Combinatorial Algebraic Topology for Diagrammatic Reasoning

Jean-Louis Giavitto, Erika Valencia
LRI, Université Paris-sud, Orsay*
{giavitto, erika}@lri.fr

6th May 1998

Key Words: Formal Method, Diagrammatic Knowledge Representation and Inference, Mathematics of Diagrams, Diagrammatic Reasoning in AI Applications, Analogy Solving, Classification, Ontology, Hierarchy, Algebraic Topology, Simplicial complexes.

1 Introduction

The idea developed here is that *combinatorial algebraic topology* (**CAT** in short) is an adequate and unifying framework to specify and analyze diagrammatic representations and diagrammatic reasoning.

Our arguments can be sketched as follow:

1. Diagrammatic reasoning is the use of spatial relationships (neighborhood, border, dimension, path, hole, ...) to represent and structure knowledge. It cannot be restricted to visual (i.e. low-dimensional) representation of knowledge, nor to the development of a logical account of spatial relations.

2. Objects and relationships implied by diagrammatic representation have a pre-metric, discrete and finite nature even if some *continua* are involved.

3. Combinatorial algebraic topology is a theoretical framework which precisely formalize finite discrete spatial relationships. This well established mathematical theory provides a sound basis to specify diagrammatic representations and diagrammatic operations. Furthermore, the algebraic approach of the theory enables a *constructive* description (that is: an algorithm can be derived and the diagrammatic reasoning can be automated).

To support these affirmations, we discuss them briefly. We present a possible application of some elementary CAT concepts to diagrammatic knowledge representation, originated in the works of Atkin on the *Q-analysis* [Atk74].

Then, instead of rephrasing well-known diagrammatic applications in the CAT framework, we have found more illuminating to present a topological formalization of applications that are obviously diagrammatic (they involves lattice graph and geometric configuration) but that have not received until now a specific diagrammatic treatment.

The first application is a categorization problem. The second one concerns the taxonomic reasoning and the problem of restructuring ontologies (the presentation remains at a general level). The third application, more widely presented, has raised the development of the ESQIMO system [VGS98, VG98] for solving analogies in unsupervised IQ-tests.

*LRI, URA 410 CNRS; Bat 490 U-Paris-Sud; 91405 Orsay Cedex; France

2 Diagrammatic Reasoning, Spatial Relations and Algebraic Topology

Diagrammatic reasoning is a field of research investigating the use of spatial relations for knowledge processing. This include knowledge representations, retrieval processing, inference making, etc. Several issues are addressed in these active fields (see [GNC95] for an excellent introduction), e.g.: visual formalism [Har95], diagrammatic inference [GPP95, Lin95], diagrammatic approach of logic [Shi91, BE95], qualitative physics [For95], cognitive issues [Gar83, Arn69, GNC95, chap. III], logical formalization of spatial relationships [Got94, LP97, Ham97, Ben94]. The last example account for the search of a formal theory of diagrammatic representations. A unique conceptual framework cannot encompass simultaneously all the issues investigated in the field of diagrammatic reasoning. However, it is possible to develop a formal framework to describe the basic objects and processes that are specific to it.

Current logical approaches to diagrams suffer from a “loss of graphicality” [LP97]. The intuitive representation and the direct expressiveness that makes diagrams so appealing for knowledge representation and processing must be preserved. So, we advocate the use of combinatorial algebraic topology for such a linguistic support.

Diagrammatic representation exploit spatial relations like: neighborhood, point of contact, intersection, containment, object configuration, paths... These relations are also basic notions studied in *geometry*.

Although geometry studies these relations, we are not interested in the continuous and metric structure of geometrical objects. The primitives objects and relations involved in diagrams have a finite and discrete nature. For instance, a graph pictured in the plane involves edges represented as line segments. A line segment has a continuous nature but this is irrelevant for the graph structure: the precise shape of the edges does not matter, only the connection implied between two nodes. The same remark can be made about *Venn diagram*, *state-charts*, symbolic map, and many other kind of diagrams: it is the configuration of finite sets of objects that is relevant.

When metric aspect turn to be important, they are generally *relative*: A is bigger that B, C is closer from D than E, path F is shorter that path G, etc. Therefor, only an order structure is necessary, not a metric structure.

Moreover, we cannot restrict diagrams to plane geometry. For example, the realizability of a Venn diagram representing an arbitrary predicate requires working in a 3 dimensional space [LP97]. Path equivalence depends of the underlying space structure (e.g. all closed paths are equivalent in the plane, and this is not true on a torus). So we have to consider general spatial structures in any dimension.

Hence, if we neglect quantitative diagrammatic representation (like bar-chart, geological survey map, etc.) we can focus on *n-dimensional combinatorial algebraic topology*. Topology studies object and properties that are invariant under continuous deformation, thus it does not account for quantitative spatial information (metric properties). Combinatorial topology focus on the study of a finite set of objects satisfying some spatial relation. And algebraic topology develops the application of algebraic tools to topological problems. Such an approach is very attractive because we are particularly interested in the development of “constructive” object, i.e. object that can be managed by a program.

In the rest of this paper, we present the use of some basic CAT notions to knowledge representation and diagrammatic reasoning.

3 Algebraic Topology and Diagrammatic Representation

3.1 Simplicial Complexes

The following definitions are standard in topology. A k-cell is an abstract space of dimension k with no hole. These k-cells are elementary bricks to build a cellular complex K under the two conditions: the edge of any k-cell of K is the union of k-cells of K and the intersection of k-cells of K is another cell of K or is empty.

A cellular complex is the description of a space build piecewise. A possible and rather abstract algebraic instantiation of this concept is the one of *abstract simplicial complex* defined below.

