1. A Topological Framework for Modeling Diagrammatic Reasoning Tasks

We propose to model some diagrammatic reasoning tasks within a topological framework. We only focus on problems that require a specific diagrammatic reasoning procedure to be solved rather than on reasoning on diagrams. To model such cognitive tasks, we propose to represent the problem with topological objects and to model the diagrammatic operations performed on them as topological operations. The idea underlying this proposition is that *combinatorial algebraic topology* is an adequate and unifying framework to specify and analyse diagrammatic representations and reasoning.

To illustrate such a proposal, we present here three applications of this topological framework: the first concerns a categorization problem, the second deals with hierachy restructuring, and the last one is the ESQIMO system for simple intra-domain analogy solving in unsupervised IQ-tests.

1.1 Introduction

Several issues are addressed in the fields of diagrammatic reasoning (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], logical formalisation of spatial relationships [Got94, LP97, Ham97, Ben94].

The last example accounts 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.

We are interested in reasoning on a problem with an internal diagrammatic representation of knowledge. More precisely, we are not interested in reasoning *on* diagrams but *with* (some kind of) diagrams on *various* kind of problems[1]. The formal framework that we explore here comes from combinatorial algebraic topology, and will be called CAT in this paper.

[1] We will not enter into the debate concerning the distinction of internal and external diagrammatic representations (see for example the discussion at TWD97). We only propose to explore a representational system in which we develop conceptual models of cognitive tasks.

Instead of rephrasing well-known diagrammatic applications in the CAT framework, we have found more illuminating to present a topological modeling of cognitive tasks that can be diagrammatically solved (they involves lattice graph and geometric configuration) but that have not received until now a specific diagrammatic treatment.

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

1.2 Algebraic Topology for Knowledge Representation

We were guided towards topological tools for several reasons. First, we are interested in diagrammatic reasoning as the use of spatial relationships such as neighbourhood, border, dimension, path, hole, ..., to represent and structure knowledge.

Although geometry studies these relations, we are not interested in the continuous and metric structure of geometrical objects. The primitive objects and relations involved in diagrams have a finite and discrete nature. For instance, a graph 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 does. The same remark holds for *Venn diagram*, *state-charts*, symbolic maps where it is only the configuration of finite sets of objects that is relevant. When metric aspects turn out to be important, they are often restricted to represent *partial order* relationships: A is bigger that B, C is closer from D than E, path F is shorter that path G, etc.

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

Hence, if we neglect quantitative diagrammatic representations (like barchart, geological survey map, etc.) we can focus on *n-dimensional combinatorial algebraic topology.* 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" objects, i.e. objects that can be tractable by a computer program.

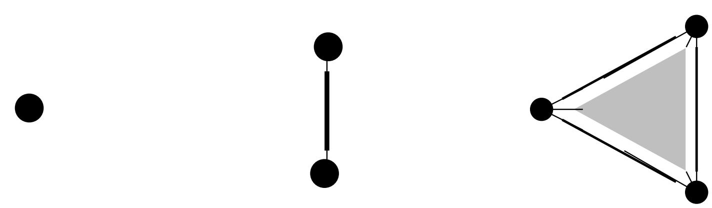
1.2.1 Simplicial Complexes (SC)

Simplicial complexes are topological abstract structures that generalise the notion of *graph*[Hen94, HY88]. Indeed, all complexes of dimension less than 2 are graphs. We find it interesting to consider some spatial properties of graphs and then generalise them to many dimension to express more information. Simplicial complexes are the abstract objects that realises this generalisation. The following definition is standard in algebraic topology.

Definition 1.2.1. *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.*

All p-complexes with $p < 2$ are graphs. Indeed, graphs are composed of edges and vertices of dimension 1 as shown on figure 1.1(b). Simplicial complexes are particularly attractive to generalise semantic networks by keeping the possibility to express hierarchies like in a relational graph [Hir97].

Definition 1.2.2. *Let $\alpha = (\sigma_0, \sigma_1, \ldots, \sigma_n)$ 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 couples (σ_i, σ_{i+1}), $\sigma_i \cap \sigma_{i+1} \neq \emptyset$. The dimension of α is the smallest dimension of $\sigma_i \cap \sigma_{i+1}$.*



(a) 0-simplex

(b) 1-simplex, it is also an edge in a graph.

(c) 2-simplex

Fig. 1.1. Geometrical representation of p-simplexes for p varying from 0 to 2.

A p-simplex s is noted: $s = \langle v^0 v^1 \ldots v^p \rangle$, where $v^i \in V$. The figure 1.1 shows the geometrical representation of 0, 1 and 2-simplexes. 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 itself with a 0-chain.

Now, we propose to use simplicial complexes to represent knowledge.

