Studies on the Uses and Usefulness of Diagrams

Shai Berger

December 5, 2000

Chapter 1

Introduction

Cognitive Science as we know it today was built around linguistic behavior. It was the research on language which brought down Behaviorism in Cognitive Psychology and connected Psychology with Linguistics and Computer Science. But as the field evolved, increasing attention was paid to other forms of communication.

In the last two decades, we have seen a growing interest in graphics as subject of research of different paradigms. Graphics have been classified by designers, their cognitive effects were studied by psychologists, linguistic analysis has been applied to them, and logicians and computer scientists have been studying their structural and computational properties, all in an effort to further our understanding of their usefulness in presenting information to humans.

There are two main motivations behind this research. The more practical one is the idea that the principles underlying the usefulness of graphics in communication, when well understood, can be used to enhance the quality of presentations. The more theoretical drive stems from the intuition that information presented graphically is often very easy to consume — much more so than the same information presented verbally. Thus, understanding the principles of graphical communication may give us important hints to the workings of our cognitive system.

In a related, simultaneous development, the availability of graphical tools, following the recent advances in computer technology, gave rise to many uses of graphics as research and presentation aids, in Cognitive Science and science in general. This may bring forth a positive feedback loop: More use of graphics means more substance for research, whose insights may produce better, more focused presentation techniques, which can then be used for yet better research, and so on.

My primary motivation for this research is the theoretical one — I want to learn more about how people think. I believe that graphic representations of data help people solve problems, and I want to study how. This interest leads me towards the body of research in diagrammatic reasoning. Within this subfield, I focussed on two, seemingly remote, areas.

As a computer scientist by training, I look into the computational complexity of a typical problem encountered with diagrams in Chapter 3. The problem, which we all solve on an almost daily basis, is that of finding, extending, and verifying the correctness of diagram legends, which are mappings of diagram features to their meanings — real-world entities and properties. I first formalize this as a general computation problem. The general problem is NP-hard, but I attempt to show that for some diagrams it is tractable. The feasibility of solving this problem is an important factor in the ease of use of any representation, so showing it to stem from properties of the graphic medium goes some way towards explaining the usefulness of these diagrams.

As a curious person with an interest in Cognitive Science, I turn my gaze on a problem of psychological research in Chapter 4. The studies I look into are concerned with the effects of using diagrams in education, for teaching logic. I expose what I believe to be a methodological fault which made the researchers jump to some not-entirely-warranted conclusions. In an attempt to help remedy the problem, I suggest a new diagrammatic teaching system, and describe its principles and prototype.

To place this research in its context, I review some previous related work in the next chapter. I also use this review as an excuse to tackle some philosophical issues.

Acknowledgements

I would like to thank a lot of people for teaching me, in the roles of teachers, fellow students, or just company, but I cannot mention each and every one here... I want to thank Mehdi Dastani, for putting me on the track that lead to this thesis. I want to thank my supervisor and advisor, Michiel van Lambalgen, for his support and guidance, and for his criticism of my ideas. Michiel's PIONEER project also supported my trip to ESSLLI, where I presented a preliminary version of Chapter 4 at the Student Session. I would like to thank all my professors and TAs, the ILLC staff, and my other friends, for making this year an enriching and educating experience. I want to thank Johan van Benthem and his project Logic in Action for the scholarship which made this year possible financially; and last, but not least, I want to thank my wife Sigal, and both our families (in the words of Ian Anderson), for nothing in particular and everything in general.

Chapter 2

Related Work

Diagrams have been approached from different directions, usually utilizing tools and methods from other fields. I find most relevant the approaches relying on disciplines of more-or-less formal study of communication. Thus, I will mainly look into accounts which try to implement ideas from Linguistics, Logic, and Computational Complexity.

I should mention in brief other directions of investigation, less intimately related with the issues of this thesis. One is the research of perception as a formal problem: This tradition goes back at least to Marr's famed Vision [Mar82]. More recently published works include [vdDvL00], which attempts to formalize Marr's intuitions into a logic of direct perception reports, and [Das98], which uses an information-theoretic approach to account for Gestalt perception findings from psychology. In my own research, I pretty much take perception for granted, focusing on how perceived information is processed. Another direction, in some sense on the other end of my scope, is the study of graphic design. While some publications in this tradition (e.g. the seminal [Ber67] and [Twy79], and the more recent [Eng98]) classify diagrams by ways they express data, and are thus fairly close to my interests, I am not interested here in aesthetics or artistic values. One might say the verbal language analogue of the visual perception research is phonetics, and that of design is the study of literature and rhetoric. I set my sights on diagrams as a medium for informative (rather than affective) communication and reasoning, and so I will pay closer attention to syntax and semantics, inasmuch as these notions are applicable to graphics.

Now, let us follow some research on diagrams, first in terms of how they take significance, then in terms of how they help solve problems.

2.1 Analyses in Linguistic Terms

Verbal language has been studied extensively as a communication system. The research of diagrams as communication systems has taken a lot of ideas from these studies, at times modifying and extending them to deal with the peculiarities of graphical expression. Most works reviewed here deal with the connection between the internal structure of diagrams, arguably their syntax, and what they signify, their semantics. I also review an attempt to translate into the world of graphics the (by now) common linguistic wisdom that pragmatic principles may be used to explain apparent semantic anomalies.

2.1.1 Syntax and Semantics

There are two basic ways to look at the semantics of diagrams. The first is to keep the intuitions of sentential (natural and formal) language, which separate syntax and semantics. Then the diagram is seen as a sentence whose different parts are combined through some sort of grammar, a sentence in which every part — every feature of the diagram — has a meaning. Thus, the diagrammatic system is described by a language, and it is sentences in this language that are interpreted. Sometimes this interpretation is a translation into another language, and it is only the semantics of this other language which give referents to the objects and relations in the diagram. The second approach 'cuts the middleman', presenting the semantics as a direct mapping between two structures, with no intermediating language. The differences are subtle, but telling. In particular, every systematic description of diagrams should specify how to determine which features in the diagram are at all meaningful; where a logical language is involved, there may also be an issue of 'axiomatization': Choosing one set of formulas among many logically equivalent sets, which describe all the meaningful features. On the other hand, it turns out that the economy in analytical structure gained by giving up the mediating language has its costs in expressibility. Let us now examine these ways of looking at diagrams more formally and closely.

Formal Definitions

To formalize the language-mediated approach we may use the work of Wang [Wan95]. Wang defines concepts which are much more elaborate than we need here; for simplicity's sake, a cut-down, almost caricaturistic version of her ideas will suffice. Let a signature Σ consist of a set of constant and predicate symbols. Diagrams may be described in a *Picture Description Language* $\mathcal{L}$, which is a language using a signature $\Sigma(\mathcal{L})$ as its vocabulary, and accompanied by a theory $\mathcal{T}(\mathcal{L})$ characterizing the elements of Σ. Thus, $\Sigma(\mathcal{L})$ may include e.g. constants for referring to graphic elements, one place predicates for, say, different shapes, and a binary relation symbol for spatial inclusion. $\mathcal{T}(\mathcal{L})$ might then include the transitivity of inclusion, and a classification of shapes ("every rectangle is a quadrangle")[1]. A specific diagram is then (fully) described by a *situation*[2]:

Definition 2.1 ([Wan95, 4.0.3] simplified) *Let $\mathcal{L}$ be a picture description language, and $\Sigma \subseteq \Sigma(\mathcal{L})$. A* situation *over Σ is a set of literals over Σ, maximal consistent with respect to the graphical theory $\mathcal{T}(\mathcal{L})$.*

Maximal consistency means that every (meaningful) feature of the diagram is described in the situation, given a suitable graphical theory. Since maximal consistent sets of formulae are usually infinite, they do not make a plausible substance for computations. Hence, the set of formulae needs to be limited to a finite subset. Ideally, we should be able to generate the infinite set from our finite subset, so Wang's choice of subset — all the literals — makes sense. However, as we shall see, other choices can be made.

An interpretation is a translation from the picture description language to a language describing the application domain:

Definition 2.2 ([Wan95, 7.0.2] simplified) *Let $\mathcal{L}$ be a picture description language, and Σ a signature (for the application domain). An interpretation is a partial type-preserving function $\mathcal{I} : \Sigma(\mathcal{L}) \to \Sigma$, i.e. $\mathcal{I}$ transforms constants to constants and predicates to predicates of the same arity.*

The faithfulness of the interpretation can now be studied in terms of relations between a diagram and an application-domain theory . For example, Wang defines $\mathcal{I}$ to be *consistent with respect to a diagram and a theory* if, and only if, adding the interpretation of the situation encoding the diagram to the theory introduces no (new) inconsistency. As mentioned above, Wang's definitions are actually much more elaborate, and give more structure and flexibility to the semantics; I will give further details shortly.

For the direct mapping approach, we may adapt the standard definitions of model theory. The textbook formal definitions are for first-order structures, and they use linguistic constructs (a signature). For the time being, we may keep things first-order, but the language has to go:

[1]In Wang's definition, graphic elements have *sorts* which are partially ordered, and an external postulate says an object of any sort is also of any sort higher in the order, and thus this kind of classification is not needed in $\mathcal{T}(\mathcal{L})$.

[2]This has nothing to do with Situation Theory.

Definition 2.3 ([EF95, p.1] modified) *A* structure $\mathcal{A}$ *consists of a non-empty set A, the* universe *or* domain *of $\mathcal{A}$, and a set $\hat{A}$ of relations on A. Every relation in $\hat{A}$ is equipped with a natural number, its* arity*. An n-ary* relation *on A is a subset of A^n, the set of n-tuples of members of A.*

Now, a diagram is a structure — it is nothing more or less than a set of objects standing in different relations. It refers to another structure, from the application domain. For the definition of interpretation, let $f(A) = \{f(a) \mid a \in A \cap \text{dom}(f)\}$, and $f(\hat{A}) = \{f(R) \mid R \in \hat{A} \cap \text{dom}(f)\}$ (note that for $R \in \hat{A}$, $f(R)$ is set as part of f's definition and not implicitly by memberwise application).

Definition 2.4 *An* interpretation of $\mathcal{A}$ in $\mathcal{B}$ *is a partial injective function $f : \mathcal{A} \to \mathcal{B}$ preserving type, that is,*

- *$f(A) \subseteq B$,*
- *$f(\hat{A}) \subseteq \hat{B}$, and*
- *for any relation R in $\hat{A}$, $f(R)$ has the same arity as R.*

Faithfulness may now be discussed in terms of membership of tuples in relations being preserved by the mapping, that is: The interpretation f is faithful if and only if, for every relation R and tuple $\langle a_1, \ldots, a_n \rangle$ with R, a_i in $\text{dom}(f)$, it satisfies $\langle a_1, \ldots, a_n \rangle \in R \Leftrightarrow \langle f(a_1), \ldots, f(a_n) \rangle \in f(R)$.

Language-Mediated Approaches

First of all, a fuller description of Dejuan Wang's work [Wan95] is due. Wang takes the language-mediated approach with a formalistic attitude. She analyzes diagrammatic systems whose syntax and semantics are given, and presents a framework for specifying them precisely. In her framework, diagrams within a system are described by a Picture Definition Language (PDL for short. Wang refers to diagrams as "pictures" even in her thesis' title; I find this a misnomer). The vocabulary of this language is an order-sorted signature, i.e. it has a set S of sorts and a partial ordering $\leq$ of these sorts, which represent the classes of graphical objects and the subclass relation between these classes, respectively: $s_1 \leq s_2$ means that all (objects of type) s_1 are (also of type) s_2. The ordered sorts are used to define the domains and ranges of functions and the domains of predicates.

Terms in the language are constructed from variables and function symbols: Nullary function (or, constant) symbols designate basic graphic elements, and n-ary function symbols with $n > 0$ designate operations that can produce emergent objects. Thus, if a and b are both circles, $overlap(a, b)$ is the emergent object defined by their overlapping parts. An n-ary function symbol f has a *principal* type $s_1 \times \cdots \times s_n \to s$, where s, s_i are sorts; f is also of type $s'_1 \times \cdots \times s'_n \to s'$ if $s'_i \leq s_i$ and $s \leq s'$. In the above example, supposing types $Circle$, $Closed\_curve$ and Any with $Circle \leq Closed\_curve \leq Any$, the emergent-object generating function *overlap* could have principal type $Closed\_curve \times Closed\_curve \to Closed\_curve$, and then it would also be of type $Circle \times Circle \to Any$.

Atomic formulas in the PDL are constructed from terms and predicate symbols. Much like function symbols, n-ary predicate symbols have a principal domain $s_1 \times \cdots \times s_n$, and also have $s'_1 \times \cdots \times s'_n$ as their domain where $s'_i \leq s_i$. Formulas are built from atomic formulas using the regular connectives and quantifiers.

As this is a framework for the analysis and exact specification of systems whose significance is given, the "choice of meaningful features" problem is solved by decree on a system-by-system basis. A theory which describes the necessary consequences of geometric constraints (e.g., in a system where both containment and size relations are in the language, $a \subset b$ implies $a < b$) is part of the system; the "axiomatization" issue is resolved by translating a diagram into a set of atomic formulas closed under inference within this theory — the above-mentioned maximal consistent set of literals.

Notably, such a characterization defines exactly a class of diagrams which are equivalent up to meaningless features. Thus, this account is very close to the direct-mapping approach. However, it is the description, and not the diagram itself, which gets the semantics. An interpretation maps elements of the order-sorted graphical signature to elements of a language with an order-sorted application-domain signature: Sorts are mapped to sorts (via an order-preserving mapping); functions and predicates are mapped to *generalized* terms and formulas respectively. Generalized terms (formulas) are terms (formulas) with lambda-abstraction applied to them to produce functions (predicates). This allows abstractions to be expressed through the interpretation, e.g. the graphical property *orange* may be interpreted as $\lambda x(hot(x) \vee sweet(x))$ even if the symbol for hot-or-sweet does not exist in the application-domain vocabulary (provided *hot* and *sweet* do). Thus, Wang's system may be used to analyze *graphic abstractions* — schemes which allow one diagram to stand for more than one model (for more details see next section). Of course, such schemes cannot be handled in the direct-mapping approach in a straightforward manner.

Steven Pinker [Pin90] takes an attitude driven mostly by psychological evidence of the way people process diagrams, with and without legends. According to his account, the visual sensory data, which he refers to as the *visual array*, gets translated into a propositional *visual description* which partially describes the visual array. To explain which true propositions can make it into the description and which can't, he invokes principles grounded in basic perceptual research:

- The "indispensability" of space, which consists in space's importance in defining elements, the fact that most higher-order and combined predicates are based on spatial dimensions, spatial dimensions' higher discriminability, and space's major role in selective attention;
- The Gestalt laws of perceptual organization;
- Known effects in representations of magnitude, such as limited resources for verbal labels which leads to indiscriminability, differences between ratio values and absolute values, and salience of relatively extreme values (i.e. values that are extreme with respect to a set of objects in a scene); and
- The use of many interdependent polar coordinate systems for the description of objects in a scene.

These principles are claimed to constrain and essentially define the language of visual descriptions. No formal definition is given for the principles, nor an explicit derivation of the language of visual descriptions (though several proposals for such languages are referred to). Pinker further discusses processing issues which constrain which of the possible propositions actually do make it into the description:

- Processing capacity, or short time memory size, which limits e.g. the number of elements which may be considered together. This limitation may be alleviated by "rehearsing" the image, that is, attending it repeatedly; and
- Automaticity and default encoding likelihoods of features dictate (probabilistically) which features will be automatically encoded by bottom-up processes, and thus be "just noticed". The encoding of further features may be triggered by top-down processes.

These considerations give rise to distinctions between effortless, automatic encodings and deliberately enhanced ones. Pinker refers to these as the *reduced* and *complete* visual descriptions, respectively, in the context of capacity, and the *default* and *elaborate* descriptions with respect to automaticity.

The constraints imposed by these principles are Pinker's way to resolve the meaningful (in his terminology, "encoded") features and axiomatization issues. The diagram is parsed into scenes and sub-scenes, down to single elements. These parts stand in relations to each other, including the "part-of" relation.

