Rhetorical Structures in Diagrammatic Logic Systems

Frithjof Dau, Technical University, Darmstadt, Germany
dau@mathematik.tu-darmstadt.de

Abstract

Diagrammatic logic systems are formal, visual representations of mathematical logic. In the first part of this paper, it is investigated in which sense diagrams are, from the viewpoint of their readability, superior to symbolic notions of logic. This is mainly based on two facts: Firstly, in diagrams information can (partly) be represented in an iconic way. Secondly, the spatial arrangement of signs in diagrams provides the possibility to get rid of some technical overhead known from symbolic notions. In the second part of the paper, it is pointed out that the representation of information in diagrams can be varied on two different levels: The type- and the token-level of signs. This fundamental difference to symbolic notions provides further possibilities to represent information in rhetorical different ways.

The considerations will be exemplified with well-known diagrammatic logic systems, namely Peirce's existential graphs and Sowa's conceptual graphs, as well as the RDF graphs known from semantic web.

Keywords: Diagrammatic Reasoning, Logic, Rhetorical Structures, Existential Graphs, Conceptual Graphs

1 Introduction and Motivation

The research field of *diagrammatic reasoning* investigates all forms of human reasoning and argumentation wherever diagrams are involved. This research area is constituted from multiple disciplines, including cognitive science and psychology as well as computer science, artificial intelligence, logic and mathematics. In this research field, much effort has spent in the investigation when and how diagrams are easier to comprehend by humans than symbolic notions.

This paper focuses particularly on diagrammatic logic system, i.e. formal representations of mathematical logic which have diagrammatic representations. Examples of diagrammatic logic system are Peirce's existential graphs and some contemporary formal elaborations (Zeman, Shin), John Sowa's conceptual graphs, again with several formal elaborations (Chein/Mugnier et al, Wille et al), or RDF (Resource Description Framework), which is the underlying description language of the semantic web. All these examples are graph-based versions of logic. Another family of diagrammatic logic systems are Venn diagrams, Euler diagrams or spider diagrams (Shin, Hammer, Howse et al).

This paper discusses some specific differences between the usual symbolic and the graph-based notions for logic, and it aims to work out in which respects the graph-based approach is better suited for humans. This discussion will be carried out on several examples, but, due to space limitations, no introduction in existential graphs, conceptual graphs or RDF is provided. I assume that the reader is somewhat familiar with these systems, and that the examples can be understood even without a deeper knowledge of the systems. Otherwise, introductions for existential graph can be found in [12, 17, 16], for conceptual graphs in [19], and for RDF in [14].

The paper is organized as follows: In the first section, I introduce some specific notions for diagrams based on Peirce's semiotics and his existential graphs, which are helpful for the scrutiny of this paper. In the next section, some aspects in diagrams which ease the notion of formal logic (ease with respect to their understandibility by humans) are discussed. Finally, in the last sections, it will be shortly discussed why diagrams provide better rhetorical means (in the understanding of the ability to emphasize some information to humans) than symbolic notions.

2 Crucial Properties of Diagrams

At the beginning of modern logic at the end of the 19th century, two diagrammatic systems had been invented in order to formalize logic. The first system is Frege's Begriffsschrift, where Frege aimed to provide a formal universal language. The other one is the systems of existential graphs (EGs) by Charles Sanders Peirce, which he used to study and describe logical argumentation. Before Peirce developed his EGs, he contributed much to the development of the symbolic approach to mathematical logic as well. Thus, Peirce was very familiar with both approaches, the diagrammatic and the symbolic, to logic. As he was interested in an instrument for the investigation of logic, he discussed the differences, the advantages and disadvantages, of these two approaches to a large extent. Among philosophers, Peirce is better known for his extremely comprehensive semiotics

(and for his pragmaticism), and his discussion of symbolic and diagrammatic approaches to logic is based to a large extent on his semiotics. Peirce considered his diagrammatic system of EGs to be far more perfect for the investigation of mathematical reasoning than the symbolic approach he developed as well. His philosophical and semiotic arguments for this preference will give us valuable insights into diagrammatic logic.

2.1 Indices, Icons, Symbols

Peirce classifies the signs into several trichotomies, which are often again subdivided into other trichotomies, etc.[1] For the purpose of this paper, it is helpful and sufficient to consider one of Peirce's trichotomies, namely the division of signs into *icons, indices, and symbols*, a division of which Peirce says that this is the most fundamental (see [15], 2.275).

In his writings, Peirce offered in many places a definition of a sign, but the overall idea hardly differs in all these definitions. A *sign* is something which denotes some fact or object to a mind. That is, an entity is established as a sign when it is interpreted by a mind to represent another object. We have therefore three things to deal with: Firstly, we have the sign, secondly, we have the object which is represented by the sign, and thirdly, we have a mind in which the representation causes a subjective idea of the represented object. This subjective idea is what Peirce calls 'interpretant thought', or, on many place, simply 'interpretant', and the sign is "mediating between an object and an interpreting thought" ([15], 1.480).