Definition 1 *An* abstract simplicial complex *[Hen94, HY88] is a couple (V, K) where V is a set of elements called vertices of the complex and K is a set of finite parts of V such that if $s \in K$, then all the parts $s' \subseteq s$ belong also to K. The elements of K are called abstract simplexes. The dimension of a simplex s is equal to $Card(s) - 1$. The dimension of the complex is the dimension of its biggest simplex.*

A p-simplex s is noted: $s = \{v_0, v_1, ..., v_p\}$, where $v_i \in V$ are the vertices of s. A complex is a set of sets closed for the inclusion and the intersection. All p-complexes with $p < 2$ are graphs. Thus, the simplicial complexes are particularly attractive to generalize semantic networks by keeping the possibility to express hierarchies like in a relational graph.

We need also a formalization of the notion of discrete path: the *polygonal chain*.

Definition 2 *Let $\alpha = (\sigma_0, \sigma_1, ..., \sigma_n)$ be a sequence of simplexes belonging to a complex K. The sequence α is called a* polygonal n-chain *of origin σ_0 and end σ_n if for all couple (σ_i, σ_{i+1}), $\sigma_i \cap \sigma_{i+1} \neq \emptyset$. The dimension of α is the smallest dimension of $\sigma_i \cap \sigma_{i+1}$.*

We say that two simplexes σ_1 and σ_2 are q-connected if there is a polygonal chain of dimension q that connects σ_1 with σ_2. Any p-simplex is p-connected to himself with a 0-chain.

On top of these elementary entities, CAT develops among others the notions of border, study the equivalence of paths (homotopy) and the classification of simplicial complex through their closed path (homology). However, the elementary entities we have introduced are already sufficient to support interesting applications.

3.2 Binary Relations Representation: the Q-analysis

Atkin proposed to represent a binary relation between two sets with a simplicial complex: it is the **Q-Analysis** [Atk77, Atk81, Joh91a]. Q-Analysis has already been used to model traffics [Joh91b], interactions between agents [Dor86, PLC91, Cas94, chap. 8], position analysis at chess [Atk76] and social relations [Atk77, Gou80, Cas94].

Let Λ be the incidence matrix of a binary relation $\lambda \subset A \times B$. Let $a \in A$, the set S_A of b_i such that $(a, b_i) \in \lambda$, can be directly read from Λ, as the a-column (see figure 1).

All the elements b_i can be taken as vertices to represent the element a as a simplex. The whole matrix Λ can then be represented as a simplicial complex containing all the simplexes representing each element $a_i \in A$, we note it $K_A(B, \lambda)$ (see figure 1). Likewise, we can represent Λ^{-1} with the dual simplicial complex $K_B(A, \lambda^{-1})$. We say that $K_A(B, \lambda)$ and $K_B(A, \lambda^{-1})$ are conjugates.

3.3 Representing Sets of Predicates

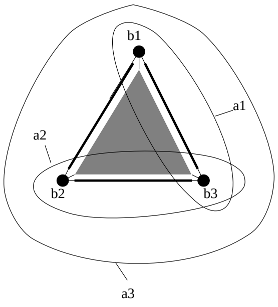
We extend the Q-Analysis to allow the representation of sets of predicates as a simplicial complex too [Val97]. We take a set of predicates $P = \{p_1, p_2, ..., p_n\}$ and represent the binary relation $\lambda \subset A \times P$ such that $(a_i, p_j) \in \lambda$ if $p_j(a_i)$ holds.

Take for example the set of integers $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and the set of predicates $P = \{\text{parity}, \text{oddity}, \text{primality}, \text{multiple of 3}\}$. The representation of this relation in terms of incidence matrix is given by table 2(a) and in terms of simplicial complex by figure 2(b).

Notice that a representation based upon simplicial complexes associates the same simplex to elements of A that cannot be distinguished. In other words, two elements will be separated only if there is at least one predicate that allows the differentiation. The same situation occurs with the dual complex:two predicates are distinguished if there exists a state of the world that differentiate them. Two simplexes that have a smaller k-simplex in common are said to share a k-face. In

λ	a_1	a_2	a_3
b_1	1	0	1
b_2	0	1	1
b_3	1	1	1

(a) Incidence matrix



(b) Simplicial representation

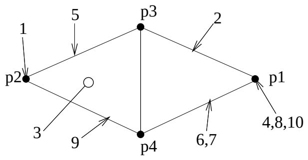
$$\{ \\ \{b_1\},\ \{b_2\},\ \{b_3\}, \\ \{b_1,\ b_2\},\ \{b_2,\ b_3\},\ \{b_1,\ b_3\}, \\ \{b_1,\ b_2,\ b_3\} \\ \}$$

(c) Set Representation of a_3 viewed as a *complex*

Figure 1: Simplicial representation of a binary relation λ. We have $\lambda(a_1) = \{b_1, b_2\}$. So we represent a_1 as a 1-simplex, b_1 and b_2 being its two vertices. A simplex can also be viewed as a complex: it then contains all its parts.

	p_1	p_2	p_3	p_4
1	0	1	0	0
2	1	0	1	0
3	0	1	1	1
4	1	0	0	0
5	0	1	1	0
6	1	0	0	1
7	0	1	1	0
8	1	0	0	0
9	0	1	0	1
10	1	0	0	0

(a) Incidence matrix associated with λ



(b) Simplicial representation of λ

Figure 2: Dual complex associated with $\lambda \subset \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \times \{p_1, p_2, p_3, p_4\}$ where we can see that the integers $4, 8$ and 10 are identical with respect to the p_i predicates.

terms of representation, it means that *they have k features in common.* As Freska emphasized it, we call here for the use of discriminating features rather than for precise characterization in terms of universally applicable reference system [Fre97].

4 Categorization in the CAT framework

4.1 Rephrasing Categorization in Topology

Categorization is the process that extract concepts from percepts. It can also be described as the construction of (mental) representations from the (physical) world.

The Holland model. Holland gives a simple model of the process of categorization for the construction of a *homomorphic* representation that maps many elements of the world to one element of the representation. We need a categorization function, say C, that maps the states of the world onto a smaller number of states. This function is defined in terms of detected properties [HHNT86]. The properties of the states of the world are encoded by detectors, we will model them as binary detectors, say d_1 to d_n. We can imagine those detectors as a retina, like in the

WHISPER system [Fun95]. The detectors divide the world up into equivalence classes: objects that are undistinguishables by the detectors are treated as equivalent.

