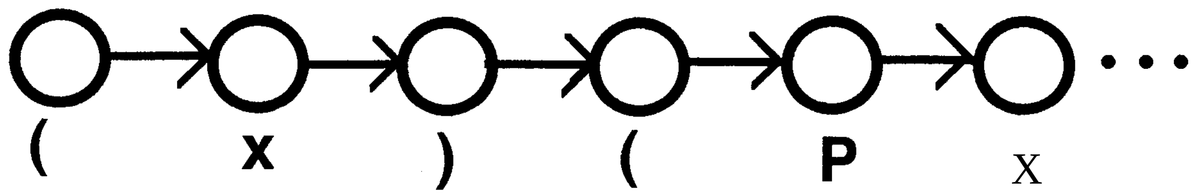
There are three issues here which need addressing. The first is a matter of purely theoretic importance. The second is a matter of cognitive relevance. The third is a matter of practical import.

The first point that I would to make is that logic is reducible to a theory of labeled graphs. For any string is also a graph. One could define the labeling of the graphs such that it is mimicry of algebraic notation.

---

[21]It is an interesting aside to note that the propositional calculus can have an inferentially complete system of rules of inference that can be expressed using a context sensitive grammar.Such as the system P of Church.
[22]Such as the graph grammars that form the basis of the **S-graph** graph algorithm package, which is a product of the Universitat Passau, and is available freely for all academic purposes via ftp from the site **<ftp.uni-passau.de>**

One could define the Graph Language so the symbols looked like this:



And then take any given system of First Order Logic, and translate the rules of well formation and the inference rules. But such a system totally misses the point of a diagrammatic Logic. The day to day using of diagrams allows for the use of cognitive facilities which are otherwise used only for vision processing. The interesting thing about diagrammatic thinking is that it is part of a larger perceptual process.

The second point was partially addressed in the previous paragraph. One does not interpret a diagram in the same manner as one interprets a sentence. The cognitive aspects of diagrammatic reasoning are very important. The processes that the brain undergoes when presented with a diagram or a picture are not the same as those that it undergoes when presented with a discursive representation. The way in which one infers from a diagram is different than the way one infers from a sentence.

The third point is one of practical import. The question of practical relevance, is not whether or not a there is a universal system of inference that is truth preserving but whether or not the solution to the task at hand makes an error or solves the problem. The question as to whether or not we can specify a system that in all cases can be applied to solve a problem, is in day to day, replaced by the question, can this problem be solved this problem in the time allotted. The local completeness for a graph or diagrammatic inference technique is an important issue. Whether or not a given inference procedure is sound can be examined formally for a given inference technique. However, the as the Graph Grammars are Turing complete, it is possible to specify the graph productions that would make the inferences that are needed, and the soundness of the productions can be analyzed on a case to case basis.

I would like to end this section with a remark on pragmatism. Whether or not the system of rules we are using to come to a conclusion is a subset of a complete and sound logic system has little bearing on day to day life. In practice what is of prime importance is whether or not the system we are using can give a solution to the problem at hand. The question of universal completeness should be supplemented by the questions of applicability and whether or not the method of inference taken will solve the problem at hand.

2.9 **Conclusions**

The conclusion of this section is as follows:

> **Diagrammatic reasoning has the potential to be formalized by using graph theory. Graph theory can provide a method of defining formal procedures for making inferences with diagrammatic symbols.**

This can be demonstrated by the ability of graph theory to formally deal with diagrammatic representations. Because of this it is possible for any formal inference procedure that is done in First Order Logic to have a corresponding inference procedure in a Graph theoretic language. Granted there may be diagrams that are difficult to formally represent using Graph Theory, but so to are there sentences that it is difficult to represent in First Order Logic. For example "John is running very quickly".

3. Inference Concepts in Graph Based Logics

The thesis of this section is as follows:

> **Diagrammatic symbol systems support a slightly different conception of inference than do discursive systems such as First order Logic.**

In order to demonstrate this I will sketch the sorts of inferences that are involved in graph theoretic logics..

3.1 Foundational Concepts

I would like to introduce the term "grounded graph". A grounded graph is a symbol which affirms or denies the existence of entities, their properties and/or their arrangement.

This concept has an analog in classical logic. Aristotle defines a premiss in the Prior Analytics as " a sentence affirming or denying one thing of another."[23]

However, Aristotle's logic does not include relational terms. The definition can be expanded by extending it to any sentence that affirms or denies a property to an object or an arrangement to a complex of objects.