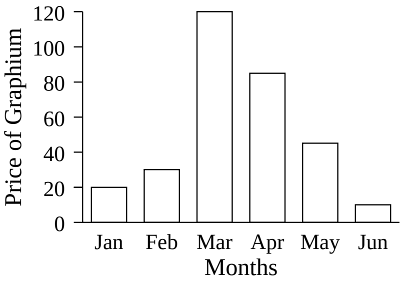
The semantics attaches *conceptual message entries*, which are propositions, to some parts of these scenes. A conceptual message entry is[3] a set of equations defining a dependency between different variables. There is a catch here — the variables, as well as the values assigned them in equations, may be from the language of either visual description or application domain, and the link between one and the other is established by a different kind of conceptual message entry which equates a visual variable with an application-domain one. This logically inelegant theoretical move is taken to account for the empirical fact that people can report pretty much the same information after viewing a graph with or without a legend. The information in legend-type conceptual messages may be incorporated into the other messages, translating them part by part from visual description language to application-domain language.

Figure 2.1, based on [Pin90, pp. 90–93], shows a diagram alongside a set of conceptual message entries which could have been extracted from it. Pinker gives a breaking of this diagram to scenes and elements, separating the graph from the axes and their labels; each of the blocks is a part of the graph scene, and each label of an axis is a part of a sub-scene of the axes.

Then, these parts are matched to conceptual message entries. Entries 1–4 are regular entries, showing different types of factual data: The first distinguishes a value as exceptional, the second is a qualitative comparison, the third records a visible tendency along a range, and the fourth is a simple value (Pinker suggests also the type of ratio, but this does not seem to fit this kind of graph). Pinker suggests that intuitively, these message entries may be understood as if they were preceded by "When", e.g. "When V_1 is March, V_2 is high". Entries 5 and 6 are legend entries; they may be accommodated by translating e.g. all occurences of V_1 to months (Pinker's example disagrees with his description on the application of this process to values: I think we should assume, that the entries we see here have already undergone some such translation, e.g. from measures of horizontal distance to x-axis labels).

Conceptual questions are conceptual message entries with missing variable values (e.g: V_1 absolute-value = April, V_2 absolute-value = ?); these may be generated by high-level mechanisms outside the scope of Pinker's work, and then fed back down to be answered (that is, to have the missing values filled

[3]Since conceptual messages are not defined generally and formally, but rather given by example — an example which is strongly slanted towards graphs — the following includes some guesswork on my part.



1:	V_1 absolute-value = March,	V_2 level = high
2:	V_1 pair = March,February,	V_2 difference = higher
3:	V_1 range = March–June,	V_2 trend = decreasing
4:	V_1 absolute-value = January,	V_2 absolute-value = 20
5:	V_1 = months	
6:	V_2 = graphium price	

Figure 2.1: A graph and conceptual message entries for it.

in) via the visual description; if the visual description lacks the relevant information, it is than elaborated by re-examining the visual array.

The forms of conceptual message entries, as well as their matching with visual description parts, are done through *Schemata*. These are supposed to be small in number, and relatively easy to test for fitting a given diagram. They include, among other things (such as labels useful for identification and later search), *message flags*. These are conceptual message equations, with missing values which are filled in when the diagram is *recognized* — that is, found to fit the schema (they differ from conceptual questions in the context of their use and in being just single equations, so they express no dependencies between variables). Thus, Pinker's treatment of the meaningful features problematic is spread across the syntactic and semantic levels of his account, and the processes which solve it are even permeable to higher mental functions.

For Pinker, then, the syntax is set by perception, and the semantics, which is self-referent to some degree, is built on a description of the diagram, with the diagram itself being used for back-up, so to speak. From an information-processing point of view, this approach has the interesting feature that the encoding is not opaque: Higher functions need to be aware of the information's source, or at least, that such a source exists and may be consulted to improve the encoding.

I would like to comment that the motivation given for the awkward (from a logical perspective) formulation of the semantics — people's ability to report graphs equivalently whether or not a legend is given — is not at all compelling, if one takes seriously a direct-mapping approach. This approach, the way I see it, strictly separates the task of reasoning with a representation from the task of reporting — about the representation, or the results of reasoning. Underlying this is a view that reasoning can be carried out without the use of language: Structures may be represented by (internal) structures, and these may be manipulated without requiring names or syntax; a more formal account is given in Chapter 3. As an example of what this comes down to, I would expect to find that many calculations performed by humans are described much better as applications of analogic image-manipulation-like operators, than as steps of discrete symbolic computation.[4]

Thus, a direct-mapping approach assumes reports are generated from internal structures in a separate process. This process may use information other than the structures (e.g. legends) when it is available on one hand, and suppress some known information on the other hand. Pinker takes reports as directly reflecting the internal representations, implicitly assuming that the process which generates the reports is trivial — at least as far as content is concerned (On a side note, Van Der Does and Van Lambalgen [vdDvL00] use reports in much the same way to motivate their *Logic of Vision*, which is in fact a logic of vision reports).

Towards Direct Mapping

A step towards the direct mapping approach is taken by Corin Gurr [Gur98], who takes a closer look at the differences between sentential and graphical ways of expression. His terminology still includes the interpretation of syntax in diagrams — he has a section titled "Syntax and Semantics". In this section, he points out that in diagrams there is a direct mapping of syntactic relations to semantic ones. In another section, though, he examines the interpretation of a diagram as a homomorphism between structures, which are abstractions of the diagram and its intended signified (e.g. the interpretation of an electronic circuit diagram is a homomorphism between one structure, which is an abstraction of the circuit over physical appearances and locations of components, and another structure, which is an abstraction of the diagram over physical locations of elements: It is only the identities of elements and connections between them which matter). It should be clear by now that these two views of interpretation do not sit well together. Moreover, he compares the spatial relations between elements in diagram to concatenation — the only spatial relation in sentential languages — rather than to syntactic relations in such languages, or, better yet, to predicates, and comes up with the somewhat confusing argument that while there is syntax in diagrams, it is not abstract, and so interpreted directly.

Gurr's colleague at the HCRC, Keith Stenning [SteTA], makes one further step away from the analogy with verbal language, and questions the very idea of calling the relations between graphical elements "syntax", as they serve a totally different function. Stenning still uses the somewhat confusing notion

[4] This view is related to the famous Imagery Debate in cognitive psychology; for references, see e.g. the papers by Kosslyn et al. and by Pylyshin cited in [SteTA] and [Pin90].

of “abstract syntax”, but in his discussion he redefines what is diagrammatic representation, in a way which can be summed up as, it has no syntax. For this last statement to make sense, we need to clarify the terminology a little.

According to Stenning, a representation is diagrammatic inasmuch as its spatial (or temporal) relations are interpreted as predicates. He does not put it exactly this way, but by his account the sentential languages are those that give representation structure a uniform, albeit meta-linguistic interpretation — like the interpretation of the relation between a sentence, its verb and its subject in English. By “syntax” we mean here this interpretation of the structure of representations.

Stenning gives two surprising examples of his distinction. On one hand, the concatenation of symbols in a word of a regular language, which stands for a state shift in an automaton, is interpreted, making regular languages a diagrammatic representation system. On the other hand, he gives a semantic network diagram where the same arrows can mean either a relation or a logical constant, and claims this representation to be non-diagrammatic (“sentential” does not seem to be appropriate...) — the arrow in itself is not being interpreted as a predicate, but only indicates which objects are related.

To preempt possible misconception, it is not the claim that syntactic relations have no semantic meaning in sentential languages: In first order logic, for example, the syntactic relation that holds between two variable symbols under nested quantifiers (like x and y in $\forall x \exists y Rxy$), represents the semantic relation of dependence between the corresponding variables. But variables are not in the domain of the language — objects are, and the dependence between variables is not defined by a structure, but only by the formula (Dynamic Predicate Logic, which uses the same syntax as FOL with different interpretations for the quantifiers, is not a counter-example: It is indeed another language). In other words: The formula does not say that one variable depends on another, as this statement only makes sense in the meta-language. In diagrammatic representation systems, the spatial and other pseudo-syntactic relations are interpreted in the structure, and so these systems have no component which is interpreted meta-linguistically — hence, no syntax.

2.1.2 Semantics and Pragmatics

In the field of verbal language, there is a long tradition (at least since [Gri75]) of using pragmatic principles to explain how, in some contexts, expressions carry a non-standard meaning. This allows the semantic description to ignore these divergent uses, making it simpler. An attempt to apply the same strategy to the analysis of graphics was made by Marks and Reiter [MR90]. Their analysis was used to explain further findings in [Obe96], which notices the difference between conventional and conversational implicatures, but accepts Marks and Reiter's claim that the implicatures they describe are conversational. I think it is worth a little digression here to argue the other way; this necessitates the explanation of some background.

In his classic paper, Grice [Gri75] argued for a standard logical interpretation of natural-language connectives by invoking the notion of implicature — that sentences carry, besides their direct meaning, additional propositions whose truth-value is separate from that of the sentence. Grice differentiated two major classes of implicatures, *conventional* and *conversational*. He exemplified the former by the sentence "He is an Englishman; he is, therefore, brave". According to Grice, this sentence asserts that its subject is an Englishman and a brave man, but merely implicates the causal connection between these two facts: If the connection was found not to hold, the sentence may still be considered true. Conventional implicatures stem from the conventional meanings of expressions — in this case, "therefore". Conversational implicatures are a little more subtle, and spring from the general principle governing (informative) discourse, which is the Cooperative Principle. The assumption that all participants in a conversation are cooperative implies certain rules, or maxims, they may be expected to follow. Following Kant, Grice specified the maxims of Quality (be truthful), Quantity (be as informative as required, but not more), Relation (be relevant) and Manner (avoid ambiguity and obscurity, be brief and orderly). When an utterance violates these maxims at the surface level — that is, at the level of what is conveyed by the conventional meanings of the expressions used — this triggers the inference of additional propositions which would make the utterance cooperative. Thus, when A asks B how their mutual acquaintance C is doing in his new job at a bank, B's answer "His colleagues are friendly and he hasn't been to jail yet" may be understood to implicate that C is likely to succumb to temptations encountered on the job, for otherwise the mention of jail would be irrelevant. After discussing several such examples of context-dependent implicatures, Grice examines so-called generalized conversational implicatures: expressions which, barring contextual evidence to the contrary, when used in an utterance, carry implicatures. Grice admits that most such cases are better handled as conventional implicatures: To establish a non-controversial example for the viability of the notion, he uses an expression ("a") which generates different implicatures in different sentences ("I am walking with a woman" implicates that the woman is not closely related to me, whereas "I broke a finger" would be surprising if it referred to someone else's finger).

Marks and Reiter [MR90], and Oberlander [Obe96] following them, claim that the implicatures generated by ill-designed diagrams are conversational. Marks and Reiter just say they follow from the Cooperative Principle, and show some examples where misleading diagrams may be construed as violating conversational maxims; they do not seem to consider the notion of conventional implicature applicable. Oberlander's argument is that since the meaning of graphical expressions is set by the diagram's legend, there are no conventional meanings for these expressions; hence, the implicatures cannot be conventional. Instead, he suggests elaborating the sub-maxim of Order (under the maxim of Manner), modeling it after the maxim of Quantity: Be as orderly as needed, and not more. With this new maxim, a lot of unintuitive design features — an element out of alignment in a set referring to equivalent entities, wrong grouping of elements, gratuitous use of a different pen-width in drawing an element, using different fonts for different titles, and even using element size, rather than shape, to designate entity type — can all be explained as too orderly or not orderly enough.

In my opinion, this account stretches the notion of Order far beyond its domain. Order, in the sense Grice uses it, is not synonymous with regularity or repetition: A discourse where the same word is used for everything is not inherently more orderly than normal discourse. An orderly presentation is one whose form matches its content: History sorted by time, logical proof sorted from premises to conclusions, physical process described from causes to effects. In this sense, one cannot be too orderly.

Further, I doubt the assumption that there are no conventions governing the use of graphical language elements, other than those specified by the legend. On the contrary, a lot of evidence to the effect that there are such conventions is reported by [Tve97]: From the obvious and general, like that physical distance is inversely proportional to strength of relation, through the insightful, like that up is more or better, and to the particular and surprising, like that time flows in the direction of writing. In particular, this includes the use of the right kind of visual variables to designate domain variables, e.g. size for ordinal data, which goes back at least to Bertin [Ber67].

Principles of this kind can explain all but one of the implicatures discussed by [Obe96] — all of the unintuitive design features listed above, which Oberlander borrows from Marks & Reiter, violate conventions of graphic depictions. Oberlander then discusses the diagram in Figure 2.2, taken from Petre & Green[5]. This diagram is a schematic drawing of an electronic circuit, made by a novice engineer, and Petre & Green find four faults in it, of three

[5]Petre, M. and Green, T.R.G. (1992) Requirements of graphical notations for professional users: electronics CAD systems as a case study. Le Travail Humain, 55(47–70). Cited by [Obe96].

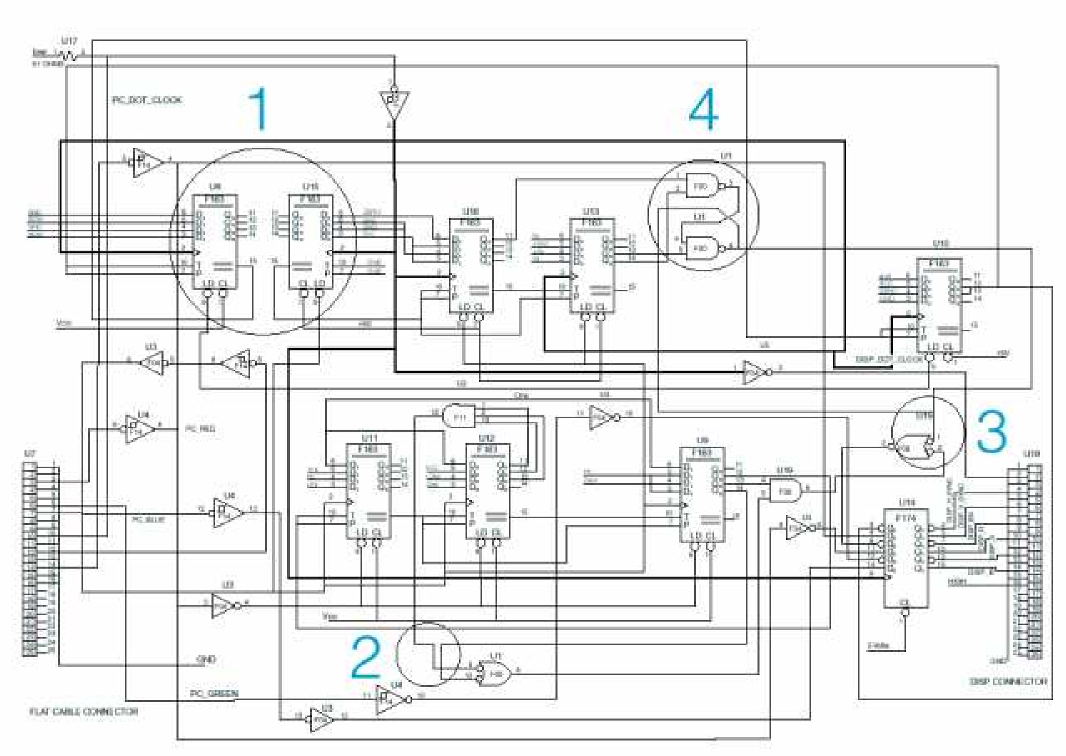


Figure 2.2: Circuit diagram designed by a novice

types:

- The fault marked 1 is the adjoining of unrelated chips, which Oberlander construes as violating of the Order sub-maxim via the physical-closeness-implies-relation convention: Rather than seeing this convention as the direct cause of the implicature, he says an implicature is generated because it is unorderly not to follow the convention. I think the invocation of order here is redundant, and the implicature is better seen as conventional.

- The faults marked 2 and 3 are instances of unnecessarily convoluted or odd wiring, which Oberlander again presents as violating the sub-maxim of Order. However, he doesn't point any concrete implicatures generated by these violations (except a vague notion that there should be a reason for the weird wiring). Indeed, the wirings are hard to follow, but not misleading — there are no obvious implicatures.