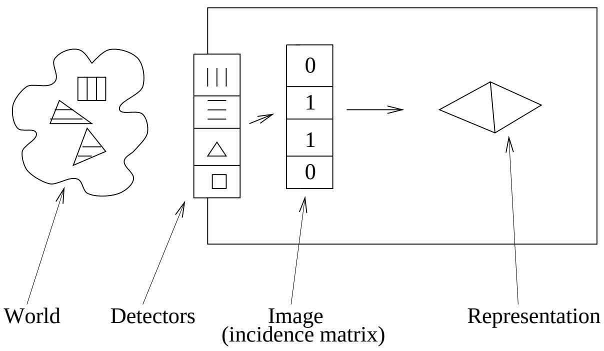


Figure 3: The detectors are lighted *on* whenever the property they encode is present in a state of the world. The two triangles are here treated as equivalent since no detector allows their distinction. The state of the detectors (*on/off*) is then used to construct a representation.

Consider any transition T of the world, such that the state of the world at the instant, say $(t+1)$ is given by $W(t+1) = T(W(t))$, where W(t) is the state of the world at the instant t. According to Holland, a homomorphic model implies the commutativity of the operations of transition T and of categorization C. Thus we should have $C(W(t+1)) = C(T(W(t))) = T(C(W(t)))$. But in practice, a model will not be completely homomorphic because there can appear new objects that were not taken into account with the first categorization function. An additional detector will then be necessarily, and the model will be quasi-homomorphic. The process of model construction can then be viewed as the progressive refinement of quasi-homomorphisms.

The CAT formulation. Starting from the preceding states of the detectors, we can build a representation of the state of the world that is being encoded. Indeed, the detectors having two possible states *on/off* (or $0/1$, see figure 3), there is a binary relation between the set of detectors and the set of states of the world that we can represent with an incidence matrix and thus with simplicial complexes.

For example, on the numbers exercise given above, the detectors would be the properties *even, odd, prime, multiple-of-3*, and the states of the world analyzed would be the numbers from 1 to 10. Then, the topological representation constructed is the one given on figure 2(b) since it represents the incidence matrix between the states of the detectors and the numbers.

A category is a simplex in the previous complex. For instance, the set of numbers satisfying $p_2 \wedge p_3$ is the category of "odd and prime numbers". The *extent* of this concept is the set $\{3, 5, 7\}$. This is reminiscent of the *formal concept analysis* [Wil97, Gan95]. This theory is a data analysis method, founded on lattice theory, that considers as categories only the simplexes s that are defined uniquely by their extent (that is, $s = \lambda^{-1}(\lambda(s))$). Although developed on top of lattice theory notions, formal concept analysis can be rephrased without problem in the CAT framework (Cf. discussion in section 7).

4.2 The (Toy) Example of the *Little Red Riding Hood*

We have sketched above how a diagrammatic representation can be obtained from a categorization process. We apply this technics to a toy example of automatic story analysis and show how a diagrammatic "formal concept analysis" can be conducted on the story of the Little Red Riding Hood. The resulting ontology will be represented as a diagram and expressed with our topological framework.

Characters, Objects and Actions	Encoding
Little Red	alive, good, small
all others human	alive, good
Wolf	alive, animal, small
Trees	alive, place, several, bad
House	place
Basket	small
Give	motor, exterior
Sleep	motor, good
Eat	motor,interior, good
Walk	motor
Talk	motor, exterior

Table 1: Detectors encoding the objects of the world of the Little Red Riding Hood.

	Detectors View	Scene of the story
1.	Red, Mother, talk, house	The mother tells Red her mission
2.	Red, Mother, give, basket, house	The mother gives the basket to Red
3.	Red, walk, tress, basket	Red walks into the woods
4.	Red, Wolf, talk, trees, basket	Red meets the wolf
5.	Red, walk, trees, basket	Red continues her walk
6.	Wolf, walk, trees	The wolf also goes to Grandma's
7.	Grandma, sleep, house, bed	Grandma is in her bed
8.	Grandma, Wolf, talk, house, bed	The wolf meets Grandma
9.	Wolf, eat, house, bed	The wolf eats Grandma
10.	Red, Wolf, talk, house, basket, bed	Red talks with the wolf (disguised as Grandma)
11.	Wolf, eat, house, bed, basket	The wolf eats Red too

Table 2: The story of the Little Red Riding Hood viewed through the detectors.

Formalizing the Story. We take a very simple set of detectors : *alive, animal, good, bad, place, small, several, motor, exterior, interior.* With those detectors, we can encode the principal characters, objects and concepts of the story according to table 1. These encodings are of course arbitrary, but the important thing here is that we have a finite number of detectors that can encode the states of the world and that allow the distinction between different objects of the world. The story can be told into say 11 successive images, or states of the world, according to the presence of the primitive objects or concepts (see 2).

Several strategies are possible to extract the simplexes considered as categories. Formal concept analysis use maximal simplexes. We have investigated two others strategies: the first one extracts concepts incrementally from the first image to the last one, and the other one builds directly and immediately a taxonomy from the 11 images as a whole. The two extractors (incremental and instantaneous) have been prototyped in the `Mathematica` programming language.

The two ontologies are pictured as *Hasse diagrams* in figure 4. In this representation, each point is a simplex and the link represents inclusion. In the Hasse diagram given on figure 4, the concepts represented at level n, are ontologically precedent to the ones represented on the level $n+1$. For instance, the category $\{$*place, motor, alive*$\}$ requires the $\{$*alive*$\}$ concept.

In the second extraction, most of the concepts extracted correspond to the reaction of one detector alone. This is due to the fact that there are enough objects in the images given as states of the world to the system so that these detectors can be distinguished. Indeed, two objects are represented as different if and only if there exist a detector that allow their distinction. But we can also say that two detectors are distinguishable only if there is a perception in the world that allows their distinction.

