Chapter 1

Diagrammatic Reasoning

This book is about computational models of reasoning involving diagrams. A diagram is a form of visual representation, a kind of picture, but unlike sketches, color drawings and paintings, that emphasize qualitative aspects of the represented objects, diagrams focus more on structural and schematic aspects of objects and spatial states of affairs, and are used mostly for analysis and problem solving. Diagrams are very ubiquitous forms of representation and are present in mathematics, logic, physics, engineering, architecture, urban planning, and many other scientific disciplines and human practices. This topic can be studied from a philosophical, psychological, design and computational perspectives, among others[1]. In the present text, the subject is addressed from a computational perspective, and the focus is on computational models of diagrammatic reasoning implemented as Artificial Intelligence (AI) programs: programs that use diagrams in reasoning and problem solving task. On of the main motivation of the present text is to understand better the sense in which a computer program can represent a diagram (i.e. the external representation on a piece of paper) and reason and solve problems using such representation.

As many objects of common experience, it is hard and perhaps not possible to provide a definition of what is a diagram in terms of necessary and sufficient conditions; diagrams are more like a family whose members are easily recognized when they are presented to us, and we introduce this kind of representations through a number of examples that can be regarded intuitively as "diagrams". The paradigmatic case of diagrammatic reasoning has

[1] An introductory survey of the issues involved in diagrammatic reasoning is provided in Chandrasekaran (1997). A more comprehensive view is the collection presented by Glasgow et al. (1995), and the subsequent conference and workshops Mainly the *Diagrams* conference, whose proceedings are published by Springer in the Lecture Notes on AI series, and also several AAAI and IJCAI workshops.

been with us since the Pythagoreans, whom already used diagrams to express and prove geometric and arithmetic theorems. The use of diagrams was also very influential in the history of mathematics, and the first body of mathematical knowledge was Euclid's Elements (Heath, 1956), where the axiomatic method of proof was first introduced and diagrams were essential to express, produce and understand the proofs.

As a first example of a diagrammatic reasoning process consider the following proof of the theorem of Pythagoras:

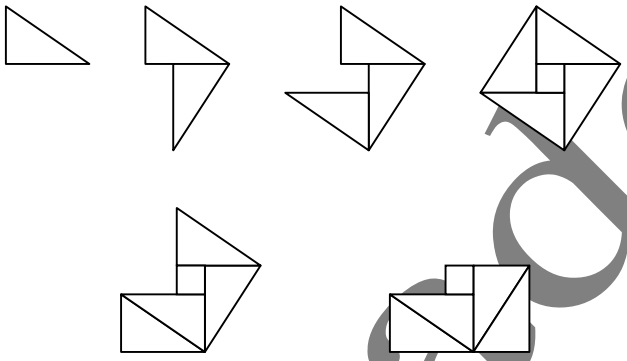

Figure 1.1 The Theorem of Pythagoras

This proof, presented by Bronowsky (1973), is based on an arbitrary right triangle, which is duplicated, rotated and translated three times, until a square on the hypotenuse of the right triangle emerges in the top-right diagram. In this figure there is also an emerging inner square whose side is the difference between the right sides of original triangle seed. In the bottom row, the left and right triangles of the upper part of the figure are rotated (counter clock-wise and clock-wise respectively) until a reflected L-shape figure appears in the bottom-right diagram. In this latter figure two adjacent squares can be visualized, one aligned to one right side of the seed right triangle and the other to the other right side of the seed. As the top-right and bottom-right figures have the same "tiles" and do not overlap, they also have the same area, so the theorem of Pythagoras holds.

Once the theorem has been "seen", the intuition that this is indeed a very general truth in geometry is very strong, and the diagrammatic sequence constitutes a diagrammatic proof.

The sequence relies on a particular seed right-triangle, with its particular size and orientation, and the place in which it is located in the plane; however, the intuition that the sequence is independent of this contingent choice is very strong, and that an equivalent sequence would have been produced if the seed with its size, orientation and position had been different. Although the diagrammatic argument is "pivoted" on a concrete object with contingent properties, the proof is so solid that its validity can hardly be denied.

This proof illustrates several aspects of diagrammatic reasoning, and next we focus on five of them: the first is the actual constructive process that generates the proof. Here the question is what kind of generative scheme is required to produce the diagrammatic sequence; how can the problem space be defined and what is its size; what kind of inference scheme can control and guide the construction to a happy result. After all, this and similar proofs have been produced by people with very limited computational resources, like a pencil and a piece of paper, and it should be possible to make explicit the underlying constructive process.

The second is what are the roles of reinterpretations and visualizations in the proof. Although the seed is a right triangle, two squares emerged in the two crucial states of the sequence, and their visualization is essential to realize the theorem and its proof. People can see the squares in the top-right diagram as soon as they appear, but the visualization of the two aligned squares in the right-bottom diagram is very hard, and yet it is the crucial inference that needs to be performed to realize the theorem and its proof. What kind of inference is this visualization and how can it be characterized are also questions that will be addressed by a theory of diagrammatic reasoning.

The third aspect is the machinery that is needed to represent the diagrammatic knowledge. How can the basic triangles and the emerging squares can be represented and referred to? What is the nature of this reference: it is a concrete reference to the overt symbols appearing in the diagram or it is rather a generic reference to the whole class of equivalent diagrams constituting the proof. We also need to discuss what is the relation between the concrete nature of diagrams as external representations and their interpretation as a general

or abstract class representing the concept expressed by the theorem. All these questions have to do with the properties of visual representations with their underlying interpretations, which have to be addressed in a theory of diagrammatic reasoning as well.

A fourth question is related to the concept of equality involved in the assertion of the theorem. How is it possible to assert that two configurations generated in a process of change have the same property. Are the triangles and squares in the different states of the sequence the same, despite that their geometric properties are different? How the identity of a geometric object is determined and preserved when the object undergoes a process of change. And also, how the equality between two general properties can be expressed and, furthermore, how can this be realized through a computational process. More generally, how can we reason about geometric change involving abstractions so effectively?