A sign can be perceived by a mind in different ways, that is, the interpretant thought emerges differently. This leads to Peirce's division of signs into icons, indices, and symbols. Among all classifications of signs by Peirce, this classification is the most important one for the investigation of diagrams, and in fact, in Peirce's writings on EGs, we find discussions on icons, indices and symbols in several places. In the following, these types will be defined and discussed. But before I do so, I have to stress that a sign rarely belongs to one of these classes exclusively. Often, a sign possesses iconical, indexical and symbolic qualities. Thus, the definitions which will be given should be understood in a prototypic manner.

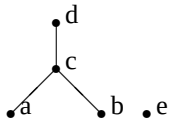
A *symbol* is a sign which gains its meaning by mere convention. The best example for symbols are words. In [15], 4.447 Peirce's writes: "Take, for example, the word 'man'. These three letters are not in the least like a man; nor is the sound with which they are associated". There is no necessary reason whatsoever that the word 'man' is interpreted as it is. A symbol is a sign if "is constituted a sign merely or mainly by the fact that it is used and understood as such", Peirce explains in [15], 2.307. The meaning of a symbol is based on laws, habits or conventions in a community. In order to understand the meaning of a symbol, we have to learn what the symbol denotes. Of course, there are other examples of symbols which are no words, like flags which are symbols for countries, or colors are often used as symbols (for example, in Europe, the color black is a symbol for mourning, in China it is the color white).

In contrast to symbols, indices and icons have a direct connection to the represented objects. Roughly speaking, for indices it is a physical causality, for icons it is resemblance. In [15], 2.299, Peirce writes: "The index is physically connected with its object; they make an organic pair, but the interpreting mind has nothing to do with this connection, except remarking it, after it is established." Some examples Peirce of are: Smoke is an index of fire (*that* which causes the smoke), weathercocks indicate the direction of the wind or low barometers indicate moist air and coming rain (all examples can be found in [15], 2.286). This induces an important difference to symbols: Symbols may refer to ideas, or abstract concepts (like 'man', which refers not to a specific man, but to the concept of man), but an index always refers to an individual, singular entity.

The most important type for diagrams is the class of *icons*. The essential feature of an icon is a similarity between the sign and the denoted object, i.e. that the "relation between the sign and its object [. . .] consists in a mere resemblance between them" ([15], 3.362). A simple example for an icon is a painting, let's say, of a pipe. A person who sees the painting will immediately understand that the painting represents a pipe.[2] No convention is needed for this, thus, the painting is not a symbol. Moreover, there is no direct, physical connection between the pipe and its painting[3], thus, the painting is not an index. This becomes even more evident if we considered the painting of an object which does not exist, for example, an unicorn. Other, popular examples for icons can be found in public buildings, like signs for restrooms, busses, trains, elevators etc in an international airport.

Iconic representations are much easier to comprehend than symbolic representations. Iconicity is an essential feature which occurs in diagrams. Let us consider an example from mathematics, namely Hasse-diagrams which are used to represent ordered sets. Below, an ordered set $(O, \leq)$ and a possible representation of this ordered set as a Hasse-diagram is given:

$$(\{a, b, c, d, e\},\\ \{(a,a),(b,b),(c,c),(d,d),(e,e),\\ (a,c),(b,c),(a,d),(b,d),(c,d)\})$$



In the diagram, an element of the ordered set is *above* an-

[1] The use of trichotomies is based on Peirce's three fundamental categories called firstness, secondness, and thirdness. But a discussion of these categories goes far beyond the scope of this paper.

[2] Recall Magrittes famous painting 'Ceci n'est pas une pipe', where he emphasizes the difference between the painting and the painted object.'

[3] This would be different if we considered a *photograph* of a pipe. In fact, photographs are for Peirce a mixture of indices and icons.

other element, if and only if there is a path from the dot representing the first element *upwards* to the dot representing the second element. Thus, the order $\leq$ is represented iconically. Without doubt, the order is easier to comprehend from the Hasse-diagram than from the symbolic notion given before. Moreover, in the Hasse-diagram, more facts about the represented order can immediately be read off. For example, we easily see that a, b and e are the minimal elements of the order. This advantage of iconic representation is described by Peirce in [15], 2.279, where her writes that "a great distinguishing property of the icon is that by the direct observation of it other truths concerning its object can be discovered than those which suffice to determine its construction." [4]

It is crucial to note that a diagram *has* iconic features, but usually, it *is* not an icon. Peirce writes in [15], 2.282 that 'many diagrams resemble their objects not at all in looks; it is only in respect to the relations of their parts that their likeness consists.' For Peirce, even algebraical equations are not purely symbolic, but they have some iconic qualities. But without doubt, other diagrams bear much more iconicity than algebraic equations. Diagrams are representations, representing some facts, and they "ought to be as iconic as possible". Nonetheless, there is no sharp distinction between diagrammatic and non-diagrammatic representations, but the discussion so far gives rise to a possible fixation of the term 'diagram':

> **Diagram:** Given a representation of some facts, the higher the amount of iconically represented facts is, the more the representation is a *diagram*.