1.2.2 Representing Binary Relations with Simplicial Complexes: Q-analysis

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

Let Λ be the incidence matrix of a binary relation $\lambda \subset A \times B$. Let $a \in A$, and the set B_a of elements $b_i \in B$ such that $(a, b_i) \in \lambda$. The set B_a can be directly read from Λ, as the a-column (see table 1.1).

We represent the elements b_i of B_a as vertices and a as a simplex build on these vertices. The dimension of the simplex S_a representing a depends on the number of vertices in B_a.

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 1.2(a)).

Likewise, we represent Λ^{-1} with the dual simplicial complex $K_B(A, \lambda^{-1})$. In this case, the elements a_i are taken as vertices and the elements b_i are represented as simplexes (see figure 1.2(b)). We say that $K_A(B, \lambda)$ and $K_B(A, \lambda^{-1})$ are conjugates, they contain the same information but present it in a different and complementary way.

We extended Q-Analysis to allow the representation of sets of predicates as a simplicial complex too [Val97]. We take a set of predicates $P = \{p_1, p_2, \ldots, p_n\}$ and represent the binary relation $\mu \subset A \times P$ such that $(a_i, p_j) \in \mu$ 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 $P = \{p_1, p_2, p_3, p_4\} = \{\text{parity}, \text{oddity}, \text{primality}, \text{multiple of } 3\}$. The incidence matrix of μ is then obviously the one given in figure 1.2.2. We can represent the dual complex of μ, each element $a_i \in A$ being a simplex build with vertices $p_i \in P$. This dual representation enlighten the fact that elements $4, 8, 10$ have exactly the same representation when taking these few predicates.

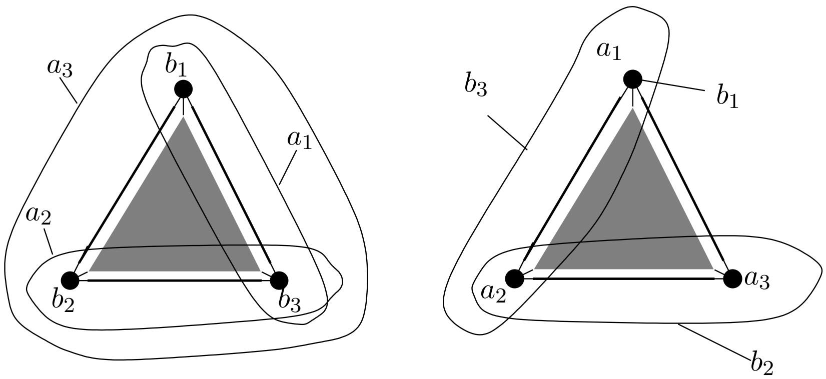
A representation based upon simplicial complexes associates the same simplex to elements of A that cannot be distinguished. In other words, two elements will be separated only if there is at least one predicate that allows the differentiation. The same situation occurs with the dual complex.

Two simplexes that have a smaller k-simplex in common are said to share a k-face. In terms of representation, it means that *they have k features in common*. As Freska emphasised it, we call here for the use of discriminating features rather than for precise characterisation in terms of universally applicable reference system [Fre97].

We can say that the identity of an element is represented by the features he shares with others and also by the ones that are specific to it [HE97].

Table 1.1. Incidence matrix associated with λ. The elements $b_i \in B$ that are λ-related to a_j can be directly read from the matrix as the a_j-column (the j^{th} column).

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



(a) Simplicial representation of λ taking b_i as vertices and a_i as simplexes. 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.

(b) Dual simplicial representation of λ taking a_i as vertices and b_i as simplexes.

Fig. 1.2. Simplicial representation of the binary relation λ using column (a) or row (b) of the incidence matrix as simplices.

	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

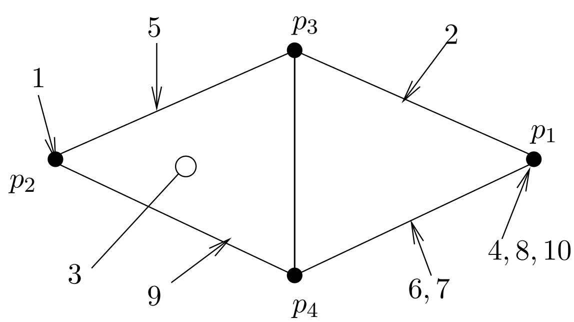


Fig. 1.3. Incidence matrix associated with μ in the numbers example and dual complex associated with $\mu \subset \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \times \{p_1, p_2, p_3, p_4\}$ where we can see that the integers $4, 8$ and 10 are identical with respect to these criteria.