The fifth question arising from this proof is how the diagram, the geometric shape or form on a piece of paper, is related to its interpretation in an arbitrary conceptual domain. Although the Theorem of Pythagoras is a geometry theorem establishing a relation between areas of squares, for instance, its importance relies to a great extent in its interpretation into the arithmetic as the well-know formula $h^2 = a^2 + b^2$ that permits to measure distances. However, this latter expression is not an arithmetic theorem, as there are an infinite number of triplets (h, a, b) that do not satisfy the relation, and indeed the expression is only true when the squares numbers represent the areas of the squares on the three sides of the same right-triangle. Hence, the arithmetic expression is only true under a representational mapping between the arithmetic and the geometry. Here, the questions of how the arithmetic expression is produced and how the representational mapping is established need to be answered. More generally, diagrams are commonly used to reason not about themselves but about their interpretation in other knowledge domains; mapping the domain concepts into the geometry facilitates greatly the reasoning process, and this is one reason why diagrams are so useful and effective representation and problem-solving devices. A theory of diagrammatic reasoning should also show how these kinds of representational mappings are established and used in reasoning and problem solving.

The Theorem of Pythagoras is a very interesting and challenging case study in diagrammatic reasoning; despite its simplicity and fundamental role in mathematics, and the fact that it is the main achievement in the second book of Euclid's geometry as proposition 47 (Heath, 1956), there seems to be no proof of it derived from Hilbert's axioms of geometry by a strict formal derivation through valid inference steps, within his program for the formalization of mathematics. Also, despite the effort in theorem proving in AI since the origins of this discipline in the 50s of the last century, there is not a theorem proving system capable to produce a fully automatic proof of the theorem of Pythagoras to the present date. The theorem is not even mentioned in the literature related to the first AI geometric theorem proving system developed by Gelenter in the late 50s (Gelernter, 1995). Later on, Pineda (1989) discussed the need to model visualizations and reinterpretations to carry on with this kind of proofs, and Barwise and Etchemendy used it illustrate heterogeneous reasoning (1990); the theorem was also discussed by Wang to illustrate the need to use generic descriptions in diagrammatic reasoning (1995), Jamnik used it to illustrate a taxonomy of diagrammatic theorems (1999) and Lindsay (1998) presented a demonstrator system that can verify different proofs of the theorem through constraint satisfaction; however, no formal proof procedure or theorem-proving system was offered in any of these investigations and non of these studies or systems provided a fully automatic proof of this theorem. This brings about the question of why diagrams are needed and what it the job that they do. This is one of the main concerns discussed in the present book.

As a second example of diagrammatic reasoning consider an arithmetic theorem stating that the sum of n odd numbers is equal to the square of n; this is, $1 + 3 + \ldots + 2n - 1 = n^2$. This theorem has a diagrammatic representation, as shown in Figure 1.2. This theorem is also due to the Pythagoreans who noticed that an odd number could be represented by an inverted *L*-shape, which they designed as a "gnomon"[2].

[2] "… But the ancient commentaries on the passage make the matter clearer still. Philoponus says: 'As a proof… the Pythagorean refer to what happens with the addition of numbers; for when the odd numbers are successively added to a square number they keep it square and equilateral… Odd numbers are accordingly called *gnomons* because, when added to what are already squares, they preserve the square form… Alexander has excellently said in explanation that the phrase 'when

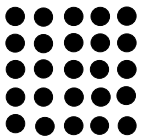

Figure 1.2 Theorem of the sum of the odds.

As can be seen in the figure, a square of side $n + 1$, conformed by $(n + 1)^2$ dots, can be construed as the union of a square of side n and a gnomon of side $n + 1$ with $2(n + 1) - 1$ dots, for an arbitrary parameter n. So, the union of n consecutive gnomons forms a square of area n^2.

The particular diagram in Figure 1.2 stands for a concrete instance of this theorem, and asserts that the union of the fourth square in the series with the fifth gnomon results in the fifth square. However, if a person sees an instance of the theorem, like Figure 1.2, and constructs the corresponding concept in his or her mind, and knows how to apply it, then the diagram is a representation of the theorem as a whole. In the same way that a single diagrammatic sequence in the case of the Theorem of Pythagoras stands for the general case, any particular instance of the diagram stands for the general case too. Here again, the visualization and generalization of the figure are very direct inferences, which once performed, the intuition that the theorem holds is very strong.

These examples illustrates that one aspect of diagrammatic reasoning is that interpreting a diagram or a diagrammatic sequence is also constructing a concept in the mind of the interpreter. In addition, the concept is produced in a way that its truth can hardly be denied. So, one main goal of a diagrammatic reasoning theory is to model the synthesis of such concepts. Of course, we do not know what is the representational format of the human

---

gnomons are placed round' means *making a figure* with the odd numbers… for it is practice with the Pythagoreans to *represent things in figure*' " (Heath, 1956, p. 359).

mind, and cannot represent the concepts expressed by diagrams in such unknown format, but we follow the AI working hypothesis that concepts can be represented in computers, and as the objects of computation are mathematical functions (i.e., Boolos and Jeffrey, 1989), we represent concepts through mathematical functions.

For instance, the interpretation of the diagram in Figure 1.2 is an inductive concept represented by the recursive geometric function $sq(0) = 0$ and $sq(n + 1) = sq(n) + 2(n + 1) - 1$ for $n \geq 0$, where the term $sq(n)$ corresponds to the square n and the term $2(n + 1) - 1$ corresponds to the gnomon $n + 1$; so, for $n = 0$, $sq(1) = 1$; for $n = 1$, $sq(2) = 4$, etc. The base of the induction rests on the observation that a gnomon of size 1 is also a square of size 1 (for the parameter $n = 1$), which is formed through the union of a square of size 0 and the gnomon of side (and size) 1.