- In the circle marked 4, two gates and the wires connected to them were

> drawn in a way which resembles a flip-flop — a well-known component, and an idiom in the diagrammatic language — when in fact they were not a flip-flop. Oberlander says this is ambiguous, bordering on false, and thereby violating the maxims of Manner and possibly Quality. However, one should notice that this violation is a result, rather than a cause, of the problem: It is not the violation which generates the implicature, but (if this is indeed an implicature) the other way around. Oberlander says this case is particularly interesting, and I agree; for a discussion which, in my opinion, handles the real issues it raises, see Davidson's *A Nice Derangement of Epitaphs* [Dav86].

There is another, methodological issue here: A lot of the explanational power of Gricean maxims comes through the analysis of the cases where they are clearly violated on purpose, to communicate something different from the surface content of what is said. Grice [Gri75] calls this "flouting" a maxim. The most famous example is the recommendation letter for a candidate for a research position which praises his social skills and says nothing about his research capabilities. The letter (or, rather, its author) flouts the maxim of relevance, thereby conveying his negative opinion on the person's abilities without spelling them out. Another example is a sentence like "Yeah, I can finish this assignment in two days, and my PhD in two weeks". Here the maxim of Quality (truth) is flouted, and the addressee is expected to understand the negation of what is actually said. A similar but different example is, when we are told that Alice saw a White Rabbit run by her, and

> "There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, 'Oh dear! Oh dear! I shall be late!' "[Car94]

Here again the Quality maxim is flouted, this time to convey meta-discursive information: The discourse is on its way to Wonderland.

Marks & Reiter and Oberlander only examine unwanted implicatures, setting their sights on eliminating them; I know of no diagram which puts this efficient communicative device — the conversational implicature — to good use, or as Grice puts it, "exploits" maxims, to convey *factual* information. It would take a lot more research to establish this, but my intuition is that pictures and diagrams can only flout maxims to implicate meta-discursive propositions (graphics which use this device are usually referred to as caricatures). A diagram is arguably a sentence, and not a discourse; as such, it would not on its own be able to flout maxims other than that of Quality. Of course, the presentation of a diagram in a discourse can flout maxims, but then we would not say that it is the diagram which flouts the maxim, but rather the discourse. Yet, if my thesis is true, it is very interesting that the dividing line of implicatures does not pass between the maxims, but within the Quality maxim. This may suggest a direction for further study of conversational implicatures, focusing on what is implicated, to refine Grice's classification of implicatures by how they are generated.

One should remember that, as in the field of verbal language, there are pragmatic issues which have nothing to do with implicatures: There is pragmatics beyond Gricean maxims, and some of it shows up in the analysis of graphics, too. For example, Twyman [Twy79] classifies diagrams, among other factors, by *directedness* of viewing them, i.e., whether the viewer's gaze is led along some path or not. This is an issue of the use of diagrams as means of communication, which has no direct semantical consequences.

2.2 Analyses of Expressivity and Complexity

Several authors have tried to expose the special traits of diagrams as the more-or-less formally specifiable properties they have as a representation system. Usefulness here is seen mostly from a computational point of view: A representation is useful if it reduces the complexity of solving some problem enough to justify the overhead of translations (of the problem instance, forth, and solution, back), and if it does not interfere in finding a correct solution. For this approach to hold any ground, diagrammatic systems have to be seen as potentially valid tools for solving problems, in spite of the tradition in mathematics to view them as no more than illustrations. Jon Barwise was probably the most important proponent of this idea.

A good example of Barwise's approach can be found in [BS95], although this paper does not specifically address diagrams as such. The paper borrows the logical machinery of Barwise's Situation Theory to define formally what it means for some system to represent another, and then goes on to investigate two phenomena that affect the usefulness of representations: Free Rides and Overdetermined Alternatives. The former are cases where the afore-mentioned translations explicate some information needed to solve the problem — in extreme cases, as soon as the problem instance is represented, the solution may be read off. A classical example of this kind is using Euler's Circles for the syllogism. As soon as we draw "All As are Bs" and "All Bs are Cs" as in Figure 2.3, we may read from the diagram that all As are Cs.

Overdetermined Alternatives occur when some information cannot be expressed in the representation without expressing some information that does not follow from it. A common example in diagrammatic reasoning is disjunctive information: In Figure 2.3, there is no way for us to replace "All As are

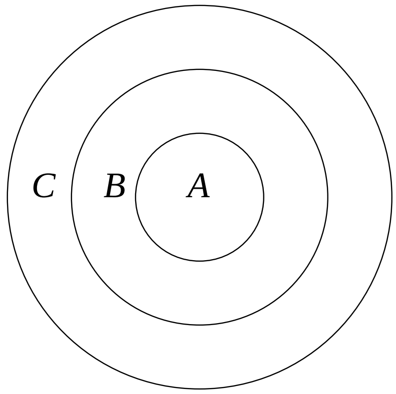


Figure 2.3: Euler's Circles: All As are Bs, All Bs are Cs

Bs" with "Either all As are Bs, or no A is C". Thus, we cannot use Euler's Circles to infer that $(C \setminus B) \cap A = \emptyset$.

One of the earliest accounts dedicated specifically to diagrammatic representations is [LS87]. In this paper, Larkin and Simon claim that diagrams help solving problems, by facilitating more efficient search and identification procedures than sentential representations do. The claim is that diagrams can group together the problem elements which are used together in inference steps, thereby aiding search for such elements; and that they typically represent all information pertaining to one element in one place, making it easier to identify the needed elements (Larkin & Simon present reasoning as the application of inference rules in a production system). Thus, for them, diagrams are essentially a way of presenting indexed information. This they contrast with verbal descriptions, which they treat as a sequence of propositions in random order and no indexing. In addition, though with much less analysis, they claim that diagrams "automatically support a large number of perceptual inferences, which are extremely easy for humans"(p. 98); I think it fair to assume part of these inferences are the Free Rides of [BS95].

A more sophisticated approach is taken by Stenning and Oberlander [SO95], who classify representation systems according to their capability for abstraction. A representation system includes representations and models. Stenning and Oberlander distinguish three classes of such systems: *Minimal abstraction representation systems*, or MARSs, are those where every representation has exactly one model. An example is a propositional language where every well-formed formula must specify the values of all propositional variables. *Limited abstraction representation systems*, or LARSs, are those which allow representations to have more than one model, but place limitations on such abstraction. An example is a system allowing partial information, where well-formed formulae are conjunctions of literals, but not all variables need to be specified (the limitation is that one may only represent certain sets of models). And then, of course, there are *unlimited abstraction representation systems* (UARSs), such as natural language. The authors mention *specificity*, which is the property of representation systems which require some information to be specified by any valid representation (a way of looking at limited expressivity). MARSs are of very high specificity, which means they are very easily queried, but not very expressive. UARSs are very expressive, but may be undecidable. LARSs trade expressivity for tractability, and thus are more useful than either extremes, under the right circumstances. Stenning and Oberlander suggest that graphical representation systems are almost always LARSs, and derive their usefulness from this trade-off.

In the monograph mentioned above, Stenning [SteTA] adds to these considerations one more distinction, which separates sentential representation systems from diagrammatic ones: Token-referentiality vs. type-referentiality.[6] In token-referential systems, identity of reference is expressed by identity of referring token (symbol occurrence), while in type-referential systems it is expressed by identity of type of symbol. For example, maps are token-referential systems: In a map of the Netherlands there will only be one token representing Utrecht, and from this one token and its relations to others we may see that Utrecht is south-east of Amsterdam, and also that Utrecht is west of Enschede. But in the representation of the same facts in the last sentence, we have used two tokens of the same type — 'Utrecht' — which refer to the same object; language is type-referential. A direct consequence of token-referentiality is that a referent can only be picked by one symbol; prominent consequences of this, such as the concentration of information related to one element, and the ease of search brought forth by it, were already noted by Larkin & Simon [LS87] as mentioned above.

Stenning draws a stronger, intuitively appealing, but in my opinion, unjustified, conclusion: "If a referent can only be picked out by one symbol on a diagram, then it cannot be picked out by two logically unrelatable ways. The complexity of reasoning in type-referential systems stem from their ability to pick out the same referent in alternative ways which the consumer cannot tell to be equivalent". The point this argument misses is intensionality: To pick a referent in logically unrelated ways, one must use different Fregean senses, that is, to pick the same extension through different intensions. Picking intensions in logically unrelated ways is much harder; it requires functions from

[6] Stenning attributes this distinction to John Etchemendy.

intensions to intensions, or expressions whose extensions are intensions, or other sophisticated devices. While natural language does have these devices, the issue of their presence is orthogonal to that of token- or type-referentiality.

On one hand, then, even in a type-referential system, when the same intension is used via the same symbol type, the reasoner can easily notice this. On the other hand, even in a token-referential system, the same extension may be picked out through different intensions: For example, in Figure 2.3, it may turn out that the circles B and C actually refer to the same set, via different properties. B could stand for $\{n \in \mathbb{N} \mid n > 2 \,\&\, \neg\exists xyz(x^n + y^n = z^n)\}$, and C for $\{n \in \mathbb{N} \mid n > 2\}$. Presuming A is also given an adequate referent, say $\{17\}$, Figure 2.3 is a perfect description of what was known about these sets in 1990, and the interpretation is still valid today. For a more intensional example, one does not need a type-referential system to talk separately about equal-sided rectangles and right-angled diamonds — combining intensions to create new intensions can be done in a representation system whether or not it is type-referential, and no repetition of tokens was needed here to produce two separate, *intensional* references to squares[7].

One must take care, then, to specify exactly which kind of referent is meant in this discussion — intensional or extensional. And one should also qualify Larkin and Simon's arguments: They apply usually, not always; and they apply much more strongly to basic elements (in the sense of Wang — e.g. circles in an Euler diagram), and less strongly to emergent ones (e.g. intersections of circles).

Corin Gurr [Gur98] adds two important insights on the efficacy of reasoning with diagrammatic representation systems. The first is that these systems tend (in comparison with sentential systems) to shift the weight of reasoning, from the manipulation and representation of premises, towards the reading off of conclusions: Whereas a typical reasoning process in a language produces only the required results, diagrammatic reasoning usually generates a diagram from which many conclusions may be drawn; the required result may not be all that easy to distinguish. Thus, he suggests to rename the afore-mentioned "Free Rides" to "Cheap Rides", noting that some rides are cheaper than others. As an example for this, he compares the ease in deducing "All As are Cs" from "All As are Bs and all Bs are Cs" using Euler's Circles (Figure 2.3) with the slight difficulty in making the same inference with Venn diagrams (as seen in Figure 2.4).

Gurr's other important insight has more to do with the direct mapping model of significance. He claims that for this model to work the legend has to be a homomorphism, preferably an isomorphism, between the represented

[7]This makes it three references.

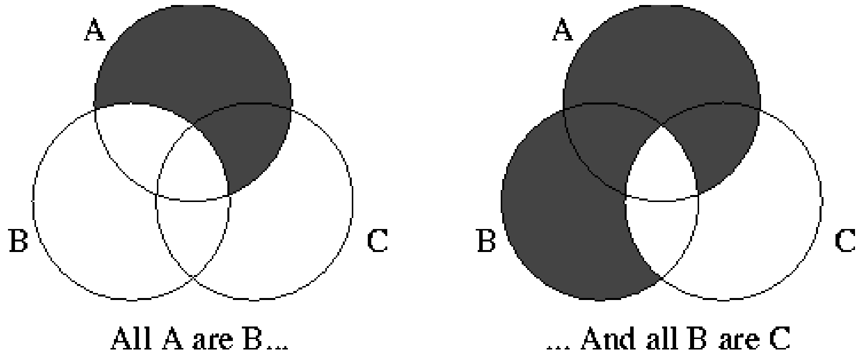


Figure 2.4: Venn diagrams for the same syllogism as Euler's Circles above

world and the representing world. This is hardly news, but Gurr also sees the importance of meta-systemic properties: For a reasoner to make use of the isomorphism, she needs to know it is there. Hence, it is easier to reason with systems whose legend is not just any isomorphism, but one which may be easily recognized. This insight provides much of the drive behind Chapter 3 of this thesis.

Chapter 3

The Complexity of Interpretation

As mentioned above, Barwise and Shimojima [BS95] contend that every computation involving representation[1] includes three phases: Representing the problem, manipulating the representation, and reading off the results. This analysis fits well within a scenario where a reasoner is given a problem, and wishes to use a certain representation system to solve it. However, if one wishes to focus on comprehension, then the scenario is different: The representation — the diagram — is given, but since in this scenario the user-reasoner is not initially (fully) aware of the diagram producer's intentions in using graphical elements, she has here the additional task of understanding the diagram's language. This is what I would like to call interpretation (after Davidson's *Radical Interpretation*).

3.1 Introduction

Interpretation, then, is the problem of matching graphical language terms with real-world entities and relations, or figuring out the legend. It should also include figuring out what are and aren't meaningful graphical terms, and what are terms at all in the diagram. In other words, this includes the identification of the graphical language as well as deciphering its semantics. An example of this problem is the Indo-European Languages chart shown in Figure 3.1, which appears in Webster's dictionary with no additional legend (except for a note that parenthesized languages are obsolete). As is, it is a badly designed diagram: What is the meaning of the different fonts? How

[1] They refer to "external" representation, outside of a reasoning organism, and call the computation "surrogate reasoning".

are the Medieval and Ancient Greek different from all others, to warrant not giving them a box? What does box width mean? What does it mean for two languages to be linked? Well, it is quite clear that linking stands for the derivation of a language from another, the parent language being the one closer to the center. But the important issue for this chapter is that we know, quite easily, which of the above we could answer and which we couldn't. This means that the search for an interpretation, whether successful or not, is efficient.

As hinted above, I subscribe (methodically) to a bottom-up modular view of cognition, where perception — the process which takes a (in this case, visual) stimulus and produces some internal representation for it — is seen as a one-time, cognitively impenetrable process, and the higher mental faculties work exclusively with its output. Accordingly, I see interpretation as a two-phase computation. In the first phase, a stimulus (the diagram) is analyzed to yield a structure of elements — a graphical 'sentence' made of graphical parts-of-speech. I would like to relegate this phase to perceptual and Gestalt psychology. In the second phase, this 'sentence' is understood to refer to some other structure. The latter structure is supposed to be (at least partially) familiar to the consumer of the diagram. The computational problem is, then, to match a given graphical structure to some known structure. Part of the matching is usually given, but part of it still has to be found.

In order to investigate this computational problem generally and effectively, we now need to define formally what we mean by the three key terms mentioned above: A graphical structure, a known structure to match against it, and the matching itself. As for the first two, we will proceed shortly to use a generalization of Definition 2.3. Matching, then, will naturally be phrased as finding a good mapping between the two structures; we will then deal with formalizing what is a 'good' mapping, and start an investigation into the complexity of the problem.