A grounded graph can be seen as a being like a premiss in that it is accepted as being the case. As a premiss is characterized by its not being doubted, so to a grounded graph is characterized by the absence of doubt as to the truth of what is being conveyed by it.

[23]Prior Analytics 24a. 16

As this is a discussion of logic, the epistemological status of a grounded graph is not relevant. It merely a symbol which is assumed to represent a true state of affairs.

A **generated graph** is one which is created by the application of an inference operation or a series of inference operations on a grounded graph.

The big difference between GL (and the Existential Graphs) and a discursive system (Classical and Modern systems) is that in the former there are no minor and major premises. The system is a natural deduction system. However, there is never the case when there is more than one grounded symbol for any given inference.

3.2 **The types of inference**

There are three types of inference that I will define for graph based systems (such as GL). They are:

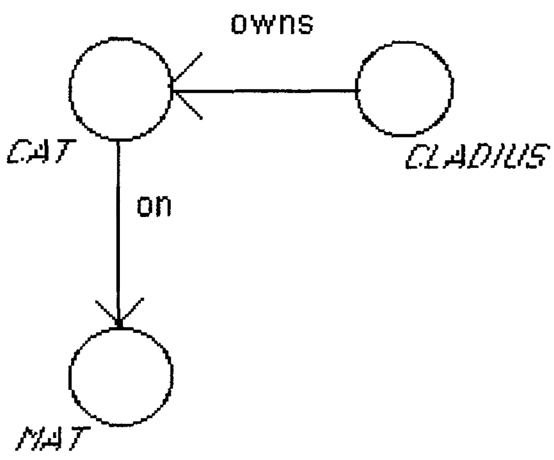
> **(1) Inference by Subgraph**
> **(2) Inference by Production Rule**
> **(3) Inference to Second Order Concepts**

There is a fourth sort of inference that can be performed using a Graph based language and that is inference by monomorphism. Inference by monomorphism is used for pattern recognition tasks. I shall not go into such inference techniques for two reasons. They are mathematically technical and they are not a province of deductive logic. Although pattern recognition may someday provide the framework for a correct theory of induction, I shall not discuss that subject in this thesis.

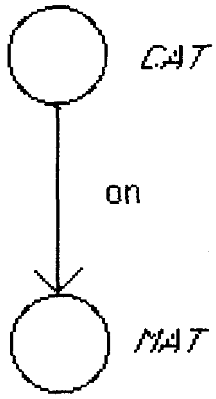
I shall provide all three grades of clearness to each form of inference. That is, I shall provide for each, an example, a definition and a guide to its use.

3.2.1 Inference by Subgraph

By inference by **subgraph** I mean the following sort of inference:



THE GROUNDED GRAPH



THE GENERATED GRAPH

Inference by **subgraph** is a transformation from a grounded graph to a generated graph which is a proper subset of the grounded graph.

Inference by **subgraph** is the same sort of logical operation as the following formal inference in propositional logic:

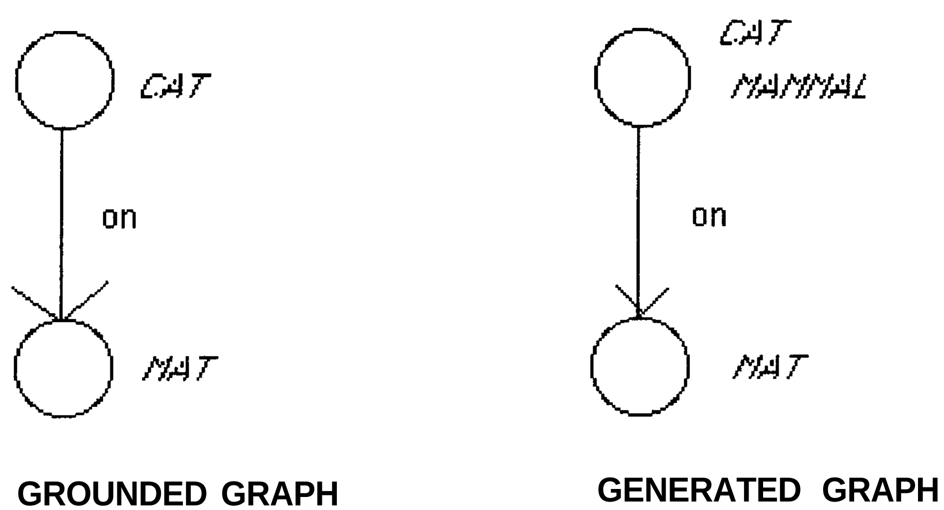
$$\underline{\mathbf{P_1} \,\&\, \mathbf{P_2} \,\&\, \mathbf{P_3} \,\&\, P_4}$$
$$\text{therefore } \mathbf{P_1}$$