Especially EGs provide several iconic features. For example, writing two graphs G_1 ,G_2 side by side is perceived as G_1 *and* G_2, and the juxtaposition of G_1 and G_2 is semantically interpreted as the conjunction 'and' of the graphs.

The iconicity of diagrams is already discussed in literature, but often different technical terms are use to describe this feature, like "'directness' in the matching between representing and represented relations", "'analogical' nature of diagrams, 'constraint preservation' or the 'homomorhpism' of diagrammatic representations" (see [11]).

Concerning our example, the representation of the order $\leq$ is partly iconic, but it is partly symbolic as well. For example, it is a convention that not all pairs $(x, y) \in \leq$ are represented with a line; this is only done when x and y are neighbours. A student who gets for the first time in contact with Hasse-diagrams has to learn this convention, that is, she has to learn which facts of the structure are represented in the diagram, and how this is done. On the other hand, she has to learn which properties in the diagram are indeed representing facts. It is sometimes puzzling for novices that the positions of the dots may vary to some extent, and that the length or arc of a line in a Hasse-diagram is of no significance. These considerations does not only show that Hasse-diagrams are partly symbolic, moreover, it shows that it is of great importance to make clear which graphical properties in a diagram represent some facts, and which not. Shin investigates in [17] the use of icons in diagrams to a large extent, and one result of her scrutiny is the following statement: 'Now we may reach the following general conclusion about the representation of icons: When we adopt an icon to represent something in general (as a symbol does), it is important to classify the observable properties of the particular icon into two different categories; properties that are representing facts and properties that are not.'

[4] In the modern research field of diagrammatic reasoning, Shimojima coined the term *free ride* for this property of diagrams.

2.2 Types and Tokens

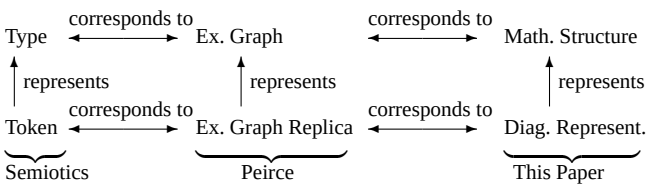
In the last subsection, we have exemplified iconicity with an ordered set and its Hasse-diagram. It is crucial to note that we dealt with two different entities: A mathematical structure and its diagrammatic representation. In any diagrammatic representation of a mathematical structure, we have to prescind from certain graphical properties of the diagrams (e.g. the lengths of lines in a Hasse-diagram), while other properties are important. Let us consider the three diagrams of Alpha EGs in Fig. 1.



Figure 1. Three diagrams of one alpha graph.

The shape of the negation ovals (Peirce called them *cuts*) or the place of propositional variables and other cuts in the area of a given cut has no significance, thus all diagrams convey the same meaning. They can be read as 'it is not true that A and B, but not C hold', i.e., 'A and B imply C'. Peirce did not understand EGs as graphical entities at all. For him, the three diagrams are not *different graphs* with the same meaning, but *different representations*, i.e., diagrams, of the same graph. He writes: "A graph [. . .] is a symbol, and, as such, general, and is accordingly to be distinguished from a graph-replica." In other words: Peirce distinguishes between *graphs*, which are so-to-speak abstract structures, and their *representations*.

This is a crucial distinction, which obviously corresponds to the distinction between *types* (graphs) and *tokens* (graph replicas), as it is known from philosophy. It can be sketched as follows:



The type-token issue if far from being settled; nonetheless, this important distinction helps us to draw the following conclusion: In any elaboration of a diagrammatic logic system, the diagrams should not be defined as graphical entities. Instead, we need a definition of mathematical structures which encompass exactly the facts which are represented in the diagrams, and the diagrams should be understood as (mere) representations of these structures. For example, in any mathematical formalization of Peirce's graphs, cuts should be introduced as abstract entities without any specific shape, but we have to formalize which propositional variables or other cuts are placed within a given cut. The relation 'within' should be captured by an abstract mathematical relation; for example, by a mapping which assign to each cut all other entities which are contained by the cut. Then, in the graphical representation, the abstract 'within'-relation of the mathematical structure is represented by the spatial 'within'-relation of the diagram.