Before delving into technical details, I want to make some methodological notes. First and obviously, this is a direct mapping approach, and as such, in its present form, it has little applicability to systems which use graphical abstractions. If, as I hope, the direct mapping approach can be extended to handle such abstractions, this would warrant a similar elaboration of this part also. Second, the analysis hereafter is orthogonal to questions of intensionality, that is, the known structure may in principle comprise intensions, extensions, or both, as long as each of its parts finds its graphical counterpart. Thus, I do not expect to find a gain in computability (equivalently, a loss in expressivity) as a feature of the way a diagram takes significance, but rather as a feature of the graphical structures themselves. Last, diagrammatic communication is not limited to the presentation of diagrams; sometimes

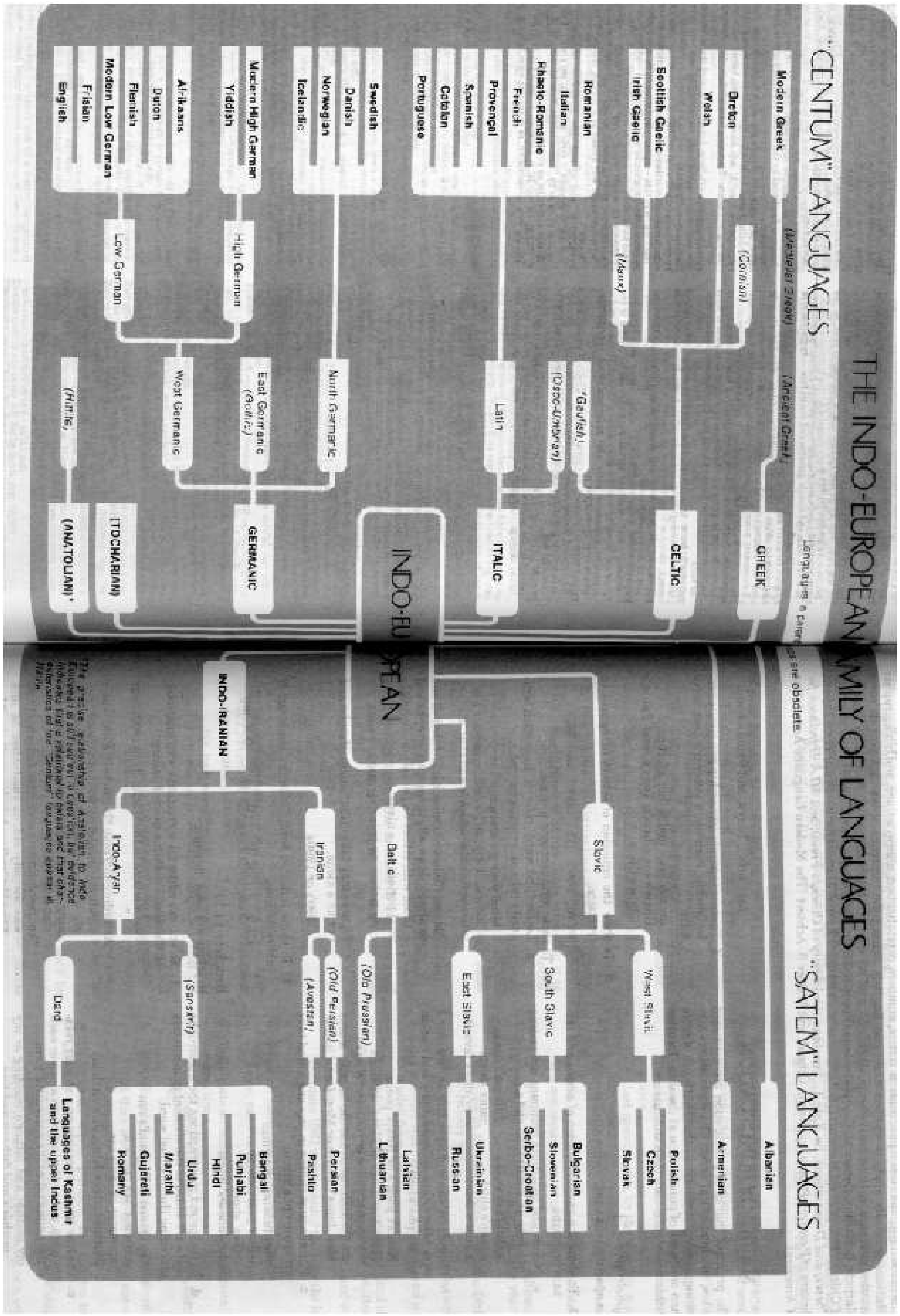


Figure 3.1: The Indo-European Languages [Neu93, pp. 1558–9]

these diagrams are modified. In some contexts these modifications are well-regulated, and are limited in ways which ensure that the resulting diagram is still sound, or, in the terminology just introduced, that the matching is still good (cf. [Gur98]). In other contexts, this might not be the case, so the matching needs to be verified (if not computed) again. This implies that the problems whose complexity is examined here may be encountered in other cases, and are not restricted to diagrams with missing legends.

3.2 Graphical Structures

A vocabulary is a set of symbols of different types — i.e. constants and predicate symbols of different orders and arities. A structure over a vocabulary is a set of objects of different orders (i.e. atoms, sets, etc.) which assigns an atom to each constant, a set of tuples to each predicate etc. and thus a truth value to every atomic formula.

As a first approximation, graphical structures are simply structures over vocabularies which include graphical relations — e.g. *red*, *blue*, *left-of*, $<$ (physical size order), and $\subset$ (spatial containment). Such vocabularies, or languages, have been suggested by many: For example, Wang [Wan95] gives several such vocabularies in examples, each fitting a specific diagrammatic system, and Pinker [Pin90] lists many examples of predicates which belong in such vocabularies.[2] In most cases these vocabularies only have first-order predicate symbols[3]. While these are sufficient to capture the perceptible properties of any diagram, it is my intuition that they are not sufficient to capture some diagrams' Gestalt-induced structure. For example, in the following 'diagram', grouping is induced by physical distance between elements. But while 'closer together' is a first-order relation (on quartets of elements), the element P is physically closer to A than to D, in contrast with the grouping.

GXQA PYJBD

While one could express the grouping as a first-order equivalence relation, this is counter-intuitive: Compare "G, X and Q are in the same group" with "G is in the same group as X and X is in the same group as Q" (where the same-groupness of G and Q is to be inferred by transitivity and the

[2] Pinker does not give a systematic description of a comprehensive vocabulary; the interested reader may do well to check his references.

[3] Wang's vocabularies are first-order; Pinker, when listing predicates in the text, only uses first-order predicates, but in his examples he uses second-order objects such as 'scene' and predicates such as 'part-of'.

converse relations by symmetry). In a similar fashion, the general form of symmetry, which may apply to a family of sets of elements, is intuitively a third-order predicate. Thus, graphical structures should be at least second-order structures.

The keen reader might be wondering, at this point, why are language elements like vocabularies creeping into an account which supposedly adheres to the direct mapping approach. The answer is that a vocabulary is a way for us to refer to the structure, but the entities involved are the structures and their elements, not the linguistic objects which describe them; the role of the vocabulary here is only to give an inventory of objects and relations of different types (orders and arities). In other words, the vocabulary is part of the meta-language, not the object language, of this discussion; the latter has only unnamed relations.

3.3 Knowledge Structure

Rivers of ink have been spilt over attempts to formalize the structure of human knowledge and memory, taking into account a myriad of phenomena. In this context, all we need is a formalization of the structure of knowledge in a way that will be compatible with the graphical structures. We should, however, pay heed to the fact that if we present human knowledge as a formal structure like those defined above, it should have a very high order — in fact, because of its self-referentiality (the language describing this structure is part of it: The word 'applies' applies to the predicate *applies*), it must either be of infinite order or lack orders altogether. For compatibility with the common descriptions of graphics, I pick the first. Also, it is unclear what are the 'fundamental' atomic objects: In the context of their genealogy (as in the above mentioned languages chart), languages may well be considered atoms. But when words are considered, languages are third-order objects (sets of sentences, which are sequences of words); and words themselves are sometimes seen as sequences of letters. Thus, it only makes sense to speak about relative order in this structure; predicates are of higher order than their arguments, and the set of atomic objects is context-dependent.

Another point to discuss is what part of the consumer's knowledge is actually involved in interpreting the diagram. I would like to take an approach which makes minimal assumptions on the organization of knowledge, which means the reply is: Everything. For a process that takes a large structure as input to be possibly efficient, the problem at hand should be polynomial-time decidable. Another approach which seems plausible is based on the compartments described e.g. in [Was00]. According to this approach, a small substructure is selected by cues in the stimulus, and matching is only done against this substructure. The substructure may in fact be small enough to make exponential algorithms feasible, and thus may provide an alternative explanation to the efficiency of the process.

3.4 Formal Definition of Structures

In Chapter 2, we gave a definition of a first-order structure. In accordance with the discussion above, we want now to get rid of limitations of order from above (only sets) and below (basic elements). That is, our structure does have basic elements, but we would like to leave formally open the possibility of extending it by breaking those up into smaller objects. Hence, we have a set of objects of different orders, with those of the lowest order being atoms, and those of higher orders being relations on (or, sets of tuples of) objects of lower orders. Formally we define:

Definition 3.1 *A* generalized structure $\mathcal{S}$ *consists of a non-empty sequence* $\{A_i\}_{i \in I}$ *of non-empty sets, for some finite* $I \subset \mathbb{Z}$*. Each member* a_i *of* A_i *has an arity* $n(a_i) \in \mathbb{N}$ *and satisfies:*

- $i = \min(I) \Leftrightarrow n(a_i) = 0$*. In this case we say* a_i *is an* atom.
- *For* $i > \min(I)$*, by the above* $n(a_i) > 0$*, and further* $a_i \subseteq (\bigcup_{j<i} A_j)^{n(a_i)}$*, that is,* a_i *is a set of* $n(a_i)$*-tuples of lower-order objects.*

We will abuse notation by writing $\mathcal{S}$ *for* $\bigcup_{i \in I} A_i$ *where this will cause no confusion.*

A generalized structure can thus be straightforwardly extended to higher orders or more basic elements as long as the set of orders is finite. In this we have undermined the above definition of graphical structures: The term ‘graphical structures’ is now to be understood intuitively, as referring to generalized structures that originate from the perception of graphics.

The generalized structures are a lot like the *general models*, introduced by Henkin, and described e.g. in [vBD83]. The main technical difference is that Henkin’s models use a more elaborate type-system, where the type of every predicate is based on the full type of each of its arguments; in the generalized structures, it is only based on the arguments’ number and orders. For example, general models have separate types for unary predicates over binary predicates over atoms (like “transitive”), and another one for unary predicates over unary predicates over atoms (such as “is a color”). While these examples show its advantages, the stricter type system has its limitations, too: For example, it would require four different predicates for the same binary spatial relation, if its arguments are to be both elements and sets (compare in the next example “C1 is left of O1”, “C1 is left of the O group”).

3.5 Structure of a Diagram — An Example

To make the above discussion a little more concrete, let us analyze the structure of a simple diagram. Consider Figure 3.2.

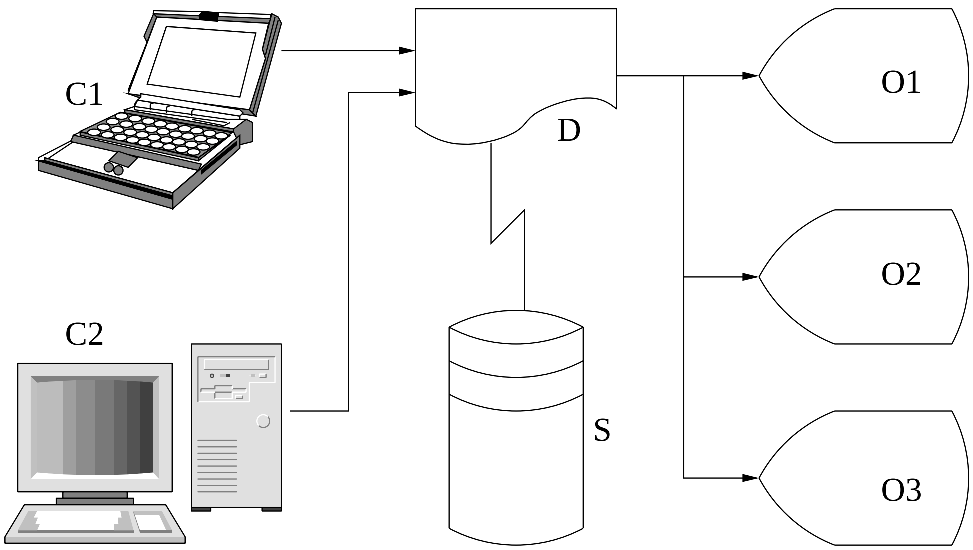


Figure 3.2: A simple diagram. Any similarity to real systems or people, dead or alive, is coincidental and unintended.

This figure is made of seven objects (marked C1, C2, D, S, O1, O2, and O3) in a certain spatial arrangement with links between them. The objects may be perceived as members of groups: O1, O2 and O3 form one such group (let us call this group O), and C1 and C2 form another (which we will call C). One may say D and S form a third one, a little less well-distinguished than the first two; in the analysis hereafter, we will ignore this grouping. Assuming the seven objects above are atoms, or objects of order 0, the group-objects O and C are of order 1. Both groups are horizontally aligned; thus we have a second-order predicate, Horizontally-Aligned, which applies to both groups (equivalently, an object Horizontally-Aligned of order 2, which is {C,O}). The group O is also Equidistant (another second-order predicate). C1, D and O1 have their tops vertically aligned, but they do not form a group, so we might prefer to encode this fact as the 1st-order relation SameHeightTop holding between each of the pairs. Similarly, we might prefer to encode the vertical alignment of the bottoms C2, S and O3 by the SameHeightBottom relation. Likewise, the spatial relations between D and S do not generate more elaborate relations. Of course, we have five different object types, each giving rise to a one-place first-order predicate, e.g. Laptop={C1}.

Some analyses (e.g. [SteTA]) treat links as relations between objects. One may also treat them as objects in their own right, as they may have properties other than the identities of connected elements. Here, we can take the former approach. Certainly, we have a link type different from all others between D and S; we might encode this as LightningLinked(D,S). Then, there is the same link type between C1 and D, and between C2 and D: LinkFromTo={(C1,D),(C2,D)}. Then, there is a question of how we see the connections between D and the elements of O, since they share part of the linking line. One may see this as just shorthand, and treat it as three more instances of LinkFromTo. Another alternative is to view this as one connection between D and O — For this to be the case, LinkFromTo would have to be second-order.

These considerations are perhaps not conclusive, but I believe they give a good sense of the issues involved in a formal presentation of the structure of diagrams.

3.6 Matching (Generalized) Structures

Having defined exactly the structures involved, we are now left with the question of defining formally what it means to match them. As sketched above, this boils down to finding a good mapping between the structures. But we still have to define, what makes a mapping good? How do we formalize the notion of a mapping that embodies comprehension?

One of the first answers to come to mind is an isomorphism. While intuitively we would like to say that an isomorphism is always a good mapping, we would like to accept other mappings as also (if less) good: Some perceptually available elements might turn out to be missing in the known structure either because they are meaningless, or because they represent new information; conversely, the diagram may fail to represent some parts of the known structure (especially if this structure is, as said above, the whole knowledge base of a person). In each of these scenarios, a diagram may still be perfectly comprehensible, so there should be good mappings possible although the structures are not isomorphic. Still, in the last two of these cases (new information in the diagram and unrepresented known facts), it would seem that the comprehension of the diagram would include accommodation of the missing information into one of the structures, and so the notion of *embedding* — finding an isomorphic copy of the generalized structure of the diagram within knowledge — would still play a crucial part.

The textbook notion of embedding, applying to the textbook notion of a structure over a vocabulary, is based on mapped elements (and induced mapped element n-tuples) being members in relations if and only if the original elements (and n-tuples) are members in relations corresponding to the same symbols:

Definition 3.2 ([EF95, p.1]) *A* vocabulary *is a finite nonempty set that consists of* relation symbols *and* constant symbols. *Every relation symbol is equipped with a natural number, its* arity. *A vocabulary is* relational *if it contains no constants.*

Definition 3.3 ([EF95, p.1]) *A* structure $\mathcal{A}$ of a vocabulary τ *(for short: a τ-structure) consists of a nonempty set A, the* universe *or* domain *of $\mathcal{A}$, of an n-ary relation R^A on A for every n-ary relation symbol R in τ, and of an element c^A of A for every constant symbol c in τ. An n-ary* relation *on A is a subset of A^n.*

Thus, a τ-structure $\mathcal{A}$ is embeddable in a τ-structure $\mathcal{B}$ if and only if there is a mapping $f : A \to B$ such that for all c in τ, $f(c^A) = c^B$, and for all n-ary relation symbol R in τ and $a_1, \ldots, a_n$ in A, $\langle a_1, \ldots, a_n \rangle \in R^A$ iff $\langle f(a_1), \ldots, f(a_n) \rangle \in R^B$. In the current framework this is only a special case, where the mapping of relations is given. In the more general framework, much like Davidson's field linguist, we can rely only on truth values (and 'syntactic' validity, that is, appropriateness of order and arity). An embedding can now be defined:

Definition 3.4 *For any generalized structures $\mathcal{S} = \bigcup_{i \in I} A_i$, $\mathcal{S}' = \bigcup_{j \in J} A'_j$, a* (partial) embedding *is a (partial) function $f : \mathcal{S} \to \mathcal{S}'$ such that*

- *f is injective (one-to-one),*
- *For all $i \in I$ there is $j \in J$ such that $\vec{f}(A_i) \subseteq A'_j$, and,*
- *for all a in $\mathcal{S}$, and $\vec{x}$ in $\mathcal{S}^{n(a)}$, $\vec{x} \in a$ if and only if $\vec{f}(\vec{x}) \in f(a)$,*

where $\vec{f}$ is f applied memberwise to sequences and sets.