Hence one challenge in diagrammatic reasoning is to produce this kind of functions out of the diagrams, and to identify the additional conceptual machinery involved in the synthetic process. It is also required to make sure that the function does represent the intended concept precisely, and that the function is correct. The intuition is that these functions and their properties can be derived from the diagrams in simply and straightforwardly, as least for people, but the computational process should be also simple. If the theorems are not known beforehand, diagrammatic reasoning is also a creative and discovery process.

The recursive function above represents a geometric concept and the corresponding arithmetic theorem results from a mapping under which geometric squares represent square numbers and gnomons represent odd numbers. However, unlike the arithmetic expression of the Theorem of Pythagoras, which is only a contingent truth, the arithmetic interpretation of this theorem is also an arithmetic theorem, which can be proved by mathematical induction. So, there are two radically different proofs of this theorem: the diagrammatic one, that has a synthetic character, and the arithmetic one, that can be derived from arithmetic axioms and has an analytical character.

However, the theorem of Pythagoras and the theorem of the sum of the gnomons are very similar in many respects: both belong to the domain of the geometry and assert a relation between the areas, and the underlying processes of reasoning about the conservation of areas in different stages of a diagrammatic sequence or a diagrammatic interpretation state, the essential property stated by the theorems, seems to be quite alike; also, in both cases, geometric squares are interpreted as squares numbers, and the union of areas is interpreted as the arithmetic sum; all of this suggests that a theory of diagrammatic reasoning should be able to provide an account of both of these theorems within the same conceptual framework.

As a third instance of a diagrammatic reasoning process consider the diagram in Figure 1.3, where a square is decomposed into its lower half rectangle and its upper two quadrants, but in addition, its top-right quadrant underlies the same decomposition process. This decomposition pattern can performed a finite number of steps, as shown in the diagram, but it can also carried out at infinitum, and the recurrent pattern can be captured through a geometric induction on the embedded squares.

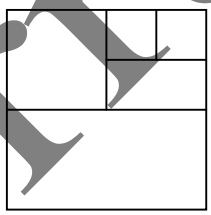

Figure 1.3 Induction decomposition of a square

As in the previous example, this diagram represents an arithmetic theorem, which in this case is the infinite series:

(1) $$\sum_{n=1}^{\infty} 1/2^n = 1$$

The truth of this theorem can be visualized by decomposing the area of the base square (normalized to a square of size 1) as the fraction corresponding to the lower half and the two upper quarters: $1/2^1 + 1/2^2 + 1/2^2 = 1$, and observing that the last quarter can be decomposed similarly as $(1/2^1 + 1/2^2 + 1/2^2) \times 1/2^2$, which is $1/2^3 + 1/2^4 + 1/2^4$, so the basic

sum is also decomposed as $1/2^1 + 1/2^2 + 1/2^3 + 1/2^4 + 1/2^4 = 1$.This process can be carried out at infinitum and the decomposition represents the infinite series $1/2^1 + 1/2^2 + \ldots + 1/2^n = 1$, so the theorem holds.

According to the discussion above, the diagram expresses a concept, which can also be represented by a mathematical function. The structure of this theorem is similar to the theorem of the sum of the odds, and as a first approximation it can be captured by an analogous the recursive function $s(1) = 0$ and $s(n + 1) = s(n) + 1/2^n$ for $n \geq 1$, which converges to 1 in the infinite. However, does this function really capture the sum in (1) or the infinite diagrammatic induction in Figure 1.3? This function can be used to compute the series, which approximates very quickly to the value of 1, but it does not seem to account for the fact that the computation only converges to this value when the parameter n is infinite. So, the present function does not capture the concept fully, and expresses less than the diagram and also than the theorem in (1). A theory of diagrammatic reasoning should provide the function representing the theorem precisely, and the construction method should guarantee that the function is indeed correct. Furthermore, the basic mechanisms through which the theorem of Pythagoras and the theorem of the sum of the odds should also be relevant to this latter theorem.

Another aspect of diagrammatic reasoning is illustrated in Figure 1.4: the same diagram under a different but very similar decomposition expresses a different concept, and represents a different arithmetic theorem. Consider that the original square is decomposed instead into the union of a *L*-shape, formed by the quadrants in the lower row and the top-left quadrant, and the top-right quadrant directly.

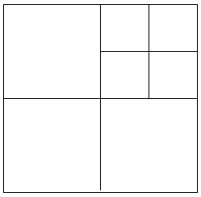

Figure 1.4 An induction on the upper right corner

This diagram represents the theorem:

(2) $$\sum_{n=1}^{\infty} 3/2^{2n} = 1$$

In this case, the area of the basic square is $3/2^2 + 1/2^2 = 1$. The first term corresponds to the two lower and the upper left quadrants, and the second to upper right quadrant. As before, the squares in this latter quadrant is decomposed by the recurrent pattern, as $(3/2^2 + 1/2^2) \times 1/2^2$, which is $3/2^4 + 1/2^4$, so the basic sum is also decomposed as $3/2^2 + 3/2^4 + 1/2^4 = 1$. The last term can undergo the same decomposition at infinitum and the diagram represents the theorem $3/2^2 + 3/2^4 + \ldots + 3/2^{2n} = 1$.

As before, the geometric pattern can be represented through a recursive function, which at a first approximation is $s(0) = 0$ and $s(n + 1) = s(n) + 3/2^{2n}$ for $n \geq 1$, where the term $s(n)$ represents the *n-th L*-shape and the term $3/2^{2n}$ represents the *L*-shape of the top-right quadrant in the same level. So, for $n = 1$, $s(2) = 0 + 3/2^2$; for $n = 2$, $s(3) = 3/4 + 3/2^4 = 15/16$, $s(4) = 63/64$, etc. The function captures the accumulative sum, but as before it misses an important aspect of the theorem expressed by the diagram and the formula in (2); the missing property is again the conservation of area in every decomposition step, and also the fact that the series converges in the infinite to 1. The study of the underlying structure of this kind of theorems should produce the function representing the missing information, and the synthetic procedure should guarantee that the function is indeed correct.