The types are mathematical structures, but, as it is argued in [5], a rigor mathematical elaboration of a diagrammatic logic system can and should be developed without providing mathematical definitions for the representations. Instead of that, it is sufficient if we have conventions – either implicit or explicit — which describe the representations, as well as the relationship between structures and their representations, in a clear and non-ambiguous way.

In mathematical logic, the separation between formulas and their representations is usually not discussed. The following question immediately arises: Why does symbolic and linear notions do not have to cope with the distinction between types and tokens? The main reason for this is the following: The linear notion of formulas corresponds the way ordinary text is written, that is, formulas and text are assembled of signs which are written side by side and which are read from left to right. As we are used to read texts, we are trained as well to read strings which shall represent formulas. In fact, formulas are abstract mathematical entities, and when we read any text on mathematical logic, only representations of formulas are provided. For example,

$$\forall x : human(x) \rightarrow (featherless(x) \vee biped(x))$$
$$\text{and} \quad \forall x : human(x) \rightarrow (featherless(x) \vee biped(x))$$

can be understood as different representations of the same formula, both of them, although looking slightly different, relying to the well-known conventions of text. So we see that even in symbolic logic, we have a separation between formulas and their representations. But in contrast to diagrams, there is no need to discuss the conventions for the representations of formulas, as these conventions are the already well-established cultural conventions of written text.

Different formulas can have the same meaning. Thus, if we want to represent some information by means of a formula, we have syntactically different, but semantically equivalent formulas we can chose from. Another simple formula which is syntactically different to the formula just given, but which has the same meaning, is

$$\forall y : human(y) \rightarrow (biped(y) \wedge featherless(y))$$

But once we have chosen a formula, i.e. a sequence of signs, is chosen, the representation of the formula is uniquely given. This is a crucial difference to graph-based logics, where a given graph can have very different diagrammatic representations. An example are the different representations of the same alpha graphs depicted in Fig. 1. So, in contrast to formulas, the way of representing some information can be varied on two levels: First, we can choose a mathematical structure which represents the information, then we can choose a convenient diagrammatic representation of this structure. So, for representing informations, symbolic and linear notions of logic have a **one layer architecture**, and diagrammatic logic systems have a **two layer architecture**.

The advantages of this two-layer architecture shall be elaborated in the next sections. For a much deeper discussion on the distinction between types and token for diagrams, please see [17] and [5].

3 Advantages of Spatial Representation

Nowadays, nearly any sort of mathematical logic is *symbolic* logic, where the term 'symbolic' can be understood in the Peircean sense. That is, logic deals with formulas, which are linear sequences of symbols. The restriction to a linear notion yields syntactical restrictions, which make formulas uneasy to understand and comprehend by untrained users. These restrictions will be discussed in this sections, and how they can be dismissed in diagrammatic logic systems, where a plane for representing information gives further benefits for the syntax.

3.1 Commutativity and Associativity

Assume we consider some version of first order logic, where we have logical connectives like conjunction and disjunction. Both operations are commutative and associative. Nonetheless, when formulas are introduced, this properties are not captured by the syntax of the formulas.

Let us first consider the commutativity property. For example, $A \wedge B$ and $B \wedge A$ are different formulas, although the trivially have the same meaning. As formulas are *sequences* of signs, all subformulas occur in a specific order in a formula. Due to this syntactical restriction, the commutativity of conjunction cannot be expressed within the syntax of formulas, thus we need in any linear and symbolic account for logic some *derivational rules* which allow to mutually transform the given formulas into each other. In diagrammatic logic systems, we have a different situation: There, we are not bound to the linear notion, thus the commutativity of

the conjunction can already be captured by the syntax. This can be best explained with Peirce's graphs. I have already mentioned that in a convenient formalization of alpha EGs, we could map to each cut the *set* of all all other entities – propositional variables or other cuts– which are contained by that cut. In the semantics of EGs, the each entity contained by a cut can be assigned a truth value, and the cut evaluates to true if the conjunction of the truth values of all these entities evaluates to false. So, when the containment of a cut is formalized as a set of entities, and as sets are not ordered, we see that a syntax like this already captures the commutativity of the conjunction, as we have simply have no means to express a order within a conjunction. In the diagrammatic representation of alpha graphs, this commutativity becomes clear, as we are allowed to scribe the entities contained by a cut into arbitrary places on the plane (provided they are scribed inside the cut). See Fig. 1 so see an example for this convention.