I use $\vec{f}$ for the extension of f to sets by memberwise application to avoid confusion, as f itself is defined extensionally for some sets — the non-atom members of $\mathcal{S}$. One may see that indeed partial isomorphism on structures of a vocabulary is a special case of this definition, where $|I| = 2$ and f is constrained as to the mapping of the non-atoms (and in non-relational vocabularies, some atoms too).

Now we have a cool computational problem to solve: given two generalized structures, find an embedding. More generally and realistically, given two structures $\mathcal{S}$, $\mathcal{S}'$ and a partial mapping f from $\mathcal{S}$ to $\mathcal{S}'$, find an embedding which extends f. More generally still, to account for cases where some graphical features are meaningless, find an embedding extending f which is in some sense maximal, although perhaps partial. Since everything here is finite, a good sense for this maximality is the number of features included. Further, it is intuitively plausible to assume that the extension is done in consecutive steps, in each of which an element from one structure is picked, and an element from the other structure is matched to it. The setting where a partial isomorphism between structures is to be extended, with success being measured by the number of elements mapped, and where the extension is performed step by step as just described, is highly reminiscent of Ehrenfeucht-Fraïssé games. There is a difference, though: EF-games are *adversarial*, that is, Spoiler always tries to look for elements Duplicator cannot match, and as soon as the former finds one, she wins the game; in general, the embedding generated by this game (as far as it can be played) is far from maximal. Interpretation, where the goal is to find a maximal common substructure, may be presented as a *cooperative* EF-game — where the Spoiler actually wants to lose.[4] In many cases there is no difference between adversarial and cooperative games: For example, in a complete one-sided game on finite structures, where the number of rounds is not limited, Duplicator wins iff the structure Spoiler plays on is embeddable in the other structure, regardless of Spoiler's selections, because eventually Spoiler will be forced to select all objects. In other cases, there is a difference: Playing on the rationals $\mathbb{Q}$ and the integers $\mathbb{Z}$, Spoiler can let Duplicator win by choosing only integers (from either structure); thus, Duplicator wins the cooperative game. The problem, then, is to find the maximum k such that there is a strategy for Spoiler which enables Duplicator to win the k-round game.

While we're at it, we might as well look also at the complexity of deciding the adversarial game. In an earlier draft I have tried to claim that it is a fitting description of the communicative scenario sketched earlier, where the addressee has to be ready for whatever change in the diagram the speaker may

[4] Of course, Spoiler may not repeat moves, otherwise this is trivial.

perform. Now, this seems to me more than somewhat contrived. However, while it is hard to justify interest in it from a Cognitive-Science perspective, it is an interesting mathematical and complexity-theoretical question. Further, it is interesting to see the connections between the complexities of deciding the two kinds of EF-games on a given class of structures. Intuitively, deciding an adversarial game should always be harder than deciding the corresponding cooperative game, because the former involves alternating quantifiers over strategies (for all Spoiler strategies S there exists a Duplicator strategy D such that the game played according to S and D generates a partial isomorphism) and the latter involves only existential quantifiers (there is a Spoiler strategy and a Duplicator strategy etc.). We will be interested mostly in one-sided games, where Spoiler can only choose elements from one of the structures, for this limitation corresponds to looking for embeddings, as opposed to general isomorphisms.

Unfortunately, in the general case none of these games is polynomial-time decidable. Deciding any of them for τ-structures decides the substructure isomorphism problem, which is equivalent to its special case where the structures are graphs — the well-known NP-complete subgraph isomorphism problem. Moreover, the adversarial k-round game on τ-structures has been shown to be PSPACE-complete for both one-sided and two-sided games [Pez99] (The proof of hardness for one-sided game is by reduction from the two-sided game. This is an instance of a general phenomenon, where deciding embeddability is consistently harder than deciding a corresponding isomorphism. A more famous instance of this phenomenon is with graphs: While Subgraph Isomorphism, or graph embeddability, is NP-complete, Graph Isomorphism is one of the great open problems of complexity theory, not known to be either NP-complete or polynomial. But I digress). These results are summarized in Table 3.1.

What we can still do, is look for solvable cases of the problems. Before going there, we should note again that this attempt to solve the problem as presented follows from an approach which minimizes known data organization. There are other approaches: There is the above-mentioned idea to rely on compartments of knowledge to reduce the relevant structures to size manageable by exponential algorithms; a different approach allows Messmer and Bunke [MB95] to solve the subgraph isomorphism problem in time polynomial in the size of the embedded graph, with the help of intensive pre-processing of the other graph (they build decision trees which may grow exponential in its size). An approach based on preprocessing established knowledge for fast treatment of online data may be justified in trying to explain user performance, since acquired knowledge does go through some unconscious processing and restructuring. This claim should be convincing to

Adversity	Sides	Rounds	Decision Question	Complexity
Adversarial	1-Sided	k	Every k-subset embeddable	PSPACE-complete
Adversarial	2-Sided	k	k-isomorphism	PSPACE-complete
Cooperative	1-Sided	k	Exists embeddable k-subset	NP-complete
Cooperative	2-Sided	k	Exists isomorphism of size k (equivalent to 1-Sided)	NP-complete
Either	1-Sided	∞	Embedding	NP-complete
Either	2-Sided	∞	Isomorphism	Unknown

Table 3.1: Different versions of EF games, and the complexity of deciding them on finite first-order structures. The parameters are always the sizes of the structures (number of atoms). Where relevant, k is part of the input.

anyone who ever 'slept on a problem' to find it easily solved in the morning, but it is also documented, e.g. by Bartlett's famous experiments on memory in the 1930's (Bartlett told his subjects stories with details that didn't make sense or didn't seem to fit. When the subjects were asked to repeat the stories, weeks later, most of the details were modified to fit better into well-known patterns; even plot lines were changed to make the story more coherent). While an algorithm close in spirit to Messmer and Bunke's, which uses exponential space and builds explicit decision trees, sounds unlikely, there might well be room for other intensive-preprocessing approaches.

In the above two paragraphs, we used τ-structures as a special case to argue that the problem is hard, which is a valid argument; then we implied that regular methods on graphs may be useful for generalized structures, which requires justification. To this end, we need to show that all embedding problems on generalized structures as defined above reduce to (sub-) isomorphism problems on regular structures and on graphs. This will be our first step in the quest for efficiently solvable classes of structures.

3.7 Reductions to Structures and Graphs

First of all, let us see that a game on generalized structures can be translated to a game on structures. This is quite straightforward: Given a generalized structure $\mathcal{S}=\bigcup_{i\in I} A_i$, define a structure $\mathcal{A}$ over the vocabulary $\tau = \{R_k \mid \exists a \in \mathcal{S} : n(a) = k > 0\}$ as follows: The domain of $\mathcal{A}$ is $\mathcal{S}$, and

$R_k(a_0, a_1, \ldots, a_k)$ holds in $\mathcal{A}$ if and only if in $\mathcal{S}$, a_0 is a k-ary relation and $a_0(a_1, \ldots, a_k)$ holds. Intuitively, the relations R_i together encode the meta-linguistic relation 'applies to'.[5] Now, every game on the generalized structures is immediately translated to a game on τ-structures, move for move, with results being preserved. It is also clear that this reduction is polynomial. This also justifies using the above complexity results as upper bounds, and not just as lower bounds.

We can also define a directed graph representation for a generalized structure:

Definition 3.5 *Let $\mathcal{S}$ be a generalized structure then its directed graph representation $G(\mathcal{S}) = (V, E)$ is defined as follows:*

1. *For every $a \in \mathcal{S}$ there is a unique vertex $a \in V$;*
2. *For every n-tuple $\langle a_1, \ldots, a_n \rangle$ of $\mathcal{S}$ members, that is itself a member in some a_0 in $\mathcal{S}$, there is a unique* tuple node *u_0 with a chain of* tuple member nodes*, that is, a vertex u_i for each a_i, with edges $\langle u_{i-1}, u_i \rangle$ and $\langle a_i, u_i \rangle$ for $i = 1, \ldots, n$;*
3. *For every n-tuple $\vec{x} = \langle a_1, \ldots, a_n \rangle$ and a in $\mathcal{S}$, if u is a tuple node for $\vec{x}$ and $\vec{x} \in a$, then there is an edge $\langle a, u \rangle \in E$;*
4. *There are no other vertices or edges in $G(\mathcal{S})$.*

The most important feature of this representation is the reconstructibility of $\mathcal{S}$ from $G(\mathcal{S})$: The members of $\mathcal{S}$ are the vertices of indegree 0; the tuple nodes are the vertices v where for all edges $\langle u, v \rangle$, u is a $\mathcal{S}$ member (or, a 0-indegree vertex); all other vertices are tuple member nodes, and it is straightforward to see that one can read in the graph whether a given relation holds for a given tuple. Thus, the problems of isomorphism and embedding of generalized structures may be reduced to equivalent problems on graphs. The reduction is polynomial in the size m of the structure, though exponential in the maximum arity r of relations: It is easy to see that there can be, say, $O(m^r)$ vertices in the representing graph (for example, let $\mathcal{S}$ be a complete graph, that is, one binary relation E whose extension is all the pairs of atoms, and look at $G(\mathcal{S})$), but no more than $O(rm^r)$ vertices (as there can be at most $\sum_{j=1}^{r} m^j$ tuples, each of them contributing at most $r+1$ nodes, in addition to the m nodes representing $\mathcal{S}$ members).

[5] The relations R_k parallel, with required modifications, the relations ϵ_τ used in [vBD83] for the same purpose — reducing higher-order structures to first-order. τ there stands for a predicate type (see Section 3.4), and is not to be confused with my use of τ for a vocabulary.

The problem with this representation is that it is not known to give a reduction of partial embeddings and isomorphisms. For two structures of maximum arity r it is easy to see that if Spoiler wins the adversarial two-sided game in k moves, then she can win the same game on the graphs in $k+r+1$ moves (by first following the winning strategy of the original game and then following a tuple which shows a mismatched relation), but I know of no way to show any further correspondence. It is also quite clear that for the cooperative games, playing a cooperative game on the representation graphs will be of very little use: the Spoiler there can stick to playing tuple member nodes, which are the overwhelming majority of the vertices, and Duplicator will be able to win as long as there are enough tuples of every size, with little regard to the identity of original structure members. Thus, the vast literature on subgraph isomorphism can be of no help here, unless a better reduction can be used.

3.8 Classes of Structures: Bounded Degree and Grids

3.8.1 Bounded degree

The first attractive target, in order to get a grip on the partial embedding problem, should be its proofs of intractability. That is, the problem may be tractable on classes of structures which do not support the reductions used in those proofs. For both adversarial and cooperative scenarios, these proofs rely on structures where nodes have many neighbors: In the former, Pezzoli's proof of PSPACE-completeness of two-sided EF games [Pez99] uses complex 'gadgets' — graph parts of specific types — to reduce the satisfiability of quantified boolean formulas to the k-isomorphism of structures, and a reduction from two-sided games to one-sided games for the k-embedding; the proof of NP-completeness of the maximal embedding, for the cooperative game, is a trivial reduction from the problem of finding a k-clique. To add to our joy, isomorphism of graphs of bounded degree is decidable in polynomial time [Luk80].

Unfortunately, this limitation does not make sense for our application domain: after we translate from generalized structures to τ-structures or graphs, limiting the degree of the structure (that is, for any member of the structure, the number of tuples containing it that are members in relations) means limiting the size of the relations in the generalized structure. This would mean, e.g. for a color encoding, limiting the number of red elements. So we must look for some other limitation.

3.8.2 Grids

Rather than look for another general limitation on structures, 'from above' as it were, we may try to look from below: Find a relevant class of structures where the problem can be solved, and try to extend it as far as we can. An obvious candidate, if we are to investigate representations on a plane, is the class of grids, or Cartesian products of linear orders. If the problems may be decided in polynomial time on grids as structures, then they are polinomially decidable on grids as generalized structures, because there are only two relations. Also, this class of structures already represents two interesting classes of diagrams: On one hand, all kinds of X-Y charts, including bar-graphs; on the other hand, block diagrams where the two-dimensional location of a block is meaningful.

Definition 3.6 *A finite grid $\mathcal{H}$ is a triplet $(H, <^x, <^y)$ where*

- *H is the set of pairs (x, y) with $0 \leq x < X$ and $0 \leq y < Y$ for some natural X, Y. We say $\mathcal{H}$ is of dimension $X \times Y$,*
- *$<^x$ is a binary relation on H defined by $(x_1, y_1) <^x (x_2, y_2)$ iff $x_1 < x_2$, and likewise,*
- *$<^y$ is a binary relation on H defined by $(x_1, y_1) <^y (x_2, y_2)$ iff $y_1 < y_2$.*

Before analysis of the different game variations, I would like to throw in a word about the input parameters, in terms of which we evaluate complexity. As one may guess from Definition 3.6, and as will become apparent in the following, all the information actually expressed by a grid is its dimensions. Hence, a grid can be given in $O(\log X + \log Y)$ bits. To keep the analysis of grids in the general context of structure matching — both to enable comparison with the intractability results cited earlier, and to prepare for extensions of grids — I have used instead the size of structures, $X \times Y$, as my input size. Actually, since the decision procedures indeed rely only on the sizes of structures and the number of rounds in the games, I have just assumed that comparison and multiplication take constant time, and exponentiation, accordingly, time logarithmic in the exponent. I have also assumed that in spite of the above, the sizes of the structures are available at no effort; this assumption can be dropped for an additive linear cost factor. As before, the number of rounds k is taken as part of the input. Under these assumptions, most of our problems are polynomial-time, and even linear-time, decidable on finite grids.

Lemma 3.1 (Trivial) *Cooperative k-round EF-games on grids can be decided in constant time. So can any ∞-round games.*

Proof: The maximal common substructure of grids $\mathcal{H}_1$, $\mathcal{H}_2$ of respective dimensions $X_1 \times Y_1$, $X_2 \times Y_2$ is a grid of dimension $\min(X_1, X_2) \times \min(Y_1, Y_2)$, so the maximal partial embedding can be solved in constant time. In a cooperative EF-game on non-isomorphic grids, Duplicator wins if and only if the number of rounds is smaller than or equal to the size of the common substructure.

∞-round games correspond to the problems of embedding (one-sided) and isomorphism (two-sided). Both embeddability and isomorphism can be trivially decided by comparing the sizes of the grids. QED.

The decision problems of the adversarial games on grids reduce to the same games played on linear orders by a straightforward divide-and-conquer argument: if Duplicator can win a game on grids $\mathcal{H}_1$, $\mathcal{H}_2$, then by simple projection of the winning strategy on each of the axes, she wins the same game on $<^x_1$, $<^x_2$, as well as on $<^y_1$, $<^y_2$. Conversely, if she wins the two games on linear orders, this leads to a winning strategy in the game on $\mathcal{H}_1$, $\mathcal{H}_2$: if Spoiler plays (x_0, y_0) on one grid, Duplicator can play (x_1, y_1) on the other grid, where x_1 is her response to x_0 in the game on $<^x_1$, $<^x_2$ and y_1 is her response to y_0 in the game on $<^y_1$, $<^y_2$.

Theorem 3.2 *Adversarial one-sided k-round EF-games on grids can be decided in linear time.*

Proof: Assume w.l.o.g. that Spoiler plays on $\mathcal{H}_1$. If $X_1 < X_2$ and $Y_1 < Y_2$ then Duplicator wins trivially for any number of moves. Assume then, again w.l.o.g, that $X_1 > X_2$ and further $X_2 < Y_2$ or $Y_1 < Y_2$ (that is, pick the unembeddable order where the corresponding order in $\mathcal{H}_2$ is shorter), then the game reduces to a game on orders of sizes X_1, X_2; Spoiler wins in n moves if $X_2 < 2^n - 1$, and loses otherwise. The proof is in both directions by induction, using a divide-and-conquer strategy.