It has been argued by Jamnik (1999) and also by Foo (1999) that the first three theorems presented above are representative of three different classes of diagrammatic theorems: standard Euclidean proofs, exemplified by the theorem of Pythagoras (although using a different proof), theorems involving a finite induction on one parameter, exemplified with the theorem of the sum of the odds, and theorems involving an infinite recurrent patters, as theorems (1) and (2). Jamnik (1999) also presents a theory to model the reasoning process of theorems of the second kind (the sum of the odds), and Foo (1999) discusses theorem (1). However, although each example illustrates a different aspect of diagrammatic reasoning, there are several aspects the reasoning process that are common in all three examples; in particular, all five questions that were posed for the case of the theorem of Pythagoras are also relevant in the other cases, and the three examples involving an infinite diagrammatic recurrent pattern are very similar. So, a theory of diagrammatic reasoning

should be able to explain all three cases and their variants with the same underlying machinery.

Diagrams are also representational devices to support logical thinking supported through diagrams, like Euler Circles or Venn Diagrams. This kind of reasoning has been the subject of a considerable amount of work (e.g. Shin, 1995). In this latter case the inclusion, intersection and disjointness of diagrammatic objects represent the corresponding set relations. This kind of representation is used to support valid reasoning quite straightforwardly. For instance, in syllogistic reasoning each premise can be represented by a diagram, and the conclusion by their superposition. This is illustrated in Figure 1.5 where a diagram represents the premise *All A are B*, another *All B are C*, and the conclusion can be read directly from their superposition. Here again, concrete diagrammatic objects, the circles in this case, represent general classes, and the geometrical relations between them represent the general relation between classes. This kind of representation faces also the problem of representing logical negation, where marks like crosses or textures may be interpreted as stating that corresponding region has an empty extension.

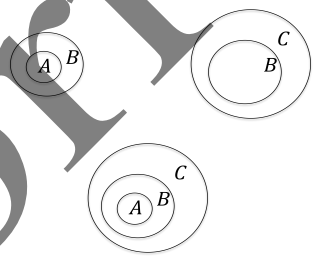

Figure 1.5 Diagrammatic Syllogistic Reasoning

Unlike the previous examples, Figure 1.5 includes a number of textual labels to mark the intended interpretation. This is also a common feature of diagrams. The texts are used here to emphasize the identity of the diagrammatic symbols, but they would not be required if the corresponding circles in the three figures were considered the same. In the three previous examples, labels were not used because the relevant identity relations are suggested by the diagrammatic context, but we could also add them, as it is often done. However, the underlying question in relation to the labels in a theory of diagrammatic

reasoning is how the identity of symbols is established in a diagrammatic sequence or in different diagrammatic interpretation states, and how such identity relation is used in the reasoning process. When this is clear the labels can be used to highlight the identity relations, and to name properties and relations, although they are somehow subsidiary facilities for the communication process.

In all previous examples a diagram is interpreted as an arithmetic or logical theorem, and the reasoning involved has a formal character in the sense that the theorem follows necessarily from the premises, and diagrams involve "valid reasoning" in a strong logical or mathematical sense. However, unlike logical schemes involving deductive inference, that has an analytical character, diagrammatic reasoning seems to have a strong synthetic orientation involving a visual constructive process which guaranties implicitly the "validity" of the solution. In these kinds of scenarios "the conclusion" is not a logical consequence of "the premises" and the problem can be best thought as a case of pragmatic inference. Furthermore, there are settings involving no interpretation of the diagram into a conceptual domain, and the reasoning task has a pure spatial character. These latter properties (or lack of properties) are illustrated, for instance, by the kind of diagrams used in intelligence tests, like Raven's Progressive Matrices for non-verbal reasoning (Raven & Raven, 2008), where diagrams are used to express and solve problems through "visual thinking", as illustrated in Figure 1.6.

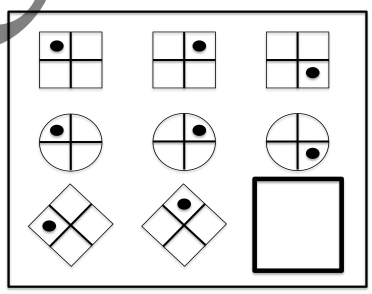

Figure 1.6 Visual Thinking

The solution of this kind of problems seem to depend on the ability of constructing a spatial abstraction from the example diagrams, that needs to be satisfied by the solution, which is

also expressed as a diagram directly. In Figure 1.6, the top row is constituted by a sequence of three up-right squares divided in their quadrants and there is a dot in a quadrant in each square, which shifts in a clock-wise direction from quadrant to quadrant along the sequence. The same pattern is presented in the second row, but with circles instead of squares, and with circle slides instead of quadrants, and the lower row repeats the pattern on the upper row but with the squares tilted, but with its last figure missing, and the solution of the problem consists on synthesizing this last figure. The first and second columns conform also to a pattern, and the third one should also conform to such pattern too, when it is completed with the synthesis of the bottom figure, which is also the solution of the problem. In addition the solutions for rows and columns must be the same, constraining further the properties of the solution.

One way to think about this problem is by "reading" the rows and columns as descriptions, abstracting over shapes (i.e., squares and circles) and orientations (i.e., upright and tilted) and the position of the dot (i.e., static, shifted). The hard part of solving the problem seems to be to induce such descriptions from the visual information, as the construction or verification of the solution seems to be a direct inference once such descriptions are in place. This suggests that a theory of diagrammatic reasoning should also explain how this kind of spatial abstractions can be constructed and applied in problem solving, and how the space is represented and reasoned about in a more informal and qualitative setting. These kinds of problems have been modeled in analogical reasoning and inductive structural learning in AI (e.g., Levet et al, 2007, 2010; Hegmann, 2011, among others), and illustrate that diagrammatic reasoning involves not only deductive, but also inductive and abductive inferences strategies.