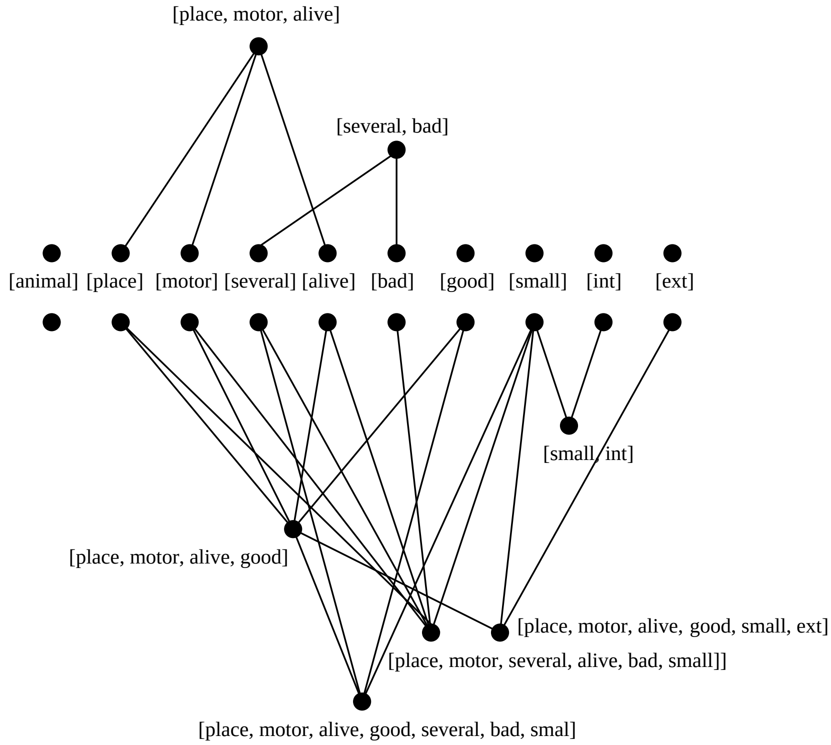


Figure 4: Two ontologies extracted from the Little Red Riding Hood story. The ontology extracted instantaneously is represented top-down on the higher part of the figure, the one extracted incrementally is represented bottom-up on the lower part of the figure.

Discussing the results. The concepts extracted as simplexes from the instantaneous or incremental perception of the images do not necessarily have a name in our usual vocabulary. Indeed, in our way of telling the story in English, we use concepts that have been learned, or extracted from other perceptions of the world. However, in the small world of the story, there are some "concepts" that are never separated, they are useful for the description of each scene.

Among these groups of "detections", we can study the particular case of the concept $\{$*motor, alive, place*$\}$. These detectors are always lighted *on* because these three objects are present in each scene. In fact, they are necessary for the description of any event in a story, they are used in the description of any place, any action, made by anyone! This is not surprising since when we tell a story we only tell events happening in a place to someone to add new information to the story, especially when we simplify the text to the clue-scenes.

With this representation, we can analyze a tale and extract some patterns that are used for its description. We could for example, compare the analysis of this tale to the analysis of another tale and see if the patterns are always the same. We could also use the incremental analysis to compare the structure of different tales through time. Another point of investigation would be to analyze the characters hierarchy, a task well known in the field of narratology.

5 Inheritance Restructuring in Object Oriented Programs

As a further example of the use of a topological framework to formalize diagrammatic reasoning for ontologies, we present now an algorithm for *inheritance hierarchy restructuring* proposed by [moo] in the field of object oriented programming. The aim of this algorithm is to infer or restructure the inheritance hierarchy of classes to achieve smaller, consistent data structure and better code re-use. We chose this example because it is simple to explain and well formalized. The CAT framework provides a concise and clear language to specify this algorithm and exhibit its diagrammatic nature.

We will call *features* any property, behavior, instance variable or method that can be used for the description of objects. These features are grouped into classes. Thus, a class corresponds to the description of a type of objects sharing a set of features. Using inheritance to specify classes, express explicitly the hierarchy relationships between the class definitions. It makes initial definition and evolution of the system easier.

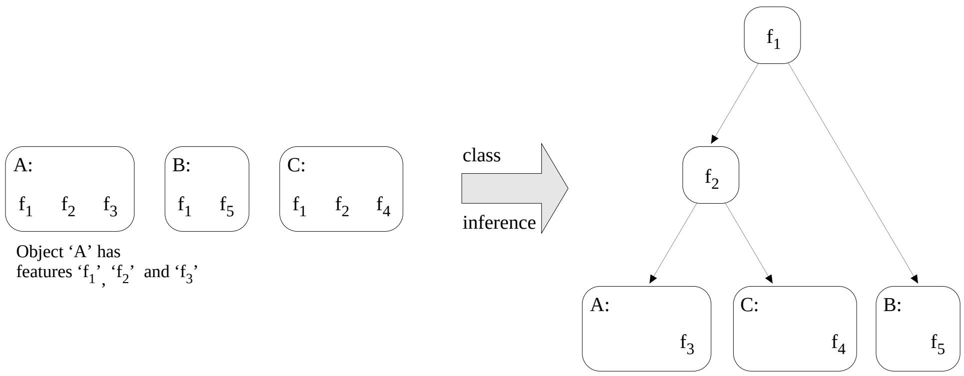


Figure 5: From a collection of concrete objects to an inheritance hierarchy. A set of concrete objects is given on the left. A possible hierarchy that accepts this set of objects is given on the right. The features linked to a class are the set of features defined for this class merged with the features inherited (recursively) from the parent classes. A class without name is called an *abstract class* in the object oriented programming terminology and corresponds to an internal node of the inheritance graph (hierarchies with multiple inheritance will be graphs rather than trees).