Assume, then, that $X_2 < 2^n - 1$, and we will present a winning strategy for Spoiler. For the base of the induction, let $n = 1$ then $X_2 < 1$, that is, the order on Duplicator's side is empty, and Spoiler's first move wins. For the induction step, let $n > 1$ and assume the result holds for $n - 1$. In the n-moves game, Spoiler plays $s_1 = \frac{X_1}{2}$ (rounded, say, down) for her first move. Duplicator plays some move d_1. Assume w.l.o.g. that $d_1 < \frac{X_2}{2}$ (otherwise the argument is mirrored) then Spoiler can (by induction) win the $(n-1)$-moves game on the segments of the original orders below (when mirrored, above) s_1 and d_1. One may see that $s_1 > d_1$, and, since $X_2 < 2^n - 1$ means $X_2 \leq 2^n - 2$, we get $d_1 < \frac{X_2}{2} \leq \frac{2^n-2}{2} = 2^{n-1} - 1$.

For the converse, assume that $X_2 \geq 2^n - 1$, and we will present a winning strategy for Duplicator. For the base again let $n = 1$ then any response by

Duplicator for any move by Spoiler wins. For the induction step let $n > 1$ and assume the result holds for $n-1$. Whatever is Spoiler's first move s_1, Duplicator responds with $d_1 = \frac{X_2}{2}$ (rounded down). Now, Duplicator considers the game to be divided to two independent games, one above and one below s_1 and d_1. The two games are both of size $\geq 2^{n-1} - 1$, and at most $n-1$ moves may be made in each, so each of them can be won by induction.

The decision procedure, then, consists of a constant number of comparisons (to identify which of the dimensions in the input takes the role of X_2 in the analysis above), and then, in the worst case where the Spoiler's structure is not embeddable in Duplicator's structure, calculating $2^k - 1$ and one more comparison. The only variable factor here is the exponentiation, which takes time logarithmic in k and hence linear in its size as input. QED.

For adversarial two-sided k-moves EF-games, I have no proof of polynomiality, but a possible route to such a proof goes through the following two results:

Theorem 3.3 ([IK89, Theorem 14]) *In adversarial EF games on linear orders, if Duplicator has a winning strategy for 3-pebbles, n-moves game, then she has a winning strategy for the k-pebbles, n-moves game for any k.*

This theorem holds for both one-sided and two-sided games, and it extends to grids by a straightforward argument, a lot like the one above: if Duplicator can win the 3-pebbles, n-moves game on grids $\mathcal{H}_1$, $\mathcal{H}_2$, then by simple projection of this strategy on each of the axes, she wins the 3-pebbles n-moves game on $<^x_1$, $<^x_2$, as well as that on $<^y_1$, $<^y_2$. By Theorem 3.3, she then wins the k-pebbles, n-moves games on these pairs of structures, and this leads to a winning strategy in the k-pebbles, n-moves game on $\mathcal{H}_1$, $\mathcal{H}_2$: if Spoiler plays (x_0, y_0) on one grid, Duplicator can play (x_1, y_1) on the other grid, where x_1 is her response to x_0 in the game on $<^x_1$, $<^x_2$ and y_1 is her response to y_0 in the game on $<^y_1$, $<^y_2$. Then we want to apply something like:

Theorem 3.4 (Follows from [IL90]) *For any finite k, the adversarial ∞-round k-pebble two-sided game on any structures is decidable in polynomial time.*

This still leaves open the question as to n-moves (k-pebble) two-sided games; I would guess that they are polynomial-time decidable, because of the polynomiality of the n-moves one-sided games on one hand, and of unlimited k-pebble games on the other, but I have no proof of this yet.

3.8.3 Extending grids

Our hopes to extend decidability on grids to more interesting structures are held short very soon. The obvious next step is to add one-place predicates, or, moving to colored grids. Alas, deciding any one-sided game on colored grids — cooperative or adversarial, unlimited-rounds or n-rounds with n part of the input — is NP-hard, by a reduction from k-CLIQUE, as follows.

Definition 3.7 *An* ordered graph G *is a triplet* $(V, E, <)$ *where*

- V *is a non-empty set (of vertices),*
- E *is a binary relation on* V*, and*
- $<$ *is a linear order on* V

An ordered graph is just a graph, with an ordering of its vertices.

In most discussions of graph algorithms, it is assumed that one may choose an arbitrary vertex from a set of vertices. Under this assumption, an arbitrary order on a graph (V, E) can be computed in time $O(|V|^2)$ by the following straightforward algorithm:[6]

```
def Order(V):
    less_than = ∅
    small = ∅
    for v in V:
        for u in small:
            less_than = less_than ∪ {(u,v)}
        small = small ∪ {v}
    return less_than
```

An ordered graph $G = (V, E, <)$ can be translated to a colored grid $\mathcal{H} = (H, <^x, <^y, D, E_H)$, as follows:[7]

- H is $V \times V$,
- $(x_1, y_1) <^x (x_2, y_2)$ iff $x_1 < x_2$,
- $(x_1, y_1) <^y (x_2, y_2)$ iff $y_1 < y_2$,
- D is the diagonal of H, i.e. $D((x, y))$ iff $x = y$, and

[6] The pseudocode syntax is that of the Python programming language. Blocks are defined by indentation, `def` starts a function definition, and `for` loops through the elements of a set. For more details about this language, see `http://www.python.org`.

[7] This translation was given by Anuj Dawar (personal communication).

- $E_H((x,y))$ iff $E(x,y)$

Clearly, each of these elements can be computed from G in time $O(|V|^2)$. This is indeed a translation: Vertices u in G are represented by elements (u,u) of D in $\mathcal{H}$; for any G vertices u,v, we have that $u < v$ iff $(u,u) <^x (v,v)$ iff $(u,u) <^y (v,v)$. A little more complicated is the representation of edges: $E(u,v)$ iff $\exists z((u,u) =^x z \wedge (v,v) =^y z \wedge E_H(z)$, where $=^x$ and $=^y$ are the e-quivalence relations on H induced by $<^x$ and $<^y$ respectively. More intuitive-ly, to find out from $\mathcal{H}$ if $E(u,v)$, find the grid points u' and v' representing u and v respectively, and check E_H on the point z which lies in the row of u' and in the column of v'.

Now, the question of embeddability of the complete graph K_k is indifferent to ordering: Any ordering on a complete graph is meaningless and arbitrary. The reduction from k-CLIQUE is based on finding K_k as a subgraph in the input graph: Given an unordered graph $G = (V,E)$, calculate the ordered graph $G' = (V,E,Order(V))$. Likewise, calculate from $K_k = (V_k,E_k)$ the complete ordered graph $K'_k = (V_k,E_k,Order(V_k))$. Translate G' and K'_k to colored grids $\mathcal{H}$ and $\mathcal{H}_k$ respectively, and play a one-sided game on $\mathcal{H}$ and $\mathcal{H}_k$ (Spoiler plays on $\mathcal{H}_k$). There are k^2 points in $\mathcal{H}_k$. Duplicator wins the unlimited-rounds game, or, equivalently, the k^2-rounds game, whether Spoiler tries to win or lose, if and only if there is a k-clique in G.

Thus, other directions for extensions should be followed. I would suggest two questions which I think may turn out particularly interesting, at the intersection of the math and the application-domain: adding constants to colored and uncolored grids, and the effects of the given partial embedding which have not been studied here at all.

3.9 Conclusion

Different researchers have tried to explain the efficacy of diagrammatic rea-soning in different ways. Most of the more logically oriented accounts look at diagrams as part of representation systems, and focus on computation-al gains through the qualities of the connections between representing and represented objects. Here, I took a different route: I was able to show that for some classes of diagrams, the problem of Interpretation is easy to solve because of qualities of the diagrams themselves, seen as structures.

I took a direct-mapping approach, which is ‘computation-friendly’: In the terminology of Stenning & Oberlander [SO95], this approach can only deal with MARSs, the least expressive class in the hierarchy of representation sys-tems, and the easiest to compute with. Even with this approach, I ran quite quickly into hard computational problems. This is because what seemed like an innocent, little step — the adding of colors to grids — in fact enriched the structures enough to express almost anything. I believe more interesting results can be brought forth by more careful examination of classes of structures, and variations of EF-games (Johan van Benthem has suggested looking into more involved relations approximating isomorphism, such as bisimulation. This runs against my intuitions of how the human cognitive system works, but might be interesting nonetheless). Also, the direct-mapping approach may be extended, or replaced by a more expressive line of analysis. This would make the account more complicated, but polynomial-time solvable problems may still be found. Proofs of intractability of viable classes of diagrams may also be valuable, providing predictions that these classes of diagrams would not retain usefulness as they grow in size and complexity (Colored grids are not a viable class of diagrams. This class includes far too many objects, e.g. plain pictures).

I think the main achievement here is the paving of the road: I have defined a general framework for this direction of research, and I think the results reported here show, that it holds a potential for better understanding of diagrams and how they are useful. In addition, this research accentuates the promise of mutual fertilisation between Cognitive Science (and in particular, the study of Diagrammatic Reasoning), and the mathematical fields of Complexity Theory and Finite Model Theory.

3.10 Appendix: A Little Lemma on Colored Grids

In the process of studying the questions discussed in Section 3.8, I was at one point trying to prove that n-moves, two-sided adversarial EF-games on colored grids are polynomial-time decidable. I now have a good reason to believe this cannot be proved — adversarial games are usually harder to decide than the corresponding cooperative games, and I have shown the respective cooperative game to be NP-hard (In the reduction from CLIQUE, the cooperative game is indifferent to one-sidedness). However, in trying to prove this, I have come upon a curious little lemma on colored grids, which, because of the generality of these structures, may be of interest.

Lemma 3.5 *Let $k > 9$. On colored grids, any sentence of quantifier rank k and k variables can be expressed in (has an equivalent sentence of) the same quantifier rank and $k - 1$ variables.*

Proof: We prove the equivalent statement: For all $k > 9$, and pairs of colored grids, Spoiler can win the k-moves, k-pebbles game starting from an empty configuration (i.e. no pebbles may be set on the board when the game begins) if, and only if, she can also win the k-moves, $(k-1)$-pebbles game on the same grids. Clearly, if Spoiler wins the k-moves, $(k-1)$-pebbles game on a pair of grids (or any structures), she also wins with an additional pebble; the interesting part is the converse.

Assume, then, that Spoiler has a winning strategy for the k-moves, k-pebbles game, and we will show how she can save a pebble. In the k-moves, $(k-1)$-pebbles game, let her play the first $k-1$ moves from the winning strategy of the k-pebbles game. Now, by the winning strategy, there is (in the k-pebbles game) a move (x, y) in one of the structures which is a winning move. In the $(k-1)$-pebbles game, Spoiler can play (x, y) only by removing some pebble (x_0, y_0). But it will still be a winning move if the relations $<^x, <^y$ between (x, y) and (x_0, y_0) could be somehow made to hold (since the relations between (x_0, y_0) and all other pebbles hold by the previous phase of the game, and those between (x, y) and all other pebbles by the final phase). This, in turn, can be done if (x_0, y_0) can be picked in such a way that there are pebbles between (x, y) and (x_0, y_0) along $<^x$ and along $<^y$. Look at the four quarters of the grid generated by the axes going through (x, y): The presence of 3 pebbles in any of these quarters ensures the existence of a suitable (x_0, y_0). Consequently, the presence of 9 pebbles on the grid ensures that there is a winning move in the game, QED.

Note[8]**:** This lemma cannot be used to prove a similar property for graphs using the reduction given above, because the (quite straightforward) translations from the language of ordered graphs to colored grids and back add variables. That is, if we took a formula in the language of ordered graphs, translated it to an equivalent formula in the language of colored grids according to the reduction of §3.8.3, used Lemma 3.5 to eliminate a variable, and translated the modified formula back to the language of ordered graphs, we would end up with more variables than we started with.

[8] Anuj Dawar (personal communication).

Chapter 4

Visual Reasoning and Case-Based Reasoning: Hyperproof and The EGG

Hyperproof is an interactive computer system designed mainly for the purpose of teaching logic. The main theme of the system is connecting sentences of First-Order Logic (FOL) with (diagrams of) models for these sentences. In [BE99] and in related papers, its creators describe this system and the ideas guiding it, and claim that student-users of Hyperproof engage in Visual Reasoning. Subscribing to that view, Oberlander et al. report in [OMC$^+$99] a study on multimodal reasoning, using Hyperproof, which relates students' cognitive styles — their preferences for using different modalities — and the change in their performance following multimodal teaching with Hyperproof.

In this chapter I argue that Hyperproof's approach does not distinguish Visual Reasoning from Case-Based Reasoning, although the two can (and should) be separated; and that ignoring this distinction in cognitive research — as has been done in [OMC$^+$99] — can lead to overstated results. We suggest a different interpretation of the empirical findings of Oberlander et al., based on this insight. Finally, we propose to test the two interpretations with a teaching system that uses a graphical logical language. By using graphics for the logic, rather than for the model, we may gain a better understanding of the effect of visual aids: The effect of reasoning with a concrete model may thus be eliminated, and one can better establish the relation between modality and performance.

4.1 A Short Introduction to Hyperproof

This section is based on [BE99], which describes the evolution and motivations of Hyperproof. Barwise and Etchemendy mention first two related systems, *Turing's World* and *Tarski's World*.

Turing's World is an aid for teaching Turing machines. It presents — and allows manipulation of — state diagrams for Turing machines. It also allows the user to feed input to a machine and watch it run. Some properties of machines, which are not easily seen in the equivalent 'sentential' representation — quartets representing the machine's state transitions[1] — become obvious in these diagrams; some notions become much easier to grasp. For example, in checking whether a machine halts one needs to identify its cycles, which is significantly easier using a visual representation.

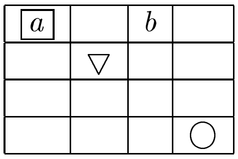
Tarski's World presents to the user two displays: one contains sentences of FOL, the other shows a simple 'world': A checkered board populated by 3-dimensional blocks of different shapes and sizes. In the simplest task, students are invited to decide whether the sentences are true in the displayed world. More complex tasks ask the students to produce one of the two parts on their own: Given a world, to express facts about it in FOL, or, given a set of sentences, to construct a model for them. When a student gives a wrong answer, the system engages her in a game, based on the standard game-theoretic semantics for FOL, which in most cases shows her the reason why a formula's evaluation in a given model is not as she thought.

Watching their students at work with Tarski's World, Barwise and Etchemendy realized that the kind of reasoning that the students were performing was very different from the kind formalized by proof systems; namely, they would draw conclusions directly from the diagram, and not through FOL sentences. Yet, this kind of reasoning had logical characteristics: Some identifiable kinds of deduction were valid. The next problem and its solution exemplify the kind of inferences involved:

> In Figure 4.1, letters stand for blocks in the matrix. A bare letter is a block of unknown shape, and a shape with no letter is not (yet) identified. Find out b's shape from the information in the matrix and the accompanying sentences.

The solution might go like this:

[1] These quartets are interesting in themselves. For example, while each of them stands for a single state, making them token-referential, their members are type-referential. I accept Barwise and Etchemendy's description of them as 'sentential' for this discussion, but note that there is a philosophical issue here.



- There is another block with b's shape, but not on b's row.
- A block named c has shape different from b's, and is placed in a row other than b's.
- c is to the right of b

Figure 4.1: What is b's shape?

> from the first sentence, b is either a circle or a triangle. By the second sentence, c is either the given triangle or the given circle. Further, if c is a triangle then b is a circle and vice versa, so we are left with the two possibilities depicted in Figure 4.2. The third sentence eliminates the left option, so we deduce b is a triangle.

When they realized the validity of this kind of inference, Barwise & Etchemendy proceeded to study it in two different ways. One was the development of a theoretical framework to analyze it, using the influential *Situation Theory* (see e.g. [CMP91]). The other was to design Hyperproof, a system where this kind of reasoning could be methodically engaged in: Formal proofs can be generated and 'written' by the student, and they may include inference rules such as *Observe*, which justifies a sentence by the fact that it holds in an associated diagram. Figure 4.3 is a snapshot of a Hyperproof session, showing the situation pane (block-world diagram), sentence pane with a proof in progress, the virtual keyboard (which facilitates inserting symbols such as $\exists$, not found on most keyboards), and a goal to be achieved. The interesting part is of course the sentence pane: The proof system is a natural deduction system, and the diamond shapes represent

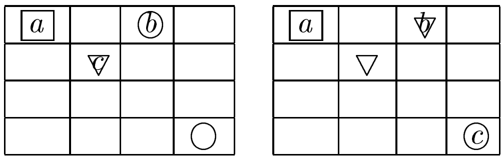


Figure 4.2: The last two possibilities

diagrams. The situation pane displays the diagram at the highlighted step (the given situation), and the other diagrams were derived from it by making further assumptions (which would be discharged later).