A theory of diagrammatic reasoning should also explain other kinds of reasoning schemes supported by diagrams. Tables or tabular representations, for instance, are very effective devices to structure and present relational information, and relational databases are tabular structures that are used very efficiently for storing and accessing larges amounts of information, and can be thought of as diagrammatic representations too. Visual languages in which a vocabulary of symbols with a set of geometric properties and relations are given

conventional interpretations for expressing knowledge in a direct and efficient way are also diagrammatic forms of representations, and should be studied within the same conceptual and methodological framework.

This brief summary illustrates a range of phenomena and problems posed by the use of diagrams in inference and problem solving. The present text is an attempt to gain a deeper understanding of these problems from the AI perspective, but taken into account relevant insights from diverse related disciplines, like computer science, philosophy, psychology and even some aspects about the representation of images that come from the neurosciences. The present reflection is presented from the perspective of a computational theory of diagrammatic reasoning that has been developed over the last years, and a computer program named *Pitágoras* that implements the representational and inferential machinery postulated by the theory (Pineda, 2007), and also from the perspective of a system for the diagrammatic representation of functions and abstractions, the system *F* (Pineda, 2011), which illustrate diagrammatic reasoning in which a memory buffer represents the diagram, and inference corresponds to operations performed on the buffer directly. These and a number of case studies are presented in detail along the text. In the present theory the computational objects correspond directly to the diagrammatic objects with their interpretations, and the inferences are quite direct and natural, reflecting why diagrammatic inference seem to be "easy" and useful in learning and problem solving.

One strong intuition underlying much work in diagrammatic reasoning is that the process has a visual component, which is central to the task, and that it is a case of "visual thinking". This in turn suggests that diagrams, the external representations, are represented "internally" as images in the mind, which are the objects of thought directly. However, images are only accessible through introspection, and the very idea that there are "images" was strongly challenged early in the XX Century in analytical philosophy and positivism, in psychological behaviorism, and later on in the original presentation of AI, which had a strong "mental" and "symbolic" orientation (Turing, 1950); in these quarters it was held that knowledge has to be expressed through descriptions of a rather linguistic character, and this became a central tenet in knowledge representation in AI. However, the appearance of

cognitive psychology and the discovery of imagistic phenomena, like mental rotation (Shepard and Metzler, 1971; Shepard and Cooper, 1982) and Kosslyn's imagery program (1983), permitted to articulate a new notion of image that can be the subject of empirical research, and has been extensively investigated in cognitive psychology, neuropsychology and the neurosciences (Kosslyn, 2006). The opposition between the descriptionalists and the imagery views gave rise to the so-called "imagery debate" (Tye, 1991). The nature of this debate and its implications for diagrammatic reasoning is the subject of Chapter 2. There it is argued that there is indeed a place for images and visual thinking in diagrammatic reasoning and knowledge representation more generally.

The imagery debate has also a counterpart debate in AI, where there is an opposition between propositional or Fregean representations, that correspond to descriptions, versus analogical representations, which correspond roughly to images. This is the subject of Chapter 3. There, the difference between the two formats is reviewed from a computational perspective. It is shown that analogs, which are also called "direct representations", support "direct inference", in which the conclusion is "read" directly from the representation, as opposed to the corresponding inference in descriptions or logical formats, where the conclusion follows from the application of a valid inference scheme. It is also illustrated that in diagrammatic inference, questions about content are translated into questions about the representational format, and that direct inference takes advantage of the spatial properties of the format to produce the answers quite efficiently. In this chapter it is also discussed whether diagrammatic information can be expressed to descriptions, and what conditions should be meet to carry on with the translation. The discussion is also placed in relation to a hierarchy of levels of representation proposed by Newell (1981), and it is argued that although diagrams and descriptions are different representational formats, both belong to the plane of expression at the so-called symbol level, and that the knowledge that is expressed by the two kind of symbolic structures has an equal status at the knowledge level, which corresponds to the interpretation of symbolic structures in people's heads.

The relation between the two representational formats is further discussed in Chapter 4, where the issue of the identity of diagrammatic symbols and their depictions in

diagrammatic sequences is analyzed. It is first argued that diagrammatic symbols that bear the same depiction can have different geometrical properties in different diagrammatic states, and that diagrams need to be thought of as intensional representation, as opposed to extensional systems that obey Leibniz Law. Intensionality in diagrams comes from the separation of the issues related to the objects identity from the issues related to their geometric properties, and also from the knowledge of the space that is represented through geometric algorithms. In this setting a diagrammatic sequence is thought of as a single diagram at the intensional level, where each diagram in the sequence corresponds to a different extensional state. The distinction between the intensional and extensional levels permits to model a class of diagrammatic constraint satisfaction problems by direct interpretation process, instead of complex numerical computations traditionally used to solve this kind of problems. This is illustrated with a well-known example of constraint satisfaction in computer graphics. It is also shown that the addition and deletion of diagrammatic symbols in diagrammatic sequences produces new diagrams, and a diagrammatic sequence is thought of as a sequence of intensional objects, where each object can have several extensional states. The intensionality of diagrams is also supported by reinterpretations and visualizations, and it is shown that an image corresponds to an interpretation of the overt extensional information from a particular perceptual perspective that is represented through intensional descriptions, and that the reinterpretation of a particular diagram, like the bottom right diagram in Figure 1.1 renders a different diagram. Diagrammatic sequences and reinterpretations interact in complex ways in diagrammatic reasoning, and reinterpretations define a parallel plane to the overt diagrammatic sequence, and it is within this plane were novel an interesting relations occur, like the reinterpretations of the triangles and squares in the diagrammatic sequence expressing the Pythagorean relation. This chapter is concluded with the notion of “image” that emerges from the discussion: a dual object that, on the one hand, is imprinted in a buffer, which contains extensional information, but on the other, is interpreted in relation to an intensional perspective, which is represented through intensional descriptions. In this sense, the image is an intensional object right at the interface between perception and thought. This notion of image offers a perspective to the imagery debate, and also to the opposition

between propositions and direct representations in AI, and also permits to integrate much work developed within diagrammatic reasoning in AI in a coherent way.