The same considerations can be applied to the associativity of some logical connectives. Usually, when formulas are introduced, conjunction is handled as a binary logical operator, and if the conjunction of more than two formulas shall be expressed, this operator is applied more than once. Then, in order to keep a unique derivational history of formulas, brackets are used to make clear how a formula is composed from subformulas. For example, $(A \wedge (B \wedge C))$ and $((A \wedge B) \wedge C)$ are, strictly speaking, different formulas, although they have trivially the same meaning. In many textbooks on logic, after the definition of formulas, informal conventions are introduced which allow to add or omit brackets in order to improve the readability of formulas. Again, in diagrammatic logic systems, such conventions are not needed, as convenient definitions of the syntax already capture the associativity of conjunction.

This has two benefits for diagrammatic logic systems: First of all, the structures are easier to handle, and the diagrams are easier to comprehend, as there is no need to take care of the associativity and commutativity of logical operators. Secondly, if a diagrammatic logic systems is well designed, we do not need derivational rules in the calculus which encompass these properties of logical operators.

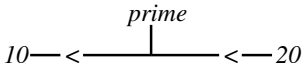
3.2 Variable-Free Logic

Assume now we consider some version of logic where we want to speak about unqualified objects. To provide an example, assume we want to express that there exists a prime number between 10 and 20. Omitting brackets, this could in symbolic logic formulated as follows:

$$\exists x \in \mathbb{N}\,(10 < x \wedge x < 20 \wedge prime(x))$$

In order to refer in the formula to the unknown prime number, we have to use a variable. Of course, variables are the usual means to refer to unqualified objects. As we refer several times in our formula to the prime number, the variable occurs several times in the formula. Due to the linear notion of logic, there is no alternative than using a sign denoting the object such that this sign occurs several times in a formula. Now, if a reader wants to grasp the meaning of a formula where variables occur, she must keep track of all occurrences of all variables.

Diagrams provide different means to denote unqualified objects. In Peirce's EGs, heavily drawn lines, called *lines of identity*, are used to denote objects. These lines are allowed to branch, but a branching heavily drawn line still denotes one object. An EG expressing our sample proposition could be depicted as follows:



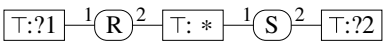
In this diagram, it is much easier to see to keep track of the branching line of identity, especially as it is connected, thus it is easy to understand that this line denotes a single object. Thus, lines of identity provide an (partly) *iconic* way to represent unqualified objects. We see that diagrams provide means to develop *variable-free* notations of logic. The combination of this variable-free notion, together with the iconic quality of the diagrams, make them easier to comprehend than formulas.

Besides the fact that lines of identity are easier to comprehend, the have two syntactical advantages: Firstly, in contrast to variables, there is no separation between lines of identities and *occurrences* of lines of identities. Secondly, in symbolic logic, we need rules which allow to *rename* variables, as the name of a variable has to significance (instead of x, we could have used arbitrary other variables, like y or z, in our formula). For formulas, this is often done by a convention called 'alpha-conversion', or by rules in the calculus. In diagrammatic reasoning systems, there is no a priori need for such conventions or rules.

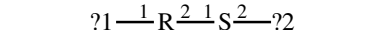
To show another example of a variable-free notion of logic, we consider the relation product of R and S of two relations R, S on a set A. A common mathematical notation is the following:

$$\{(x, y) \in A \times A \mid \exists z \in A : (x, z) \in R \wedge (z, y) \in S\}$$

Next a conceptual graph version of expressing the relation product is depicted. Conceptual graphs are based on Peirce's EGs, but the have a different syntax. In CGs, the generic marker '$*$' is used to denote (existentially quantified), unqualified object, thus we can depict the relation product as follows:



The corresponding EG is even more elegant:



In both graphs, numbered question marks are used to indicate the places of the relation we construct. Please note that for CGs, as we have only one symbol for denoting unqualified objects, again we have no need to distinguish between generic markers and its occurrences, or to introduce conventions or rules to rename this marker. A more comprehensive discussion on this issue can be found in [6].

3.3 Multiple Readings

An essential feature of common symbolic logic is the inductive definition of the formulas, which leads for each formula to a unique derivational history. Formulas are understood to be composed of atomar formulas, and the handling of formulas is based on the atomar subformulas and how the are connected by means of logical operators. The semantics is then carried out along this definition, i.e. the semantics is inductively defined as well. Even the calculi are usually grounded on this inductive understanding (which can, for example, easily observed on natural deduction calculi). The overall approach to symbolic logic is so-to-speak a reductionistic one.

Humans usually comprehend diagrams not in a purely reductionistic manner. Instead, a diagram is first recognized as a whole, before subparts of it are more closely observed. Surprisingly, this more holistic approach to diagrams can be mathematically captured.