1.3 A Categorisation Problem

Holland [HHNT86] gives a simple model of the process of categorisation for the construction of a *homomorphic* representation that maps many elements of the world to one element of the representation. We present now the construction of a simplicial representation by a categorisation task according to Holland's model.

Let C be the categorisation function that maps the states of the world onto a smaller number of categories. The categorisation is made with the detection of the states of the world through detectors. Let $d_1, \ldots, d_n$ be the binary detectors, that can take the value 0 if they are off or 1 if they are on.

When a state S_1 is perceived, the detectors take with values 0 or 1. We can represent the values of the detectors for this state by the vector $V_1 = (V_1^{d_1}, \ldots, V_1^{d_n})$ of length n where $V_1^{d_1}$ is the value of d_1 and so on. Then V_1 represents the state S_1.

We can now consider many successive states $S_1, \ldots, S_p$ and their encoding into binary vectors $V_1, \ldots, V_p$ of length n. If we write the vectors representing this list of states, we construct the matrix of table 1.2 of the relationship ν between the detectors and the states.

Table 1.2. Incidence matrix of the relationship ν between the n detectors of a system and a succession of p states detected and encoded as vectors V_i.

	d_1	$\ldots$	d_n
V_1	1	$\ldots$	1
$\ldots$	$\ldots$	$\ldots$	$\ldots$
V_p	1	$\ldots$	1

Starting from this matrix, we can now build a simplicial representation of the states encoded. Indeed, we build the matrix by writing the lines V_i corresponding to the encoding of each state S_i, but we can now see that each detector has a representation as a column of the matrix. Thus, each detector, that detects a particular feature, can be represented as a simplex. The representation of the whole matrix as a complex and its dual representation, will show the categories extracted through this perception. Indeed, two states undistinguishable by the detectors will be represented as equivalent.

Analysing the *Little Red Riding Hood* Tale

We illustrate now this construction model with a concrete example. We try to extract an ontology from the perception of the successive states that describe the *Little Red Riding Hood* tale.

To represent the objects of the *The Little Red Riding Hood*, we chose for example the detectors: *alive, animal, good, bad, place, small, several, motor,* *exterior, interior* that encode the principal characters, objects and concepts of the story according to table 1.3 (left). 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.

Note that, this analysis is held at a naive level. The story can be told for example in the 11 states of the world presented table 1.3 (right), also called images.

Several strategies are possible to extract the simplexes considered as categories, we implemented two [Val97] in the `Mathematica` [Wol88] programming language. The first one extracts concepts *incrementally* from the first image to the last one. This means that a base of categories is extracted from the first image. Then if this base is not sufficient to express the second image as a linear combination of the simplexes of the base, we add to the base the simplexes necessary to express it and so on. This is done with a `Mathematica` function, that we called `Incremente`[2]. We illustrate briefly its functioning with an abstract example before using it on the images of the tale.

If we take the abstract sequence `l` of images:

```
l = {{1, 2, 3},  {3, 4},  {5, 6},  {1},
      {1, 2, 3, 4},  {3, 4, 5, 6},  {1, 2, 3}}
```

where an image is written between { } and the story itself, composed of images is also written between { }. When we ask for the *incremental* base, we get the simplexes:

```
Incremente[l]
cSimplex[{1, 2, 3},  {3, 4},  {5, 6},  {1}]
```

The base is expressed with simplexes (a list with head `cSimplex[]`). The first elements of the base is the first image itself, since it is perceived alone with no "history", and thus no base to express it. When the second, third and fourth images are perceived, they are also entirely added to the base since they are necessary to express themselves. But then, these simplexes are sufficient to express the last three images as intersections and unions of the previous ones.

The other strategy builds immediately a taxonomy from the 11 images detected as a whole. This means that we get a minimal basis necessary to express all the categories. This strategy is implemented with the function `SimplexBase`. If we take the same succession of images `l`, we will not get the same base:

```
SimplexBase[l]
cSimplex[{1},  {2},  {3},  {4},  {5,6}]
```

where only the objects 5 and 6 can not be distinguished since they appear together each time they appear in an image. For all the other objects, there is an image (a state) that makes possible their distinction by the detectors.

[2] The function names are in French.

Table 1.3. *Left*: Detectors encoding the objects of the world. *Right*: The Little Red Riding Hood told in 11 scenes or states of world.