Such inferences, called weakening in natural deduction systems, serve several functions. In most applications they serve to remove irrelevant information represented in the graph. Such inferences allow for a transformation of the symbol which removes all unnecessary information and thus focuses on that information that is required for the task at hand.

Inference by subgraph also has a practical application in robotics. It serves to make the inference from a three dimensional symbolic representation of an object or scene to a symbol that describes how that scene or object can be viewed from a particular angle. Such a system works in practice and has been published in the IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 11. no 3, under the title: **Recognition and Shape Synthesis of 3-D Objects Based on Attributed Hypergraphs** (by A.K.C.Wong, S.W.Lu, Marc Rioux).

3.2.2 Inference by Production Rule

The following is an inference by a production rule:



Production Rule:

To any vertex that is attributed "cat", attibute "mammal"

A production rule is a transformation that alters graph symbols by removing, adding or modifying an element or elements of that graph. Production rules use only the formal aspects of the symbols, which include the labels.

The example above was a production rule that involved only a single graph element. There are production rules that involve more than one element of the graph.

Production rules for graph transformation can be utilized to allow for implementation of any rule in the form:

Pxyz.. -> Qxyz...

They can also be used if the graph is set up to model the evolution of a system, for finding the optimal road layout, shortest route between two places, the vulnerability of a communication or transportation system to breakdown and so on. Inference by production rule is the most useful form inference that can be applied.

3.2.3 **Inference to Second Order Concepts**

A good example of this sort of inference can be found in a recent book on Conflict Analysis[24]. Conflict Analysis is a development of game theory. Unlike classical Game Theory modern techniques of conflict analysis do not use cardinal utility.[25] However, there is some similarity in the techniques used in conflict analysis and those used in classical game theory. The primitives are descriptions of game states and the movement options of each player. From these primitives are derived the various equilibrium concepts. These equilibrium concepts are game states that satisfy certain conditions.

This sort of inference is possible in a graph theoretic language. The recent book **Interactive Decision Making** used graph theory as its primary method of representing interactive decision situations. That is to say the game states, and the individual players possible moves are all represented by graph elements. The preferences for game states are represented by the predication of

[24]**Interactive Decision Making: The Graph Model for Conflict Resolution,** by Liping Fang, Keith W. Hipel, D. Marc Kilgour.
[25]Beyond all the conceptual problems with cardinal and intersubjective utility, the practical problem of acquiring the information in a real world scenario is impossible,

preference values to the vertices corresponding to game states. The inferences can be implemented in graph transforms and the final predication pertains to the graph in its entirety, and not to individual elements that comprise the game state.

Equilibrium concepts are game states, but do not say anything about the game state itself independently of the remainder of the graph symbol. They are something said of symbol in its entirety. Thus they say nothing about the graph element alone, but rather say something about the entire graph.

An inference to a second order concept is an inference from things predicated of elements of the grounded graph to things predicated of the graph in its entirety.

Inference to second order concepts is a form of inference that is involved in situations where the graph is intended to represent a phenomenon, and the elements of the graph correspond to elements of the phenomenon. There are times when we wish to describe a phenomenon and make statements about the phenomenon qua phenomenon rather than about the components of the phenomenon. This sort of inference is common when the graph is used to model a phenomenon such as a conflict situation, or any sort system that is modeled using graph theoretic techniques. Graph theory can provide a method of modeling complex systems in a manner that is more effective than using a simple verbal description.26

Inference to second order concepts has no direct analog in First Order Logic, for the conclusion says something about the whole on the basis of its parts. The definition of part/whole relations has not been seriously explored in studies of First Order Logic. The inferences that are made in a system of First Order Logic do not easily support the predication of a property to a complex of entities from the properties of the component entities and their structural relations. I do not want to claim that this is impossible in First Order Logic, only make the point that such inferences are not easily accomplished in such a

[26]This is accomplished by allowing the predication of vector quantities to the elements of the graph

system. Nor does the traditional conception of inference for discursive systems facilitate an understanding of such inferences.