Moore [moo] proposes an algorithm, called IHI, to infer automatically the inheritance hierarchy from the flat description of the objects by their features. In the computed hierarchy, there must

Object Oriented Concepts	CAT Translation
Feature	Vertex
Class	Distinguished simplex the simplex is made by the set of features linked to a class (but note that every simplex is not a class
Class inheritance direct and indirect	Simplex inclusion
Inheritance graph of a set of classes	Complex made by these classes

Table 3: Correspondence between the OOL terminology and the CAT framework

.

be a class corresponding to each concrete object (see. fig 5). The inheritance structure can take various form (e.g.: a unique disconnected class for each object). Further criteria must then be specified to constrain the possible hierarchies. The following five criteria are natural in the field of programming languages:

1. Every feature should appear in only one class (this correspond to the maximal sharing of features between classes).
2. There is a minimal number of class.
3. All inheritance links that are consistent with the object structure must be present (that is: if the set of objects which inherits from a class `C` also inherits from a class `D`, then `C` should inherit, directly or indirectly, from `D`).
4. The number of explicit inheritance links must be minimized (the links which are implied by transitivity of the inheritance relationships must not be made explicit).
5. The concrete objects should correspond to leaves of the inheritance hierarchy.

These criteria together are sufficient to specify a unique solution.

The problem of inferring a hierarchy from a set of concrete objects can now be rephrased into the CAT framework with the "dictionary" given on table 3.

Note that only *some* simplexes are classes because a simplex corresponds to a *subtype* of the types implied by the classes (for the notion of sub-type Cf. **??**). For example, the simplex $\{f_2, f_3\}$ is a subtype of class `A` but does not appear in the classes of the hierarchy pictured in fig. 5.

This translation of a class hierarchy into a simplicial complex does not rely on a technical trick. It depends on a deep correspondence between the lattice structure and the simplicial complex structure (Cf. discussion in section 7).

The criteria used to constrain the class inference are then additional *topological constraints that have a simple intuitive diagrammatic meaning*:

1. Every feature appears in a distinguished simplex[1].
2. A minimal number of simplexes are distinguished.
3. The third property is automatically achieved within our translation.
4. A class inherits from the maximal classes it contains[2].
5. Concrete objects are simplex of maximal dimension.

[1]This condition simply states that the hierarchy preserves the features of objects. The maximal sharing of features between classes is automatically achieved by the nature of our translation.

[2]Here, *maximality* of a simplex s in a set S of simplexes simply means that there is no $s' \in S$ such that $s \subset s'$ and $s \neq s'$.

This conditions are simply expressed (the formal expression of the initial criteria take about two pages in [moo]). Moreover, the problem of inferring an inheritance hierarchy is then translated in the problem of distinguishing some simplexes satisfying the previous properties in the complex made by the concrete objects.

6 Simple Diagrammatic Analogy Solving : the ESQIMO System

As a third application of a topological framework for diagrammatic representation and reasoning, we explored the possibility of a topological representation to support analogy [VGS98, VG98]. The analogy solving between a source and a target domain is modeled as a topological transformation of the representation of the source into the representation of the target in some underlying abstract space of knowledge representation.

The task is to answer a typical IQ-test by giving an element called D such that it completes a four-term analogy with three other given elements A, B and C: "find D such that it is to C what B is to A". This kind of analogy solving has already been studied by Evans [Eva68], but in our work the solution has to be build from scratch since no set of possible solutions is given to choice. We call this kind of IQ-test-like problems, *non supervised*. This four-term analogy solving is usually decomposed into four steps [Eva68].

- Find the possible relations R_{AB} between A and B.
- Find the possible relations R_{AC} between A and C.
- Apply R_{AB} to C only on a domain determined with R_{AC}.
- Verify the symmetry by applying R_{AC} to B.

6.1 Diagrammatic Representation of the problem

Usually, IQ-tests are given in terms of geometrical elements so that they can immediately express different properties and still be simples. We will also chose a geometrical and simple universe Ω similar to the one investigated in [WS90] including basic shape, form and color properties:

- *shape*: round, square, triangular;
- *color*: white, black;
- *size*: big, small.

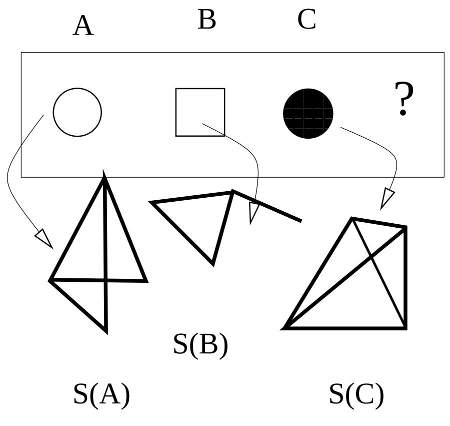
Each figure is composed of elements of Ω. In the case of one-element figures, we represent them as simplexes S_A, S_B and S_C, while in the case of composed figures (more than one element each) we represent them as sets of simplexes $\{S_A\}$, $\{S_B\}$ and $\{S_C\}$.

6.2 Algorithm

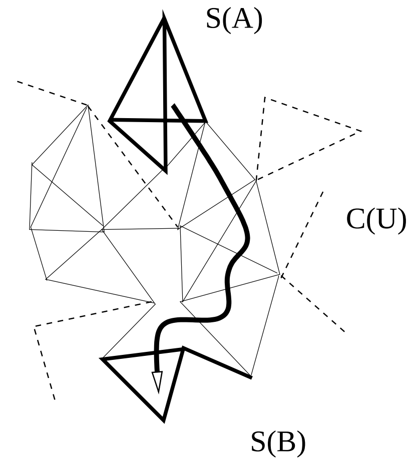
To solve the four-term analogy, we first look for a relation R_{AB} between A and B, that is we look for a topological transformation T_{AB} between S_A and S_B (see figure 6) .