Shin investigates in [17] by means of Peirce's EGs the differences between symbolic and diagrammatic logic, and one of her main claims is that the rigor and power of diagrammatic systems is often not understood because diagrammatic system are evaluated in terms of symbolic systems. According to her, one of the essential differences between symbolic and diagrammatic notions for logic is that different readers perceive a graph in different ways, thus obtaining different readings for the same graph. This shall be exemplified with Peirce's alpha graphs.

Strictly speaking, Peirce's alpha graphs provide only two logical connectives: Scribing graphs side by side is understood to be the conjunction of the graphs, and enclosing a graph by a cut is negating it. For propositional logic, it is sufficient to have only the operators $\wedge$ and $\neg$, and indeed, when alpha graphs are mathematically elaborated, they are usually translated to $\wedge, \neg$-formulas of the propositional calculus. But, even Peirce provides aids to read his graphs differentially. For the device of two cuts one inside the other, he explains that they can be understood as a material implication, and he introduces the term 'scroll' for this device.

Consider the left graph with a scroll, where A and B might stand for arbitrary other graphs. Strictly speaking, the translation of this graph to a formula is $\neg(A \wedge \neg(B))$, which is of course equivalent to $A \rightarrow B$. A similar consideration yields that the right graph can be read as $A \vee B$.



The crucial point is that in a graph with several cuts, different readers might perceive scrolls (or the device which can be understood as a disjunction) in different places of the given graph, that is, as Shin says, different readers carve up a graph in different ways. This yields different (but of course semantically equivalent) readings of the same graph. For example, the right graph could read as $\neg(\neg A \wedge \neg B)$, $\neg A \rightarrow B$, $\neg B \rightarrow A$ or $A \vee B$, all of them being equivalent.

Shin claims this feature to be much more natural for diagrammatic systems than a uniquely defined reading. Moreover, she provides a mathematical translation of alpha graphs to formulas which assign to each graph a *set* of formulas, which are obtained by carving up the graph in different ways.

4 Using the Layers for Rhetorical Purposes

After discussing why and how information represented by diagrams is easier to comprehend by humans than symbolically represented information, I will now turn to the rhetorical means provided by diagrams. I have already stated that, in symbolic notions as well as in diagrammatic notions, a given amount of information can have syntactically different, but semantically equivalent representations (in form of formulas or diagrams). Being 'semantically equivalent' is a crisp mathematical notion, but of course, this notion does not capture the fact that among semantically equivalent representations, their meaning is for some of them easier and for others harder to comprehend by human readers. Moreover, if a some amount of information is given, among all possible representations of this information, we can often choose representations which *emphasize* some part of the information to a human reader. The different possibilities of emphasizing some information is what I will call the *rhetorical power* of a representation system. In thew following, the rhetorical power of diagrammatic logic systems will be discussed by means of some examples.

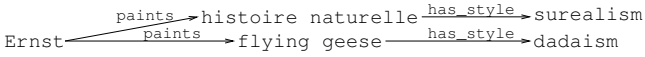
For any representation system, symbolic or diagrammatic, its rhetorical power obviously relies on the fact that it provides different representations of some given amount of information. A first idea for measuring the rhetorical power of a representation system could be: The more different representations a given amount of information has, the better is the rhetorical power of the system. Of course, this understanding is very rough and far too coarse-grained, but it might serve as a first indication. This shall be discussed with a diagrammatic version of the language RDF (Resource Description Framework), which is the underlying description language of the semantic web.

In RDF, information is encoded as statements, which are subject-predicate-object triples of information resources. Mathematically, in the *RDF Primer* [14], an *RDF graph* is defined to be a set of triples (s, p, o), where each triple is understood that s and o are related by the property p.

The term RDF graph is based on the understanding that these sets of triples can be represented in form of labeled graphs, which is explained in the RDF Primer as well: Subjects and objects of the statements are the nodes of the graph, which are connected by directed edges labeled with the respective predicates. For example, the RDF graph consisting of the triples

```
(Ernst, paints, histoire_naturelle),
(Ernst, paints, flying_geese),
(histoire_naturelle,has_style,surrealism),
(flying_geese, has_style, dadaism)
```

could be diagrammatically be depicted as follows:



However, the representation of RDF triple sets as mathematical graphs is only informally described. RDF triples have a peculiarity which does usually not occur in common mathematical logic: Some entities can occur as predicates in some statements and as subjects or objects in other statements.

If the set of properties of an RDF graph is not disjoint from its set of subjects and objects, it is not possible to transform an RDF graph in triple notation to an ordinary mathematical graph (V, E) by assigning to each object or subject a node and assigning to each predicate an edge. In the informal graphical representations for RDF graphs, there are different "workarounds" to cope with this problem (for example, edges which link vertices and edges, or the appearance of the same name both on vertices and edges). A comprehensive discussion of this problem can be found in [8], a short in in [9]. Moreover, it is implicitly assumed that each RDF graph has a uniquely given graph representation. As RDF graphs are *sets* of statements (s, p, o), we have (nearly) no possibility to express the same information in different ways. This is a very convenient feature for the machine processing of RDF (which is one of its main purposes). But as argued, to have a high rhetorical power of RDF, this feature is not desirable.