The discussion in chapters 2 to 4 provides a summary of issues that arise in diagrammatic reasoning, form different disciplines and methodological frameworks, and provides a wider understanding of the underlying issues that arise in diagrammatic reasoning. On the basis of these considerations, a model for diagrammatic reasoning is introduced in Chapter 5. In this model, a diagrammatic reasoning system (DRS) is defined as a 4-tuple $<D, I, \Phi, \Omega>$, where D is a diagram or a diagrammatic sequence, the external representation proper, I its interpretation, Φ a representation relation that associates diagrammatic structures in D with their corresponding interpretations in I, and Ω is a set of operations on diagrams. The nature of the representation relation is discussed in detail. In particular, every entry in the representation relation is designated as a "representational key" that states the interpretation of diagrammatic symbols, properties and relations in the intended interpretation domain. For instance, a representational key in relation the diagrammatic sequence in Figure 1.1 states that squares in the diagram are interpreted as squares numbers in the arithmetic and that the geometric union between areas is interpreted as the arithmetic addition. A derived representation key states that the geometric proposition "the area of a square on the hypotenuse of a right triangle is the same as the union of the areas of the squares on its right sides" is interpreted as the famous arithmetic expression $h^2 = a^2 + b^2$. Interpretations are computed compositionality by an *interpretation function* that maps diagrammatic structures into expressions representing their interpretations in relation to the representation relation. The model permits to define the expressivity of a diagrammatic system as the totality of propositions expressed by a diagram in relation to the interpretation function and the representation relation. An inverse interpretation function that assesses whether a proposition is expressed by the diagram is also defined, and the system supports direct inference through this latter device. The DRS model is related to the notion of image developed in Chapter 4, and when these are taking together, it is possible to identify three main settings of diagrammatic reasoning systems in AI, which are named systems Type-1, Type-2 and Type-3, depending on whether the diagram is represented computationally through descriptions, though a memory buffer or both respectively, and some representative

systems of each kind are briefly discussed. The model is illustrated with a diagrammatic system called *Graflog* for dynamic definition of visual languages (Pineda, 1989), which is a system of Type-1.

Systems of Type-2 are illustrated in Chapter 6 with the system *F* for the diagrammatic representation of mathematical functions and abstractions with finite domain and range (Pineda, 2011) where abstractions defined as sets of functions sharing a set of values for some of their arguments, and the representation captures such internal relation. In this system the diagram *D* is a table in which horizontal and vertical dimensions corresponding to the function's and abstraction's domain and range respectively, the Interpretation *I* corresponds to the knowledge of such mathematical objects, the representation relation Φ establishes that marks in the table's cells correspond to the assignments of values to the arguments in functions and abstractions, and the operations in the set Ω are functional abstraction, that produce composite abstractions, and reduction, that decompose abstractions into their constituent parts. The degree of structure of an abstraction or its informative content is captured with a definition of abstraction's entropy, which is also given. The representational format permits to express incomplete information and constraints, and the definitions of the abstraction and reduction operations with the corresponding entropy are extended for this latter case. The system is illustrated with the representation of a conceptual hierarchy including incomplete information and constraints, where generalizations and exceptions are captured in a simple and systematic way. The system suggests that extending a representation with a new proposition is a case of abstraction, as an alternative to logical formulations using non-monotonic deduction, in the tradition of default logics started by Reiter (1980). The representation can also be viewed as an associative memory that is accessed or indexed by its own content (e.g., Kosslyn, 2006, pp. 46), where the abstraction operation corresponds to memory register and reduction to memory recall. The system is also analogous the λ-calculus for the representation of functions, where abstractions in the system *F* correspond to λ-expressions, and abstraction and reduction correspond to λ-abstraction and β-reduction. The discussion suggests that there is a trade-off between the expressivity of the representational system and the extent to which abstraction and reduction are reversible; also the entropy of an abstraction is related

to the computational cost of inference, as abstractions with low entropy capture the internal relations between the objects included in the abstraction, and the cost of inference is reduced accordingly. The system also suggests that the use of good abstractions facilitates inference and problem solving, in opposition to the so-called knowledge representation trade-off, which states that abstract thinking involves highly expressible representational language, but the cost of inference goes in hand with the expressivity of the representation (Levesque and Brachman, 1985), and there is a compromise between expressivity and tractability. However, this opposition is dissolved when the knowledge representation trade-off is placed in relation to the more fundamental computational trade-off between expressiveness and reversibility. The knowledge representation trade-off is defined in relation to the λ-calculus and logical languages, which stand at a particular position of the trade off between expressiveness and reversibility, where information is preserved, abstraction and reduction are reversible, but there is a limitation in the expression of incomplete information and the interaction between the objects in abstraction. The λ-calculus and Turing Machines are equivalent, as well as to other representational formats like recursive functions and abacus computable function (Bools and Jeffreys, 1989), which stand at the same position in this latter trade-off. However, the system *F* shows that there are alternative positions in this trade-off with interesting properties and applications.

In Chaper 7 a model for heterogeneous reasoning with diagrammatic and propositional representations is presented. This extends the basic DRS system model with a logical theory *T* that enriches the interpretation *I* of the diagram *D* in relation to an interpretation relation Φ. The extension is illustrated with the system *Graflog* too, in which knowledge could be expressed through language in addition to the diagram. In this setting, direct inference is embedded within logical deduction, and the system supported a form of direct inference in a simple and systematic way. The model is illustrated further with a heterogeneous reasoning problem presented within the context of the *Hyperproof* program (Barwise and Etchemendy, 1994). Unlike the basic model where the representation relation Φ is fully specified, and direct inference is supported directly, this latter kind of problems underspecified Φ and both linguistic and diagrammatic knowledge can be under specified too, and this is a case of reasoning with incomplete information and constraints too. Hence,

the solution focuses in the determination of the representation relation using diagrammatic and linguistic knowledge, as once this relation is available the original problem is solved through direct inference. This is achieved by model construction through abstraction operations using the system *F*. The use of both descriptions and a memory buffer into an integrated reasoning systems is already a case of a system Type-3, which incorporates the properties of systems Type-1 and Type-2. The present case study suggests the architecture of a system of Type-3 that could solve heterogeneous reasoning problems in a fully automatic way. Finally, the present discussion illustrates the interweaved interaction between memory inference and logical inference, and provides additional evidence for the trade-off between expressiveness and the reversibility of abstraction and reduction.