6.2.1 Relation between A and B

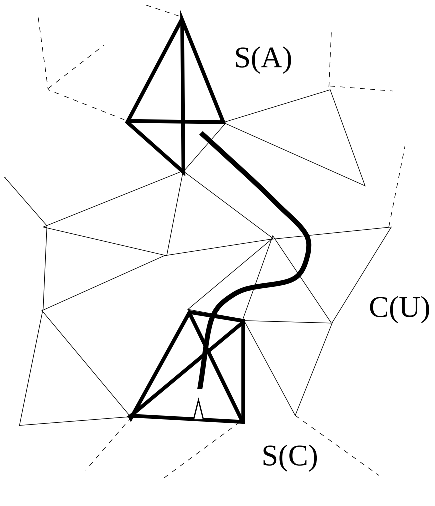
In the case of simple figures, the transformation T_{AB} is seen as a polygonal chain from S_A to S_B in $\mathcal{C}(\Omega)$. An elementary step linking S_i to S_{i+1} in a chain is then viewed as an elementary transformation $T_{S_i,S_{i+1}}$. A polygonal chain from S_A to S_B is then a transformation of A into B given by: $T_{S_l,S_B} \circ ... \circ T_{S_A,S_1}$.



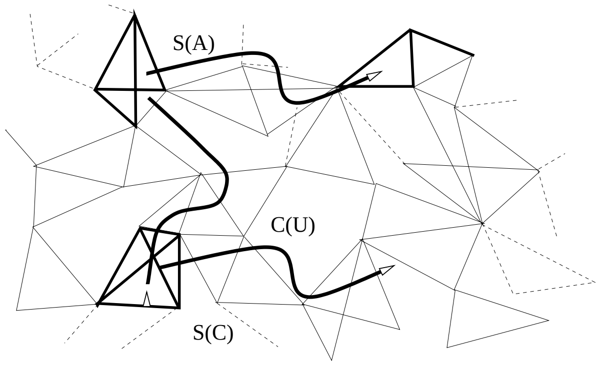
(a) Figures A, B and C are represented as simplexes



(b) Transformation T_{AB} is a polygonal chain from S_A to S_B into $\mathcal{C}(\Omega)$



(c) Domain of S_C to which we apply T_{AB}



(d) Construction of S_D

Figure 6: Four steps of ESQIMO's algorithm to solve IQ tests

If there are several chains, then we say that there are several possible relations between A and B. We can choose to minimize the number of possible solutions, by giving a higher priority to polygonal chains that are short and of higher dimension, that corresponds to choose a transformation that requires less steps and that preserves more properties. This is comparable with selecting a *best* solution according to some measure of satisfaction like in [Eva68].

For composed figures, the transformations can be of several types: destruction, creation, metamorphosis, division, junction (like in the changes introduced by Hornsby [HE97]). We first pair the simplexes of $\{S_A\}$ with those of $\{S_B\}$ and look for transformations between the simplexes of each pair. The transformation T_{AB} is then the parallel application of the transformation found for each pair.

There are many possible pairings leading to different or to the same solutions. The only constraint we need here is that all the vertices and faces of $S(B)$ are paired with vertices from $S(A)$. A pairing of a vertex of $S(A)$ with $\emptyset$ means its destruction, the pairing of a vertex of $S(A)$ with one vertex of $S(B)$ means its transformation and the pairing of a vertex from $S(A)$ with several vertices of $S(B)$ means its duplication and transformation.

6.2.2 Construction of D

To apply T_{AB} to S_C we have to extend the domain of T_{AB}, and so extend T_{AB} to T'_{AB} such that $T'_{AB}(S_C) = S_D$ and $T'_{AB}(S_A) = S_B$. There are different possible strategies to determine the domain of $S(C)$ on which we can apply T_{AB}. Several strategies have been implemented considering only the things that changed between $S(A)$ and $S(C)$, or considering only the invariants between them, or some other hybrid methods.

6.3 Examples of Analogy Solving with ESQIMO

We give examples of IQ-test solving with ESQIMO, where the geometrical elements have names going from `e1` to `e12` (see figure 7). The corresponding geometrical representation of composed example is given on figures 8.

ESQIMO has been prototyped in the `Mathematica` programming language [Wol88] and implemented in `ML` [Ler93]. To ask ESQIMO to solve the IQ-test, we call the function (for more details see [Val98]):

```
A1={e1,e2};
B1={e7,e5};
C1={e3,e1};
Resolve[A1,B1,C1,App2,AppApp2]
```

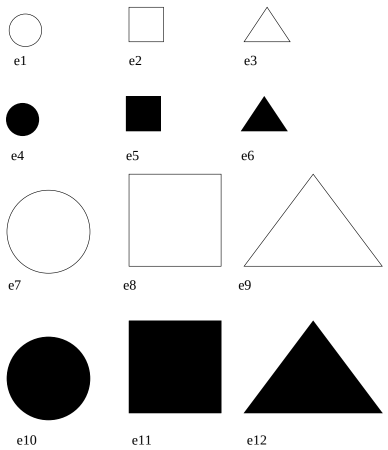
With the solution, the system gives output about: the pairs chosen in terms of association sets called `AssocSet`; the elementary transformations `D-elem` applied simultaneously (`Par`) or sequentially (`Seq`) ; the set `Choose` of solutions. Here we only have one best solution composed of the elements `e9` and `e4` being a big white triangle and a small black circle (see figure 8).

```
Choose[AssocSet[FromTo[1,{1}],FromTo[2,{2}]],
       AssocSet[FromTo[1,{1}],FromTo[2,{2}]]]}

Par[Domain[1,Seq["D-elem"[SmallQ->0,BigQ->1]],{e3}],
    Domain[2,Seq["D-elem"[WhiteQ->0,BlackQ->1]],
          {e1}]]}

Choose[{e9,e4}]
```

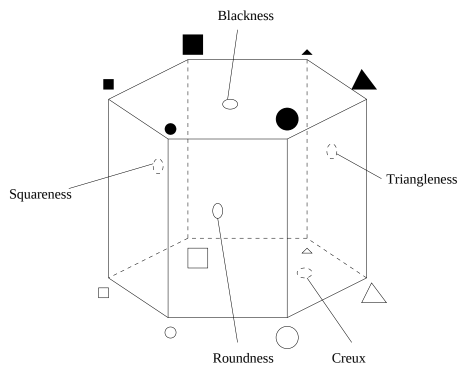
6.4 Discussion on the ESQIMO's algorithm

Many choices made in ESQIMO's algorithm can be discussed, or can be seen as other additional strategies parameterizing the ESQIMO kernel:

- The description of the properties of each figure in terms of predicates can be a problem for properties such as position. In that case, we could give each possible position a predicate that could be true or false, or we could only take relative positions into account.
- The way we associate a transformation to a given polygonal chain is not unique. In particular, our transformations could be called $0-$degree since they preserve the minimum of topological properties along a chain. The next step consists in pairing complexes for composed figures.