Let us first cope with the problem that a precise translation of sets of RDF statements and into mathematical graphs is not provided yet. In [8, 9, 7] as well as in [1], the following reasonable graph formalization of RDF graphs is provided: We assign to each *subject, property and object* a node in a mathematical graph, and to assign to each *statement*, i.e. a triple (s, p, o), a hyper-edge which links the nodes of s, p, and o. This way each RDF graph in triple notation can be translated to a labeled, directed hypergraph, thus this graph formalization can be understood as explicit representations of RDF graphs in triple notation as mathematical graphs.

To see an example for this formalization, consider the first four axiomatic triples provided in [10].

```
(rdf:type rdf:type rdf:Property)
(rdf:subject rdf:type rdf:Property)
(rdf:predicate rdf:type rdf:Property)
(rdf:object rdf:type rdf:Property)
```

In Fig. 2, a diagrammatic representation of a graph formalization of the four triples is given. The mathematical graph of Fig. 2 is fragmented into elementary graphs, i.e., graphs which carry atomic information, which directly correspond to the triples of the RDF triple set. Therefore, no information of the RDF triple set is emphasized in its diagrammatic representation.

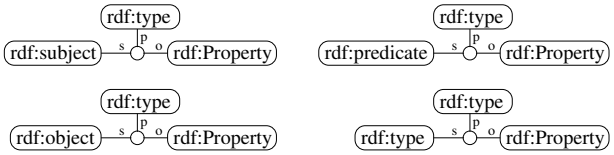


Figure 2. A RDF/DRS graph for the four triples

Compared to RDF graphs in triple notation, their graph formalizations provide richer means to express some given amount of information. This is mainly due to the fact that a vertex in an RDF graph can be incident with several edges. Assume that we have several RDF statements where an entity occurs (as subject, predicate, or object). If we use RDF graphs in triple notation, then we will have a corresponding number of occurrences of this entity in some triples. In contrast to that, if we use graph formalizations of RDF graphs, we can *choose* the number of vertices which stand for this entity. The richer means to express information, together with the various ways a graph formalization of an RDF graph can be graphically represented, provides a better rhetorical power. The two graphs in Fig. 3 are other

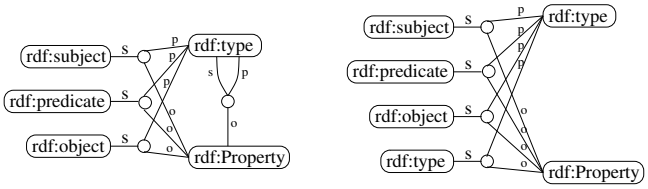
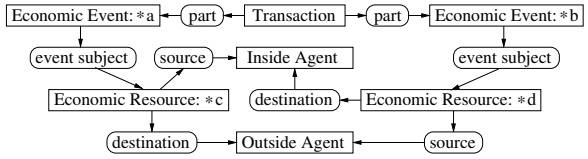


Figure 3. Two other graphs for the four triples

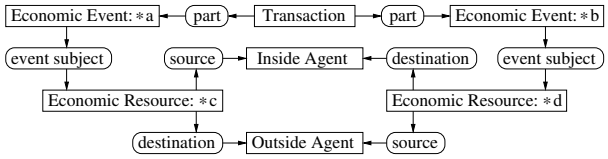
possible graph formalizations of all four triples. The different graphs (layer 1) as well as their resp. visual representations (layer 2) are used to emphasize some specific view on the represented information. The left graph emphasizes that `subject`, `predicate` and `object` stand in the same relation to `type` and `Property`. This is done by choosing a graph where we have three edges which differ only in the first place, namely the vertices labeled with `subject`, `predicate` and `object` (that is, we use the rhetorical means on layer 1), and by placing these vertices on the left

side of the diagram and the vertices labeled with `type` and `Property` on the right, (here we use the rhetorical means on layer 2). It is also also easy to see that `type` plays a special role, as it appears both as subject and predicate of an edge (layer 1). The right graph emphasizes that all four entities `subject`, `predicate`, `object` and `type` stand in the same relation to `type` and `Property`, but to see that `type` differs somehow from the other three names one has to keep track of all labels and realize that two different vertices are labeled with `type`.

Similar to the 'above'-relation in orders and Hasse-diagrams, in diagrammatic logic systems we can have spatial properties which correspond to some represented facts, and we can take advantage of this correspondence. For example, we can emphasize some sorts of symmetries in information we want to represent. Below, you find a CG taken from a paper from S. Polovina, ICCS 2004, describing some economic information.