In Chapter 8 the question of whether diagrammatic representations can express unrestricted abstractions like theorems and proofs is explored further. For this the concept of "abstraction" is discussed from different perspectives, and an operational criterion to state the expressiveness of a representation is presented, along the lines of the theory of the graphical specificity (Stenning and Oberlander, 1995). According to this theory the expressiveness of a representational system depends on the form of the representational keys included in the representation relation. The theory distinguishes three main kinds of systems in relation to the amount of abstraction that can be expressed, which are called Minimal Abstraction, Limited Abstraction or Unlimited Abstraction Representational Systems (MARS, LARS and UARS respectively). This theory also states that diagrams, and graphical representations more generally are LARS, with the consequence that diagrams cannot express theorems and proofs after all. However, the theory of graphical specificity is restricted to extensional systems, and tthe space of abstractions is much larger when intensional representations are considered. Hence, intensional information is included in the interpretation function and the representation relation too. The analysis of the representational keys used in *Graflog*, *Hyperproof* and the system *F* shows that these systems are indeed LARS. The chapter is concluded with a discussion on the conditions that have to be met for a DRS to be a UARS. This is the case if the diagram is interpreted as a generic geometric object, which is further interpreted as a generic description in the application domain, where the relation between both generic objects is stated as a

representational keys. If all representational keys of a DRS meet such condition the system is a UARS.

The machinery to express generic diagrammatic descriptions is introduced in Chapter 9. In this chapter a geometric representational language for the representation of basic and emergent diagrammatic objects, as well as its interpreter program, is presented (Pineda, 2007). The language is defined in terms of a signature of geometric symbols of different types, with their associated constructor and selector operators, and diagrammatic objects are represented through geometric abstract data types. In this language all predicate and function symbols have an associated geometric algorithm that computes directly the corresponding geometric property or relation. The system is intensional and a diagrammatic sequence is a sequence intensional objects, where each object can have several extensional states. In this way, the identity of diagrammatic objects is preserved along a diagrammatic proof. This basic machinery permits the representation of basic diagrammatic symbols and configurations. The language also includes the functional abstraction and functional application operators, permitting the expression of geometric abstractions or geometric concepts, which are represented as geometric functions. The application of these functions to geometric configurations of the appropriate types through the geometric interpreter renders whether such configurations are within the extension of the corresponding concepts. The language also includes a geometric description operator that permits the representation of emergent objects in relation to the geometric contexts in which these emerge, and geometric contexts are in turn represented through a geometric Boolean function. The geometric description operator permits to represent, for instance, the generic geometric squares that emerge in the Pythagorean proof. The theorem of Pythagoras is represented as a geometric function, and its application to three geometric squares and a right triangle is true if the squares conform to the Pythagorean relation and false otherwise; this is, the function representing the concept of the theorem computes whether an arbitrary configuration is included within its extension, and this corresponds to the knowledge that one has when he or she knows the theorem. Finally, the geometric representational language stands at a very good compromise between expressiveness and reversibility. The language is expressed through the λ-calculus, and abstraction and reduction are reversible,

so logical inference is sound, but the geometric algorithms associated to geometric predicates and function symbols do take into account the spatial interactions between diagrammatic objects, with the subsequent entropy reduction, and the system has high expressiveness, good computational properties, and is still information preserving. In the same way that memory and logical inference interact in the solution of complex problems, with the corresponding entropy reduction, as illustrated in Chapters 6 and 7, the present geometric language shows how logical and visual inference interact too, which the additional entropy reduction provided by the knowledge of the space which is embedded in perception.

There is also a further question of how the diagram is interpreted as a concrete or a generic description. The analysis of the phenomena related to the recognition and perceptual interpretation of the external representation is the subject of Chapter 10. This level of interpretation is referred to as *perceptual inference*. The discussion is approached according to the concept of image developed in Chapter 4, in which the image corresponds to an intensional description of the buffer's content, from a particular interpretation perspective. The input to perceptual inference consists of the primal sketch (Marr, 1982), where in imprints are already characterized in terms of the most salient features like corners, joints and edges, which is extensional information. However, such information can be organized from different ways or views, and each view corresponds to an extensional characterization of the buffer's content. The perceptual inference process involves the selection of the appropriate view and its promotion to the intensional level. Here again, there are different possible intensional views for the same extensional information, and each of these views corresponds to an image. The discussion shows that this is not a problem of images processing or low level recognition, as the input to perceptual inference is already the primal sketch, or an alternative representation at this level of processing. Rather, the discussion is centered on a number of aspects that have to be taken into account in the selection of the relevant extensional information and the construction of the relevant intensional view. In particular it is shown that an intensional description corresponding to the image can be concrete, but can be generic or abstract too, and that the decision of which view is produced does not depend on the external diagram, but on the perspective that is

taken by the interpreter the extensional versus intensional dimension. In this chapter is also discussed whether there is a role of the memory buffer in the production of descriptions, in particular of emergent objects, and whether the realization of the corresponding objects involves imagery operations. It is argued that this is indeed the case. The chapter is concluded with an analysis of dispute between the British Empiricist Locke and Berkley about whether an image can express a geometric abstraction, as held by Locke, or whether it needs to have a concrete interpretation (Tye, 1991). It is shown that according the present theory, Locke was right.