(a) Elements of the universe Ω of ESQIMO



(b) Representation as a Complex$\mathcal{C}(\Omega)$.$\mathcal{C}(\Omega)$ has dimension 6: this representation is only a 2-D view of one possible 3-D representation.

Figure 7: Elements manipulated by ESQIMO and their representation as a simplicial complex.

(a) (b) (c) (d)

Figure 8: The first element becomes bigger and the second becomes black.

(a) (b) (c) (d)

Figure 9: The first element becomes black and the second becomes white, is duplicated and one of the duplicates is bigger.

(a) (b) (c) (d)

Figure 10: The first element is duplicated and one duplicate is squared. When squared, the property of triangleness is not taken off, this creates then an unstable solution, called a monster.

- The way we determine the domain of S_C on which to apply T_{AB} can also lead to different strategies depending on whether we consider only the intersection between S_A and S_C or the whole S_C.
- The measure of satisfaction to select a *best* solution is here to take the shorter and wider polygonal chain between the two complexes. This does not necessarily correspond to the more natural transformation between the two complexes, other measures of satisfaction can be tested.

Note that this model does not depend on the geometrical nature of the figures. Indeed, we could, for example try ESQIMO on verbal IQ-tests more like in the COPYCAT system [Hof84].

Different computational models have been developed to model analogy solving. Among them, the ANALOGY system proposed by Evans [Eva68, Ha91], the SME system proposed by Falkenhainer to illustrate Gentner's theory for analogy [FFG89, Gen83], the ARCS system developed by Thagard and Holyoak [THNG90] to simultaneously satisfy the structural, semantic and pragmatic constraints. We can hardly compare these systems to ESQIMO in terms of performances since we only studied intra-domain analogies with the only structural constraint in this first work. Our contribution lies principally in the search for a new representational structure to model analogy in terms of diagrams. The CAT structure of representation can be seen as a hybrid structure between a purely symbolic and a purely analogical approach, like neural networks in ARCS [THNG90].

ESQIMO has been prototyped in the `Mathematica` programming language and implemented in `ML` to include many of the heuristics discussed before. Even if this model can be considered as very simple, we are convinced that a topological representational structure is well-adapted for analogy modeling. We find the results presented here already surprisingly satisfying with respect to the simplicity of the underlying machinery and this motivates further investigation.

7 Conclusions

In this paper we have advocated for the use of combinatorial algebraic topology to describe and formalize various diagrammatic tasks. Our approach has been illustrated through three examples: automatic ontology extraction (categorization), ontology restructuring (inheritance hierarchy synthesis) and analogy solving, that cover a broad range of applications.

This preliminary work shows some evidence that the CAT framework is not tailored to a specific application domain and that it is relevant for a variety of diagrammatic processes modeling.

An important point to note is that we have only used *elementary* CAT notions: simplicial complexes generalize the concept graph and polygonal chains extent the concept of path. These two notions have an immediate and intuitive meaning, even in higher dimensions and are obviously diagrammatic.

Note also that there is a strong link between the concept of simplicial complex, *open* and *closed* sets of a topology and lattice theory. The simplex of a complex are the closed set of a natural discrete topology of the complex and they are also the element of the inclusion lattice of the complex. This link between the simplicial complex structure and the lattice structure explain why it is so easy to translate taxonomic reasoning problem (which involves lattices) into a topological problem.

Future work must include the use of further CAT constructions (like simplicial applications, homotopy group, homology classes, etc.) to handle more sophisticated diagrammatic situations. We will also investigate the use of the topological structure of open and closed set associated canonically to a complex as a "logic of observations" as suggested in [Vic88].

References

[Arn69] Rudolh Arnheim. *Visual Thinking.* University of California Press, 1969. trad. française : "La pensée visuelle", Champs-Flammarion, 1976.

[Atk74] R. H. Atkin. *Mathematical Structure in Human Affairs.* Heinemann, 1974.

[Atk76] Atkin & al. Fred CHAMP positional chess analyst. *International Journal of Man-Machine Studies*, 8:517–529, 1976.

[Atk77] R. H. Atkin. *Combinatorial Connectivities in Social Systems.* Verlag, 1977.

[Atk81] R. H. Atkin. *Multidimensional Man.* Penguin, 1981.

[BE95] Jon Barwise and John Etchemendy. Heterogenous logic. In *Diagrammatic Reasoning*, chapter 7, pages 211–234. AAAI Press, 1995.

[Ben94] Brandon Bennet. Spatial reasoning with propositional logics. In P. Torasso J. Doyle, E. Sandewall, editor, *Proc. of th fouth Int. Conf. on Principles of knowledge representation and reasoning (KR94)*, San Francisco, 1994. Morgan Kaufmann Publishers.

[Cas94] J. L. Casti. *Complexification.* Harper Perennial, 1 edition, 1994.

[Dor86] P. Doreian. Analysing overlaps in food webs. *J. Soc. & Biol. Structures*, 9:115–139, 1986.

[Eva68] Thomas G. Evans. A program for the solution of a class of geometric analogy intelligence-test questions. In *Semantic Information Processing*, chapter 5, pages 271–353. The MIT Press, 1968.

[FFG89] B. Falkenhainer, K. D. Forbus, and D. Gentner. Structure mapping engine. *Artificial Intelligence*, 41(1):1–63, November 1989.

[For95] Kenneth D. Forbus. Qualitative spatial reasoning framework and frontiers. In Janice Glasgow, editor, *Diagrammatic Reasoning.* AAAI Press/MIT Press, 1995.

[Fre97] Christian Freska. Spatial and temporal structures in cognitive processes. In C. Freska, M. Jantzen, and R. Valk, editors, *Foundations of Computer Science*, volume 1337 of *LNCS.* Springer-Verlag, 1997.