A closer observation shows that the represented information contains some sort of symmetry. In order to emphasize this symmetry, we can improve the visual representation of the CG by means of layer 2 as follows:



Other examples of taking benefit of the second layer can be a spatial grouping of information (e.g., clustering the elements of an RDF graph due to facts and schemata information), or displaying informations about hierarchies (e.g., ontologies) in a visually hierarchical manner.

5 Further Research

As Shin stresses in [17], there is a 'shared view that there has until quite recently been a long-standing prejudice against non-symbolic representation in logic', which is due to the tendency 'to understand and evaluate diagrammatic systems against the criteria of a symbolic system'. According to her view, this paper aimed to work out some specific advantages of diagrammatic systems for humans, which can only be investigated if these systems are not investigated from the viewpoint of symbolic logic. Of course, this paper provides only a small and rough fragment of such a scrutiny, and much more effort has to be spent to work out the weaknesses and strengths of both, symbolic or diagrammatic, approaches to logic.

References

[1] J. F. Baget: *Homomorphismes d'hypergraphes pour la subsumption en RDF/RDFS.* RSTI L'objet, LMO 04, 2004, p. 203–216.

[2] T. Berners-Lee: *Conceptual Graphs and the Semantic Web*. `www.w3.org/DesignIssues/CG.html`

[3] M. Chein, M.-L. Mugnier: *Conceptual Graphs: Fundamental Notions.* Revue d'Intelligence Artificielle 6, 1992, 365–406.

[4] F. Dau: *The Logic System of Concept Graphs with Negation (And Its Relationship to Predicate Logic).* LNAI, Vol. 2892, ISBN 3-540-20607-8. Springer, Berlin–Heidelberg–New York, 2003.

[5] F. Dau, Types and Tokens for Logic with Diagrams: A Mathematical Approach, In: Pfeiffer, H. D.; Wolff, K. E. (Eds): Conceptual Structures at Work, LNAI, Vol. 3127, Springer, Berlin – Heidelberg – New York, 2004, p 62–93.

[6] F. Dau, *Variables in Concept Graphs*, To appear in the proceedings of the ICCS 2005, Springer, Berlin – Heidelberg – New York, 2005.

[7] F. Dau, J. Hayes: *RDF as Graph-Based Diagrammatic Reasoning System: Syntax, Semantics, Calculus.* Submitted to the 2nd European Semantic Web Conference.

[8] J. Hayes: A Graph Model for RDF. Diploma thesis, Technische Universität Darmstadt, Department of Computer Science, Germany. `purl.org/net/jhayes/rdfgraphmodel.html` (2004)

[9] J. Hayes, C. Gutierrez: *Bipartite Graphs as Intermediate Model for RDF.* In: Proceedings of ISWC 2004, The Third International Semantic Web Conference. LNCS, Springer, Berlin – Heidelberg – New York, 2004.

[10] P. Hayes: *RDF Semantics: W3C Recommendation.* 10 February 2004. `www.w3.org/TR/rdf-mt/`

[11] C. Gurr, J. Lee, K. Stenning: *Theories of Diagrammatic Reasoning: Distinguishing Component Problems.* Mind and Machines 8, Kluwer Academic Publishers, 1998, p. 533-557.

[12] E. M. Hammer: *Logic and Visual Information.* CSLI Publications, Stanford, California, 1995.

[13] G. Klyne, J.J. Carroll: *Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation.* `www.w3.org/TR/2004/REC-rdf-concepts-20040210/` (10 February 2004)

[14] F. Manola, E. Miller: *RDF Primer*. `www.w3.org/TR/rdf-primer/`

[15] C. S. Peirce, *Collected Papers.* Harvard University Press, Cambrigde, Massachusetts.

[16] C. S. Peirce, J. F. Sowa: *Existential Graphs: MS 514 by Charles Sanders Peirce with Commentary by John F. Sowa.* `http://www.jfsowa.com/peirce/ms514.htm`

[17] S. J. Shin: *The Iconic Logic of Peirce's Graphs.* Bradford Book, Massachusetts, 2002.

[18] J. F. Sowa: *Conceptual Structures: Information Processing in Mind and Machine.* Addison Wesley Publishing Company Reading, 1984.

[19] J. F. Sowa: *Conceptual Graphs Summary.* In: T. E. Nagle, J. A. Nagle, L. L. Gerholz, P. W. Eklund (Eds.): Conceptual Structures: current research and practice, Ellis Horwood, 1992, 3–51.

[20] J. F. Sowa: *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Brooks Cole Publishing Co., Pacific Grove, CA, 2000.