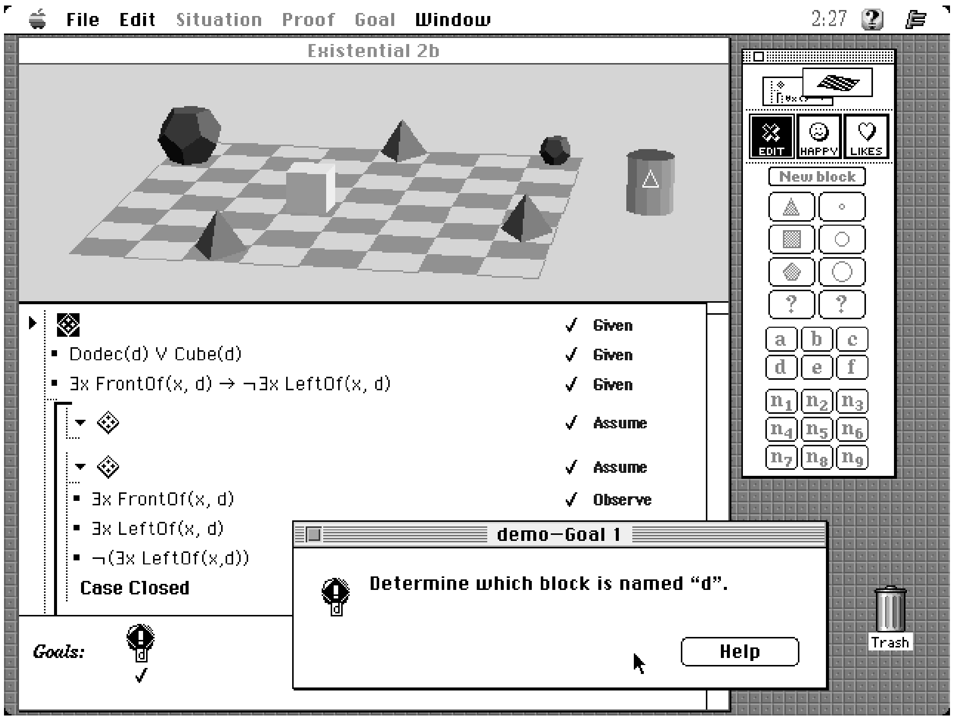


Figure 4.3: Hyperproof sample screen. Taken from Hyperproof's web page.

4.2 Research with Hyperproof

Using Hyperproof, a group of researchers in the HCRC in Edinburgh investigated the effects of using diagrams in logic education. In a series of studies, they linked the benefit students gained from courses with Hyperproof — i.e. the improvement in their performance in reasoning tests following such courses — to their cognitive style: Psychometric tests could classify the students as good or bad diagrammers. The students' reasoning skills were tested, before and after a one-quarter course, in a paper-and-pencil test; this test was assembled from problems of a blocks-world a lot like Hyperproof's, but presented using natural language rather than first-order logic. The changes in students' performance in this test were found to be in correspondence with the cognitive-style classification: After being taught a course with Hyperproof, the good diagrammers improved a lot, while the bad diagrammers improved little or deteriorated. A control group was taught a similar course with a graphics-disabled version of the software (to control for motivational and like factors having to do with the use of computers); in this group, the results were the other way around, and it was the bad diagrammers who benefited the most.

In the study reported in [OMC+99], the students of both classes also sat computerized tests at the end of the course. The questions in these tests were different for the two groups, because the students of the control group were not conversant with the graphical tools of Hyperproof. In this test, all actions performed by the students were logged, as well as the final proofs generated. Most of the paper is dedicated to statistical analyses of the corpus of proofs collected, in terms of the frequencies of rules and rules-combinations used on one hand, and in terms of measurable structuring in the proofs (i.e. using abstract superordinate cases and sub-case structures vs. a series of independent, concrete cases). The researchers tried to use these analyses to find out what makes the better diagrammers better, that is, to find differences between the statistic characteristics of proofs generated by students of the two groups. Two hypotheses were compared: According to one, the better diagrammers were just making more use of the visual information; according to the other, they were better able to transfer information between the visual mode and the sentential mode. [OMC+99] reports evidence for the latter. The details of the statistics are skipped here, as they are immaterial to the point I wish to discuss.

4.3 Visual and Case-Based Reasoning

In this chapter, I wish to draw attention to the difference between Visual Reasoning, which I see as *the process of inferring data by the interpretation and/or manipulation of images*, and Case-Based Reasoning, *the process of inferring data by considering models or situations* (One may even make a further distinction, between strictly model-based reasoning and non-model-based reasoning which uses division into cases; such reasoning is formalized in systems of natural deduction. I focus here on the effects of the use of the model, but use the name Case-Based Reasoning because the effects of divide-and-conquer strategies in proofs may also be substantial, and because it is a better description of what goes on in Hyperproof). I suggest that while Hyperproof does enable Visual Reasoning, this kind of reasoning is not explicated in it and not modeled by its operations; all the novel inference rules involving diagrams are in fact rules of Case-Based Reasoning. Consider the following problem:

> There are four guys: Alfred and Ben are the same height, and Alfred wears a top-hat. A guy wearing a cap is taller than them, fatter than Alfred, and skinnier than Ben; a guy wearing a sombrero is taller yet, and fatter than Ben. What is on Ben's head, if we also know that Ben and someone taller than him are wearing the same kind of head-garment; that Mr. Cole is taller than Ben, and wears a different head-garment; and that Mr. Cole is fatter than Ben.

This problem is isomorphic to the former problem: One may easily translate the names, garments and bodily comparison relations mentioned here to the letters, shapes and spatial relations, respectively, used there. Conversely, the solution offered for the diagrammed problem may be immediately translated back into a solution for the sentential one. I suggest further that this translated solution, like the problem, has little to do with the visual mode. If these claims are accepted, then I have shown that the kind of Case-Based Reasoning exemplified above need not be Visual.

While this certainly holds at the explicit level, a skeptic may still argue that a person would use mental images for solving the verbal version of the problem. This point may be supported by performance differences: While this has not been thoroughly tested, reviewers of early drafts of this chapter have commented that the graphic version of the problem is a lot easier to solve than the verbal one. I think this is beside the point: The issue is philosophical-methodological, not psychological or empiric. I am trying to demonstrate the contingency of visual representation for the kind of reasoning explicated and formalized in Hyperproof. By giving an explicit, formalizable, and non-visual line of reasoning, isomorphic with the visual one, I believe I have shown this contingency.

In Hyperproof, however, Visual and Case-Based Reasoning cannot be separated (This has been noted in the past, e.g. by Remko Scha[2], and I claim no novelty). When we look again at [OMC+99], with this observation in mind, we may see that their analysis shares this lack of distinction: The psychometric tests they classified students with actually measured not their ability to use visual representations, but their performance in solving 'determinate' constraint-satisfaction problems, i.e. problems which "provide premises which determine a (nearly) unique logical model" [OMC+99, p. 368]; the claim is that such problems are easy to solve by referring to a diagram or a model. While formally the groups are called 'DetHi' and 'DetLo', informally these terms are equated with good and bad diagrammers respectively. This is consistent with the use of Hyperproof diagrams in the research. But

[2] Personal communication

the claim here is that the generalization to multimodal information processing is not justified, as long as the effects of the visual mode have not been separated from those of the concreteness of the problems. Taking the role of the devil's advocate, I suggest a different reading of the findings, supporting the hypothesis of information transfer over that of visual information use: The people who are better at Case-Based Reasoning are those who transfer information between model and logical theory, not the ones who focus (only) on the model. I do not, of course, claim to have grounds to prefer this interpretation to the researchers' one: The performance differences cited earlier in this section actually give reason to favor theirs. But I do claim that their study leaves open the question: How much of the results should be attributed to representation and how much to content?

A more general question which arises here is as to the independence of Visual Reasoning from Case-Based Reasoning: Is it possible to systematically model and record Visual Reasoning separated from Case-Based Reasoning? We have arguably shown the converse independence, but this question stays open (On a side note, the converse independence implicates a way of dissociating Visual and Case-Based Reasoning — a non-graphical model-based logic teaching curriculum might enable measurement of the content effects alone, thereby solving half of the methodological problem).

Both questions could be answered if we had a teaching system modeled generally after Hyperproof, which utilized a visual logical system not based on visualizing a model for formulas; a system that visualized the logic itself, like Turing's World visualizes Turing machines, and where formulas could be manipulated as machine states can be manipulated in this program. Such a system would present a positive answer to the second question, demonstrating the independence of Visual Reasoning; and if it also supported regular first-order logic, it could facilitate investigation of the transfer of information between modalities, separating it from the transfer of information between logical theory and model.

4.4 Diagramming Logic

An interactive logic visual system, like the one sketched in the last paragraph, needs a system of diagrams, which is at least equal in expressive power to FOL. Such is the system of Existential Graphs, devised by C. S. Peirce (parts of Peirce's original work may be found in [Pei58, volume 4] and [Pei76, volume 3], but for a full, coherent description see [Rob73]). This system has two basic parts, or stages: *Alpha*, which is equivalent to Propositional Logic, and *Beta*, an extension of Alpha which is equivalent to First-Order Predicate Logic. These systems come equipped with a proof theory, i.e. axioms and inference rules, expressed as diagrams and permissible manipulations respectively. Peirce went on to extend Beta in different ways, to encompass higher-order logic, modal logic and self-reference, but the more complicated systems he produced will not be dealt with in this thesis.

4.4.1 Alpha

The basic notion of Existential Graphs, analogous to the formula of Propositional and Predicate Logic, is the graph. A graph in Alpha may be defined recursively as follows.

Definition 4.1 *A* graph *is defined recursively by:*

1. *An (empty) area is a graph, called a* sheet of assertion*;*
2. *A sentence written (in Peirce's language, scribed) on a sheet of assertion is a graph;*
3. *Two graphs on the same sheet of assertion are a graph;*
4. *A graph enclosed by a thin closed line, called a* cut*, is a graph. The area enclosed in the cut is considered a separate sheet of assertion, distinct from the one outside. The cut and its contained area are considered to be a graph on the external sheet of assertion, and the area inside is said to be dominated by the area outside. And, finally,*
5. *Nothing else is a graph.*

The semantics for this language is quite simple: An empty sheet of assertion stands for truth, or $\top$; a sentence scribed on a sheet of assertion asserts its truth in the world described by that sheet; two graphs on the same sheet are conjoined, and a cut negates its contents. Thus, Alpha has the same expressive power as Propositional Logic.

As for the proof theory, it includes one axiom — the empty sheet of assertion, and three pairs of inference rules, or permissions:

Definition 4.2 *An area enclosed in an even number (or zero) of cuts is called* a positive context*. An area enclosed in an odd number of cuts is called* a negative context*.*

1. Any graph may be scribed (written) in any negative context, and any graph may be unscribed (erased) in any positive context;

2. Any graph scribed in a dominating area may be iterated (copied) on any dominated area which is not part of the graph (or on the area containing the original copy); any graph which may be the result of such iteration, may be deiterated (erased).

3. A pair of cuts with nothing between them (known as a double cut) may be scribed or unscribed around any graph in any context.

A proof in Alpha, then, is a series of diagrams starting from some premises and progressing by permissible manipulations to the conclusion.

4.4.2 Beta

Beta adds to Alpha *lines of identity*, which are thick lines connected to places in sentences where nouns have been erased. Lines of identity express the identity of the objects signified by the spots at their ends, and are understood as existentially quantified variables. They can cross cuts, and the quantification is evaluated at the outermost extremity of the line. Lines of identity may also branch, signifying the identity of more than two objects.

A line of identity may be scribed unattached to sentences; in this case it just signifies the existence of some object. Beta was designed with an assumption that the universe of discourse is not empty, so the existence of an object is considered an additional axiom, and an unattached line of identity may be permissibly scribed in any context. The (other) permissions are modified as follows:

1. Any graph may be scribed (written) in any negative context, and lines of identity may be connected in a negative context or across an odd cut; any graph may be unscribed (erased) in any positive context, and lines of identity may be disconnected in a positive context or across an even cut;

2. Any graph scribed in a dominating area may be iterated (copied) on any dominated area which is not part of it (or on the area containing the original copy); any graph which may be the result of such iteration, may be deiterated (erased). Corresponding spots in two iterations of the same graph may be connected by lines of identity, or such lines may be removed.

3. A double cut may be scribed or unscribed around any graph in any context, and this permission is not affected by lines of identity running from outside the outer cut to inside the inner one (nor are the lines affected by the exercise of the permission).

Figure 4.4 shows an example of a proof in Beta. This is mostly intended as an illustration, to give the reader a taste of Peirce's logic; for more details and discussion of specific problems, the interested reader is referred to [Rob73]. The proof starts with the premises, which are that Frankie loves Johnny (more formally, there is something that is Frankie which loves something that is Johnny), and that if anybody loves somebody, they respect them (formally, there do not exist two individuals x and y such that x loves y, and x does not respect y). The second premise illustrates the Existential Graphs idiom for implication, which is a cut containing the condition and the negated consequence (that is, $p \to q$ is rephrased as $\neg(p \wedge \neg q)$). We wish to prove that Frankie respects Johnny. The first step connects lines of identity from the two premises. This allowed by the first permission, as it is done across an 'odd' cut — one that is enclosed by an even number of cuts (or no cuts). The next step is deiteration: The inner copy of 'loves', which could in principle have been an iteration of the outer one, is unscribed. In the next step, a double negation is removed, and finally, the redundant unenclosed proposition is deleted, Q.E.D.

4.4.3 Diagramming Logic?

We have seen that Beta is a diagrammatic system equivalent to First-Order Logic. But is it a good diagrammatization? Is it helpful? How, and what, will it help us teach?

If we try to understand the semantic meaning of propositions expressed in Beta, we are not much better off than with sentential systems: Beta, in this case, is type-referential, and collecting all the information relating to some object or predicate is just as hard as in the sentential system. We may be a little better off, because the syntactic structure is easier to detect: It is somewhat harder to miss a parenthesis, or mistake one variable for another, with Beta. For a novice running into Beta for the first time, these advantages are probably not enough to compensate for the strangeness and novelty.

But Beta's strength *does* lie in these advantages, for Beta visualizes not the semantics, but the syntax. Its cuts are a perfect visualization of operator scope, and its lines of identity are a very good visualization of the connections created between different terms when they use the same variable. When the symbols scribed on its areas are seen not as predicates and nouns, but as occurences thereof, then it is also token-referential.[3]

So Beta should be helpful for presenting, analyzing and manipulating formulae as syntactic structures (In fact, I suspect that its greatest contribution

[3]This description owes a lot to Stenning [SteTA].