[Fun95] Brian V. Funt. Problem-solving with diagrammatic representations. In Janice Glasgow, editor, *Diagrammatic Reasoning.* AAAI Press/MIT Press, 1995.

[Gan95] Bernhard Ganter. Lattice theory and formal concept analysis: a subjective introduction. In Kirby A. Baker and Rudolf Wille, editors, *Lattice theory and its applications*, pages 79–90, Berlin, 1995. Heldermann-Verlag.

[Gar83] Howard Gardner. *Frames of mind.* Basic Books, New York, 1983. (chapter 7 on spatial intelligence).

[Gen83] Dedre Gentner. Structure-mapping a theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, April-June 1983.

[GNC95] Janice Glasgow, N. Hari Narayanan, and B. Chandrasekaran. *Diagrammatic reasoning : Cognitive and Computational Perspectives.* AAAI Press/MIT Press, 1995.

[Got94] N. M. Gotts. How far can we 'C'? defining a 'doughnut' using connection alone. In Pietro Torasso Jon Doyle, Erik Sandewall, editor, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 246–257, Bonn, FRG, May 1994. Morgan Kaufmann.

[Gou80] P. Gould. Q-analysis, or a language of structure : An introduction for social scientists, geographers and planners. *International Journal of Man-Machine Studies*, 13:169–199, 1980.

[GPP95] Michelangelo Grigni, Dimitris Papadias, and Christos Papadimitriou. Topological inference. In *Topological Inference*, pages 901–907, 1995.

[Ha91] J. P. Haton and al. *Le raisonnement en intelligence artificielle.* InterEditions, 1991.

[Ham97] Eric Hammer. Logic and visual information. *Journal of Logic, Language, and Information*, 6(2):213–216, 1997.

[Har95] David Harel. On visual formalism. In Janice Glasgow, editor, *Diagrammatic Reasoning.* AAAI Press/MIT Press, 1995.

[HE97] Kathleen Hornsby and Max J. Egenhofer. Qualitative representation of change. In D.Lukose, H.Delugach, M. Keeler, L. Searle, and J. Sowa, editors, *Spatial Information Theory*, volume 1257 of *LNCS.* Springer-Verlag, 1997.

[Hen94] M. Henle. *A combinatorial introduction to topology.* Dover publications, New-York, 1994.

[HHNT86] John H. Holland, Keith J. Holyoak, Richard E. Nisbett, and Paul R. Thagard. *Induction – Processes of Inference, learning, and Discovery.* The MIT Press, 1986.

[Hof84] D. R. Hofstadter. The Copycat Project: an experiment in nondeterminim and creative analogies. A.I. Memo 755, MIT Artificial Intelligence Laboratory, Cambridge Massachusetts, 1984.

[HY88] J. G. Hocking and G.S. Young. *Topology.* Dover publications, New-York, 1988.

[Joh91a] J. Johnson. *The mathematical revolution inspired by computing*, chapter The mathematics of complex systems, Johnson J. and Loomes M. eds., pages 165–186. Oxford University Press, 1991.

[Joh91b] J. Johnson. *Transport Planning and Control*, chapter The dynamics of large complex road systems, Griffiths J. ed., pages 165–186. Oxford University Press, 1991.

[Ler93] X. Leroy. *The Caml Ligth system release 0.6.* INRIA, September 1993.

[Lin95] Robert K. Lindsay. Images and inferences. In *Diagrammatic Reasoning*, chapter 4, pages 111–135. AAAI Press, 1995.

[LP97] Oliver Lemon and Ian Pratt. Spatial logic and the complexity of diagrammatic reasonning. *Graphics and Vision*, 6(1):89–109, 1997.

[moo]

[PLC91] J. Pimm, J. Lawton, and J. Cohen. Food web patterns and their consequence. *Nature*, 350:669–674, 25 April 1991.

[Shi91] Sun-Joo Shin. An information-theoretic analysis of valid reasoning with venn diagrams. In Jon Barwise et al., editor, *Situation Theory and its Applications*, volume Part 2. Cambridge University Press, 1991.

[THNG90] P. Thagard, K. J. Holyak, G. Nelson, and D. Gochfeld. Analog retrieval by constraint satisfaction. *Artificial Intelligence*, 46(3):259–310, December 1990.

[Val97] Erika Valencia. Un modèle topologique pour le raisonnement diagrammatique. Rapport pour le DEA Sciences Cognitives, LIMSI. See also `http://www.lri.fr/~erika/Pro/erika.html`, August 1997.

[Val98] Erika Valencia. *ESQIMO User Guide.* `http://www.lri.fr/~erika/Pro/erika.html`, 1998. Electronic document.

[VG98] Erika Valencia and Jean-Louis Giavitto. Algebraic topology for knowledge representation in analogy solving. In AISB ECCAI, editor, *European Conference on Artificial Intelligence (ECAI98)*, pages 88–92, Southampton, UK, 23–28 August 1998. Christian Rauscher.

[VGS98] Erika Valencia, Jean-Louis Giavitto, and Jean-Paul Sansonnet. Esqimo: Modelling analogy with topology. In Franck Ritter and Richard Young, editors, *Second European Conference on Cognitive Modelling (ECCM2)*, pages 212–213, Nottingham, UK, 1–4 April 1998. Nottingham University Press.

[Vic88] Steven J. Vickers. *Topology Via Logic*, volume 5 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1988.

[Wil97] Rudolf Wille. Conceptual graphs and formal concept analysis. In D. Lukose, H. Delugach, M. Keeler, L. Searle, and J. F. Sowa, editors, *Conceptual structures: Fulfilling Peirce's dream*, number 1257 in Lecture Notes in Artificial Intelligence, Berlin–Heidelberg–New York, 1997. Springer–Verlag.

[Wol88] Stephen Wolfram. *Mathematica.* Addison-Wesley, Redwood City, CA, 1988.

[WS90] S. H. Weber and A. Stolcke. l_0 : A testbed for miniature language acquisition. *International Computer Science Institute*, 1990.