3.3 Truth, Conventions and Inference

In this section I would like to examine the practical problem of the truth of a diagrammatic symbol. Practical problems of truth deal with questions of the truth of an utterance or expression of a symbol. Diagrammatic symbol systems are always interpreted through conventions. The same is true of written or spoken English, although its interpretation through conventions is not as apparent.

We come to realize an expression is problematic when we notice a confusion of symbols. For example, I ask someone to calculate the area of a rectangle and they use the formula:

Area = (length x width) + 10.

This is apparently an error. But it is an error which manifests itself in a confusion of symbols. The symbol “Area” was intended to convey conception C but actually conveys conception C’. It is a confusion of symbols that leads us to understand that there is an error -- somewhere.

An analogous situation occurs when one is reading a map. On attempts to find the best route to City A, but through an error of map reading one follows the route to city B. The confusion involved is in assuming that the route in question leads to city A. Just as the confusion in the above result is in assuming that the right hand side of the equation conveys the same concept as the left side of the equation.

The problem with the rectangle measurer can be analyzed in two ways. (1) I could inform him/her of the error of his/her calculations. (2) Alternatively I could accept his/her calculations and adapt them to my own conception of “area” by subtracting 10. If one understands how someone came to a conception then one can understand how that conception can be interpreted within one’s own conceptual framework. Diagrammatic symbols make this point apparent for they are always interpreted through conventions. There are times when we are presented with symbols that have no co-ordinate with our own conception of things. But we must realize that we interpret any expression through our own set of conventions. Whether or not we accept a symbol that is presented to us depends upon its coherence with our own experience. When we deny a conception that is conveyed to us, there is no rule or procedure to decide whether we are interpreting the symbol correctly, or incorrectly.

In any case, when someone presents us with a symbol that conveys a conception which we believe to be false, we must realize that there are two approaches that we can follow. We can deny the conception that is conveyed to us or we can try to find the source of the confusion.

4. Conclusion:

This thesis addresses three issues around diagrammatic logic. The issues are:

(1) Do diagrammatic systems form a symbolism that is capable of conveying conceptions and serving as a platform for calculation?

The answer to this question is yes. They can serve as means of conveying conceptions and as a support for calculation.

(2) Does diagrammatic reasoning have the potential to be formalized by using graph theory? Can graph theory can provide a method of defining formal procedures for making inferences with diagrammatic symbols.

The answer here seems to be yes, graph theory provides a framework for representing diagrammatic symbols and it also provides a framework for describing the sorts of operations that can be done within a framework of a diagrammatic formalism.

(3) Is inference different for diagrammatic symbolisms than **non-** diagrammatic symbolisms?

When making inferences using a diagrammatic system one often makes an inference about a complex on the basis of its parts and their structural relationship. First Order Logic does not easily support this kind of inference.

Bibliography:

Aristotle, **The Basic Works of Aristotle: Edited and with an introduction by Richard Mckeon,** (Random House, New York, NY., 1941)

Biggs, Norman., **Graph theory, 1736-1936,** (Clarendon Press, Toronto, Ont., 1976.)

Bollobas, Bela., **Graph theory** (Springer Verlag, New York , NY, 1979)

Bondy, J. A, Murty, U.S.R, **Graph theory with applications, (** North Holland, New York, NY, 1976)

Church, Alonzo, **Introduction to Mathematical Logic.** (Princeton University Press, Princeton, NJ., 1956)

DeLong, Howard, **A profile of mathematical logic,** (Addison-Wesley Pub. Co., Reading, Mass.,1970)

Fang, Hipel and Kilgour, **Interactive Decision Making,** (J. Wiley, New York, NY, 1991)

Langer, Susanne K., **Philosophy in a new key; a study in the symbolism of reason, rite, and art** (Harvard University Press, Cambridge, Mass., 1957)

Roberts, Don D, **The Existential Graphs of Charles S. Peirce**, (Mouton, The Hague, Nethertlands, 1973)

Schalkoff, Robert J, **Pattern recognition : statistical, structural, and neural approaches,** (J. Wiley, New York, NY, 1992)

Wood, Derick, **Theory of computation,** (J. Wiley, New York , NY, 1987.)

A.K.C.Wong, S.W.Lu, Marc Rioux, “Recognition and Shape Synthesis of 3-D Objects Based on Attributed Hypergraphs” **IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 11. no 3, 1988.**