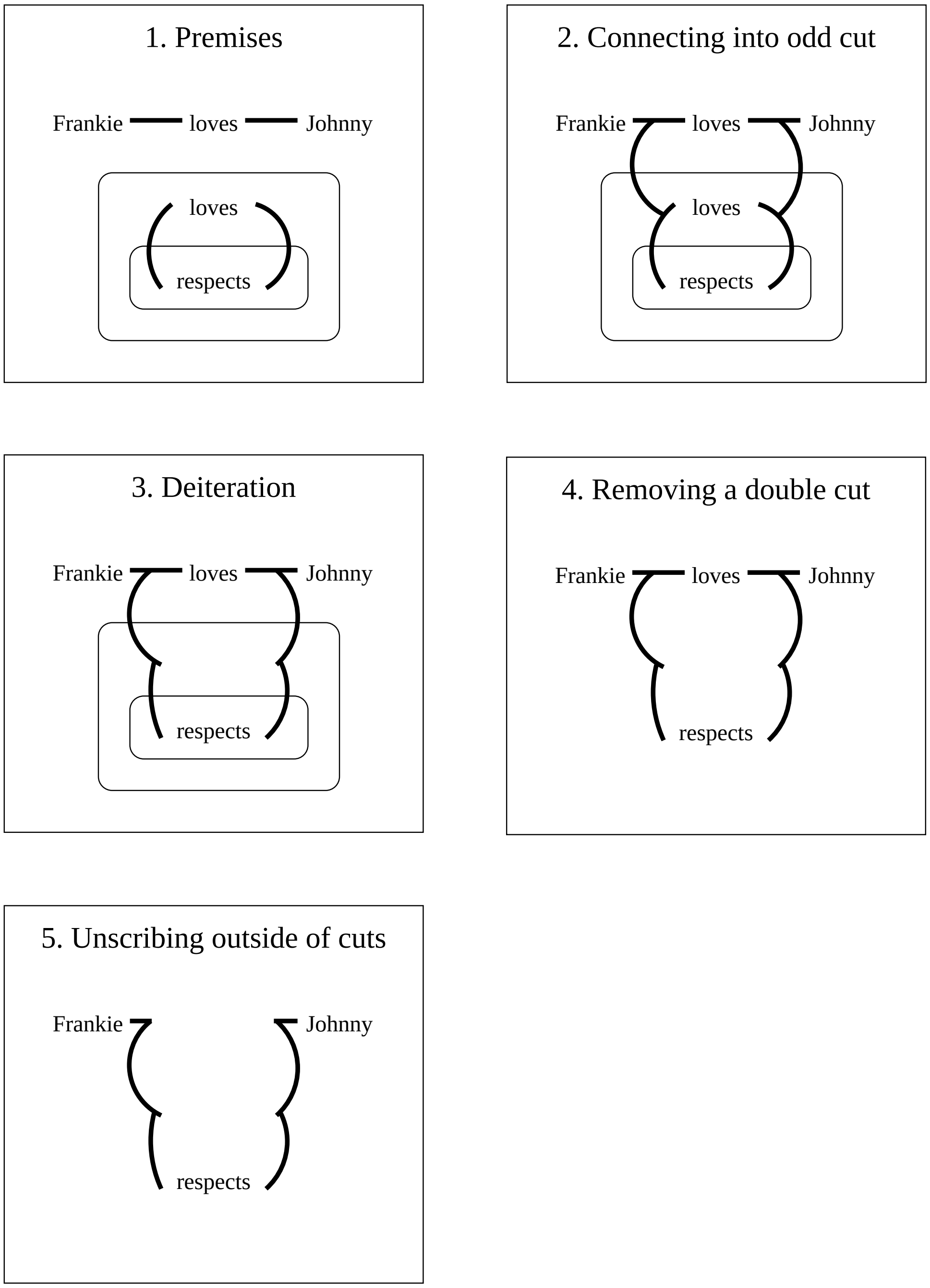


Figure 4.4: A simple proof with Existential Graphs.

might be to teach students that formulae may be seen that way). Accordingly, it would be more useful in teaching the more proof-theoretic side of logic — where students find it exceptionally hard to disconnect the formulae from their propositional semantics, and subject them to pure syntactic manipulations. This difficulty is heightened by the lack of a syntactic decision procedure for FOL, which forces students to rely on the semantics to guide their manipulations. Ideally, after studying Beta, a student will see the premises as one image, the wanted conclusion as another, and the process of finding a proof as a search for a way to transform the former into the latter.

However, if we look at Beta as presented here, even the translation of premises into a diagram is not trivial: It requires, for example, expressing all connectives in terms of negation and conjunction, a feat which already takes some logical maturity to accomplish, and a lot of practice to accomplish automatically, as portrayed in the ideal. Further, if we want skills gained by studying the transformations to transfer to classical proof systems, the permissions need to make more logical sense.

A way to remedy these problems has been recently suggested by Sun-Joo Shin [Shi99]. Shin suggests that the efficacy of Existential Graphs may be improved by two steps:

1. Changing the way Beta diagrams are read. Effectively, this ammounts to idiomatization of the connectives: Instead of teaching the system as built from conjunction and negation alone, present first the forms of different connectives (as in Figure 4.5), and only then explain how they follow from the more basic definitions. In fact, Peirce himself has partly followed this path in his Lowell Lectures [Pei58, Vol. III], where he derived everything from the implication idiom.

2. Reformulating the permissions. Shin suggests an equivalent phrasing of the permissions, especially the iteration/deiteration permission, which is the least intuitive of Peirce's permissions. The important qualities of this reformulation are:

    - The new phrasing separates more clearly what is allowed in negative contexts and what in positive contexts, limiting the (de-) iteration permission to cases which are not already permitted under (un-) scribing. In addition to its role in making the permission more intuitive, this disambiguates some permitted steps: For example, in the original formulation, iterating into a negative context is allowed by two independent permissions; with the new one, this is no longer the case.

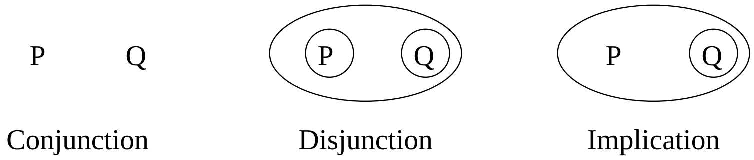


Figure 4.5: Beta idioms for the different connectives

- The new phrasing only allows iteration and deiteration with respect to immediately containing areas, localizing the permission's effects.

These modifications of the presentation of Beta, or others in their spirit, should be considered before turning the system to educational use.

We should also note that in spite of its many advantages, Beta is not perfect: For example, regardless of Shin's modifications, it is not very good in visualizing the scope of quantifiers, and in particular the interaction between quantifiers and other connectives. Still, it is quite good, and may serve as a basis for improvements and extensions.[4]

Of course, a system which only represents formulae will not be very helpful in teaching their semantics, and the diagrammatic systems of Tarski's World would serve much better there.

4.5 The EGG

The EGG (Existential Graphs GUI) is a project that aims to provide a teaching tool that combines Hyperproof's innovative methodology with Existential Graphs. At the time of writing of this paper, a prototype implementing Alpha has been produced. The system supports a 'premise mode' where the diagram may be manipulated freely, a 'proof mode' which only allows permissible actions (i.e. valid inferences), and an 'assisted mode' which explicitly offers the user valid operations. Figure 4.6 shows the program running in assisted mode, and the six permitted operations for Alpha graphs are available to the user as the buttons between the eraser and the check-mark button. The regular editing operations (delete, copy, and paste), as well as adding a sentence and adding a cut, are always available, but not always permissible. When the program is in proof mode (when the check-mark button is pressed down), only permissible modifications are allowed, and an attempt

[4] The most famous such extension — probably more famous than Peirce's own — is John Sowa's *Conceptual Graphs*, see e.g. [Sow84].

Figure 4.6: The EGG - sample screen snapshot

to perform a non-permissible operation produces an error message. Another 'educational' feature of the system (turned on in figure 4.6) is an option to display negative contexts in inverse video. The system supports saving and loading of diagrams so that problem diagrams may be prepared by a teacher and retrieved by students. The prototype does not yet support the actual posing of questions, the saving and retrieval of proofs, the logging of user actions, and other useful features of Hyperproof as a teaching and research system; these are planned for future versions. Another important feature to implement, in view of all the discussion above, is two-way translation between diagrams and FOL sentences. The Beta system, which will also provide these functionalities, is currently in design stages.

4.6 Conclusion

I hope to draw attention to some basic issues in the research of visual and diagrammatic reasoning, and to contribute to their resolution. However, I do not see the work reported here as some sort of 'final answer' to the problems discussed: The interpretation of the results of prospective research with the EGG may still be questioned. For example, while I hold my claim that the EGG visualizes logic, like Turing's World visualizes Turing Machines, and that Beta is a good visualization of the syntactic structures involved, it is far from clear that this visualization is as natural to logic as state diagrams are to Turing Machines. For example, Beta's difficulties with quantifiers may prove more important and harder to solve than they seem.

Still, I believe that by pointing at the pitfalls of research with Hyperproof and showing another way, I make a fruitful contribution to the discussion. And also, there are students of logic to consider: The studies of the group at HCRC show that for some of them, Hyperproof is not helpful; a different visual approach may prove beneficial to these students. Further, it is my intuition that while Tarski's World is a great tool for teaching the semantics of First-Order Logic, Hyperproof is not as helpful for teaching basic proof theory. I believe the EGG might be more helpful there.

Which also brings me back to topics discussed earlier in this thesis: How does all this talk about non-model-based diagrammatic reasoning relate to the direct-mapping approach? Can it be reconciled? Well, as mentioned above, Existential Graphs may be analyzed in a direct-mapping approach, except that the models they map into are linguistic models, meta-models if you like, and not object-level models. Symbolic reasoning is manipulation of structures — the structures which are formulas; Existential Graphs may make this easier to see.

There is a lot of work — both practical and theoretical — to be done before the EGG can be presented to teachers of logic and researchers of human reasoning; I believe, however, that when it hatches, it may contribute significantly to both communities.

I would like to end this chapter with the disagreement between my ideas about the EGG and Peirce's ideas about Existential Graphs. Peirce says,

> "First, although the study of it and practice with it will be highly useful in helping to train the mind to accurate thinking, still that consideration has not had any influence in determining the characters of the signs employed; and an exposition of it, which should have that aim, ought to be based upon psychological researches of which it is impossible here to take account." [Pei58, vol. IV, p. 424]

The EGG, of course, aims to aid both in psychological researches and in training the mind, in a feedback loop between the two uses. And much more importantly, later in the same lecture, Peirce says,

> "Fourth, although there is a certain fascination about these graphs, and the way they work is pretty enough, yet the system is not intended for a plaything."

Well, the EGG most certainly is.

Bibliography

[BE99] Jon Barwise and John Etchemendy. Computers, visualization, and the nature of reasoning. Available through Hyperproof's web page, http://www-csli.stanford.edu/hp/CVandNR.pdf, 1999.

[Ber67] Jacques Bertin. *Sémiologie graphique: les diagrammes, les reśeaux, les cartes.* Mouton and Gauthier-Villars, The Hague and Paris, 2nd edition, 1967.

[BS95] Jon Barwise and Atsushi Shimojima. Surrogate reasoning. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, 2(4):7–27, November 1995.

[Car94] Lewis Carroll. *Alice's Adventures in Wonderland.* Project Gutenberg, 1994. Originally published 1865. ftp://metalab.unc.edu/pub/docs/books/gutenberg/etext91/alice30.txt.

[CMP91] Robin Cooper, Kuniaki Mukai, and John Perry, editors. *Situation Theory and Its Applications*, volume 1 of *CSLI Lecture Notes.* CSLI Publications, 1991.

[Das98] Mehdi Dastani. *Languages of Perception.* PhD thesis, ILLC, Universiteit van Amsterdam, Amsterdam, The Netherlands, 1998.

[Dav86] Donald Davidson. A nice derangement of epitaphs. In Ernest LePore, editor, *Truth and Interpretation: Perspectives on the Philosophy of Donald Davidson*, chapter 24, pages 433–446. Blackwell, Oxford, 1986.

[EF95] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1995.

[Eng98] Yuri Engelhardt. Meaningful space: How graphics use space to convey information. In *Vision Plus 4: The Republic of Information*, pages 108–126. Carnegie Melon University, School of Design & International Institute for Information Design, 1998.

[Gri75] H. Paul Grice. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 41–58. Academic Press, New York, 1975. Reprinted as Essay 2 in [Gri89].

[Gri89] H. Paul Grice. *Studies in the Way of Words.* Harvard University Press, Cambridge, MA, 1989.

[Gur98] Corin Gurr. Theories of visual and diagrammatic reasoning: Foundational issues. In G. Allwein, K. Marriot, and B. Meyer, editors, *Formalizing Reasoning with Visual and Diagrammatic Representations*, Fall Symposium, Orlando, Florida, October 1998. AAAI, AAAI Press. Online version at http://www.hcrc.ed.ac.uk/∼corin/gurr-frvdr98.ps.

[IK89] Neil Immerman and Dexter Kozen. Definability with bounded number of bound variables. *Information and Computation*, 83(2):121–139, nov 1989.

[IL90] Neil Immerman and Eric Lander. Describing graphs: A first-order approach to graph canonization. In Alan Selman, editor, *Complexity Theory Retrospective*, pages 59–81. Springer-Verlag, 1990.

[LS87] Jill Larkin and Herbert Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11:65–99, 1987.

[Luk80] Eugene Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. In *21st Annual Symposium on Foundations of Computer Science*, pages 42–49, Syracuse, New York, 1980. IEEE.

[Mar82] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman, San Francisco, 1982.

[MB95] B.T. Messmer and H. Bunke. Subgraph isomorphism in polynomial time. Technischer Bericht IAM 95-003, Institut für Informatik, Universität Bern, Schweiz, 1995.

[MR90] Joseph Marks and Ehud Reiter. Avoiding unwanted conversational implicatures in text and graphics. In Thomas Dietterich and William Swartout, editors, *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 450–456, Boston, 1990. AAAI, MIT Press.

[Neu93] Victoria E. Neufeldt, editor. *Webster's New World Dictionary of American English*. Simon & Schuster, New York, third edition, 1993. Fourth Printing, with corrections.

[Obe96] Jon Oberlander. Grice for graphics: pragmatic implicature in network diagrams. *Information Design Journal*, 8:163–179, 1996. Also available as ftp://ftp.cogsci.ed.ac.uk/pub/graphics/idj95.ps.

[OMC+99] J. Oberlander, P. Monaghan, R. Cox, K. Stenning, and R. Tobin. Unnatural language discourse: an empirical study of multimodal proof styles. *Journal of Logic, Language and Information*, 8(3):363–384, 1999. Available as http://www.hcrc.ed.ac.uk/~jon/papers/jolli99.ps.gz.

[Pei58] Charles S. Peirce. *The collected papers of Charles S. Peirce*. Belknap Press / Harvard University Press, Cambridge, MA, 1931–1958. Eight volumes, different editors.

[Pei76] Charles S. Peirce. *The New Elements of Mathematics*. Mouton, The Hague, 1976. Four volumes, edited by Carolyn Eisele.

[Pez99] Elena Pezzoli. Computational complexity of ehrenfreucht-fraïssé games on finite structures. In G. Gottlob, E. Grandjean, and K. Seyr, editors, *CSL '98*, number 1584 in LNCS, pages 159–170, Berlin Heidelberg, 1999. Springer-Verlag.

[Pin90] Steven Pinker. A theory of graph comprehension. In Roy Freedle, editor, *Artificial Intelligence and the Future of Testing*, chapter 4, pages 73–126. Lawrence Erlbaum, Hillsdale, New Jersey, 1990.

[Rob73] Don Roberts. *The Existential Graphs of Charles S. Peirce*. Mouton, The Hague, 1973.

[Shi99] Sun-Joo Shin. Reconstituting beta graphs into an efficacious system. *Journal of Logic, Language and Information*, 8(3):273–295, July 1999.

[SO95] Keith Stenning and Jon Oberlander. A cognitive theory of graphical and linguistic reasoning: logic and implementation. *Cognitive Science*, 19:97–140, 1995. Also available as ftp://ftp.cogsci.ed.ac.uk/pub/graphics/ctg.ps.

[Sow84] John Sowa. *Conceptual Structures: Information Processing in Mind and Machine.* The System Programming Series. Addison-Wesley, 1984.

[SteTA] Keith Stenning. Seeing reason: image and language in learning to think. Manuscript to be published, non-final version., TA.

[Tve97] Barbara Tversky. Cognitive principles of graphic displays. In Michael Anderson, editor, *Reasoning with Diagrammatic Representations II: Papers from the AAAI 1997 Fall Symposium*, pages 116–124. AAAI, 1997.

[Twy79] Michael Twyman. A schema for the study of graphic language (tutorial paper). In P.A. Kolers, M.E. Wrolstad, and H. Bouma, editors, *Processing of visible language: Volume 1*, pages 117–150. Plenum Press, New York, 1979.

[vBD83] Johan van Benthem and Kees Doets. Higher-order logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 1, chapter 4, pages 275–329. D. Reidel Publishing Company, 1983.

[vdDvL00] Jaap van der Does and Michiel van Lambalgen. A logic of vision. *Linguistics and Philosophy*, 23:1–92, 2000.

[Wan95] Dejuan Wang. *Studies on the Formal Semantics of Pictures.* PhD thesis, ILLC, Universiteit van Amsterdam, Amsterdam, The Netherlands, 1995.

[Was00] Renata Wassermann. *Resource Bounded Belief Revision.* PhD thesis, ILLC, Universiteit van Amsterdam, 2000.