In Chapter 11 the diagrammatic generative machinery involved in the production of diagrammatic sequences is presented. Diagrammatic sequences can be modeled with generative systems of different sorts (e.g., Stinny, 1975, 2006) and a particular system developed in context of the *Graflog* (Pineda, 1993) and its extension and refining in the *Pitágoras* system (Pineda, 2007) is discussed. This system is also inspired by Piaget's notion of action schemes (Piaget, 1970), which are interiorized patterns of perceptual behavior that play an important role in playing and imagination, and the generative rules in the present theory are called *action schemes* too. Diagrammatic sequences are intensional, and the identity of the objects in the sequence is preserved by the application of the scheme. Action schemes can be global and modify a diagrammatic object as a whole, or structured, where an object is composed by a number of parts, and the scheme can manipulate such parts locally. Actions schemes are relative to a context in which these can be applied, and also to a focus object, which is a fixed reference for the change, and an *actee* which is the object that is modified by the scheme. The application of an action scheme also involves an inference to determine the focus and actee, which permits to modeled how the attention is shifted during the diagrammatic sequence and the problem solving process. Action schemes are generic and their definition is independent of the positions, size and orientations of the objects involved in the scheme. In addition, although the problem space may be quite large, the focus and actee selection process provides good heuristics to constraint the search. The machinery is illustrated with the diagrammatic sequence in Figure 1.1, which can be produced with three action schemes, in addition to a basic scheme that introduces the triangle seed, upon which the whole sequence is produced. However, the generative

machinery does not produce the emerging squares in the sequence, as these are the product of reinterpretations, and solution of the problem resides in a further problem space that interacts but is parallel to the space where the overt sequence is produced. This plane is also illustrated with the squares that emerge in the theorem of Pythagoras.

The generation of a diagrammatic sequence proceeds in tandem with the synthesis of a representation of the concept expressed by the diagram. If the diagram expresses a diagrammatic theorem or its proof, the concept of the theorem corresponds to the knowledge that one has when he or she known the theorem. In the present formulation, concepts are represented by Boolean functions, such that when applied to individual objects of the proper sort return the value of true or false depending on whether the argument belongs to the extension of the concept. There are also two semantic planes of expression: the first is purely spatial and geometric and functions in this plane represent the geometric theorems proper, and the second is a conceptual plane where the interpretation of the diagram is expressed. The former plane is represented with the geometric language, and the latter with a declarative language, which is also a functional language. The synthesis of these functions in both planes is referred to as *diagrammatic derivation* and the study of these derivations is the subject of Chapter 12. The capacity to establish judgments of equality presupposes that there is also a general concept of equality, form which particular equalities can be derived. The inputs to the two semantic planes in diagrammatic derivations are such generalized equality concepts, which in the present theory are referred to as *conservation principles*. These are also inspired in part in the corresponding notions developed by Piaget in his mental development theory, where a conservation principle is the knowledge required to assess whether a property of an abject is preserved in a process of change. Conservation principles are themselves functions, which are also expresses in the system's representational languages, and can be applied and evaluated by the corresponding interpreter program. Conservation principles can be global or structured, and in this latter case their application is relative to an argument that remains constant in the change process, which is referred to as *the focus*. As changes are produced by action schemes, conservation principles and action schemes are related through the focus, which is the same object. In a valid diagrammatic derivation the action scheme must preserve the

property associated to the corresponding conservation principle, and this restriction guarantees that the synthesized theorems and proofs are also valid. The process is also illustrated with the theorem of Pythagoras where the product in the first plane is a geometric Boolean function whose arguments are a right triangle and three squares which is true if these four objects stand in the Pythagorean relation and false otherwise, while the product in the second plane is a Boolean arithmetic function of three numerical arguments, which is true if the square of the first is equal to the sum of the squares of the last two. This latter condition is meet if the three squares numbers constitute a Pythagorean triplet, and this is the case if these numbers represent the three geometric squares that stand in the Pythagorean relation respectively.

In Chapter 13 the machinery of conservation principles and action schemes is extended to the inductive case, and the theorems and proofs in Figures 1.2, 1.3 and 1.4 are studied in detail. It is shown how the geometric relations are synthesized and represented in the geometric and arithmetic languages respectively. In this latter case, both geometric and their corresponding arithmetic theorems are represented through recursive functions, with proceed from the initial base case in the case of the theorem of gnomons, or converge to a limiting infinitesimal square in the latter two theorems. It is shown that the base case and the inductive step are produced by direct inference in the diagram, but these two basic functions need to be applied to the concept of induction, which is a pure abstraction that is independent of the space.

The overall discussion of the theory is presented in Chapter 14. First it is briefly discussed how the machinery can be applied to logical diagrams, and also to the visual analogies of Raven's test. The view of images as intensional objects with a dual aspect as descriptions and as the content of a memory buffer and imagery is reviewed on the light of the overall theory and the main case studies. This notion is the basis for the taxonomy of diagrammatic system in AI, and systems Type-1, Type-2 and Type-3 are reviewed in the overall perspective, and the way in which direct inference in the three settings is also discussed. In particular, is discussed the requirements for the full automation of systems of Type-3.

The discussion is the turned to the more general questions about the expressive power of diagrams and graphical representation more generally, and why diagrams are so ubiquitous and effective in reasoning and problem solving, and also in learning, design and creativity. This is related to the trade-off between expressivity and reversibility of abstraction and reduction. Diagrams can express incomplete information and constraints very effectively, and also the interactions between the objects within abstractions. This is reflected in both memory inference and perceptual inference, which are involved in diagrammatic reasoning in addition to logical inference, which has a deductive character. Memory inference is illustrated with the system *F* where abstractions capture the interactions between the functions that share the same values for subset of their arguments, with the subsequent entropy reduction, as illustrated in Chapters 6 and 7. On the other hand, concepts and geometric descriptions involved in spatial inference have an associated geometry associated to them, and geometric computations do take into account the interactions between the objects in the space, reducing the entropy of the representation, as illustrated by the *Pitágoras* systems. In both memory and perception, the interactions are considered, the entropy is reduced and inference, and the inferential cost is reduced accordingly. Finally, logical, perceptual and memory inference interact systematically in reasoning and problem solving.