Objects	Encoding
Red	alive, good, small
Humans	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

	Detectors activated
1.	Red, Mother, talk, house
2.	Red, Mother, give, basket, house
3.	Red, walk, tress, basket
4.	Red, Wolf, talk, trees, basket
5.	Red, walk, trees, basket
6.	Wolf, walk, trees
7.	Grandma, sleep, house, bed
8.	Grandma, Wolf, talk, house, bed
9.	Wolf, eat, house, bed
10.	Red, Wolf, talk, house, basket, bed
11.	Wolf, eat, house, bed, basket

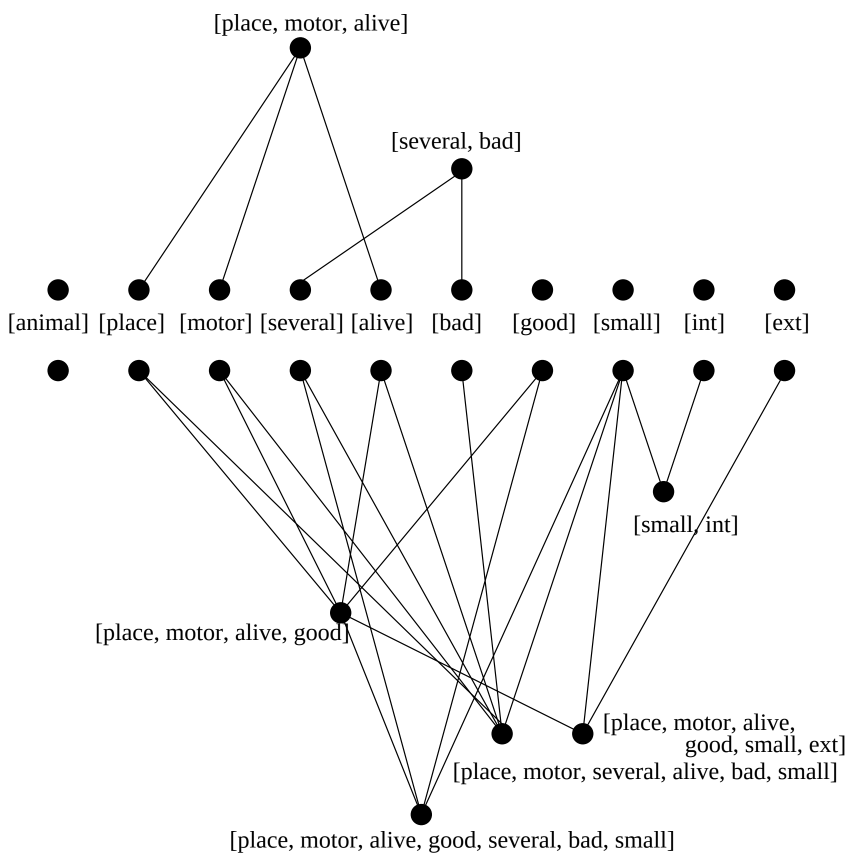


Fig. 1.4. Two ontologies extracted from the Little Red Riding Hood story. The ontology extracted instantaneously is represented top-down on the higher part of the figure; the one extracted incrementally is represented bottom-up on the lower part of the figure.

The incremental and instantaneous ontologies extracted from the Little Red Riding Hood 11 images are given in figure 1.4 where we can only see the maximal simplexes represented in a Hasse diagram[3]. In this representation, each point is a simplex and the vertices represent inclusion relationships. The concepts represented at level n, are ontologically precedent to the ones represented on the level $n + 1$ (the atomic simplexes of the inner layers are at level 0).

1.4 Inheritance Restructuring

We present now an algorithm for *inheritance hierarchy restructuring* proposed by [Moo96] 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 formalised. The CAT framework provides a concise and clear language to specify this algorithm and exhibit its diagrammatic nature.

We will call *features* any property, behaviour, instance variable or method that can be used for the description of objects. A class corresponds to the description of a type of objects sharing a set of features. Using inheritance to specify classes, we express explicitly the hierarchy relationships between the classes.

Moore [Moo96] 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 be a class corresponding to each concrete object (see. fig 1.5). Further criteria must be specified to constrain the possible hierarchies:

1. Every feature should appear in only one class (maximal sharing of features between classes).
2. Minimal number of classes.
3. All inheritance links that are consistent with the objects structure must be present.
4. The number of explicit inheritance links must be minimised
5. The concrete objects should correspond to leaves of the inheritance hierarchy tree.

These criteria all together are sufficient to specify a unique solution as showed by Moore in [Moo96].

[3] The extraction of categories from the Little Red Riding Hood is being re-thinked more rigorously and applied to the analysis of hypertexts structures. See the web sites `http://www.lri.fr/~giavitto/UTopoIa` and `http://www.limsi.fr/Individu/erika` for future developments.

The problem of inferring a hierarchy from a set of concrete objects can now be rephrased into the CAT framework. We represent the features by vertices, and the classes by different simplexes built with the vertices corresponding to the features that define the class. The inheritance relation of classes in the hierarchy is then simply modeled as the inclusion relation of the simplexes. Finally, the inheritance graph is the minimal complex containing all the representations of the classes.

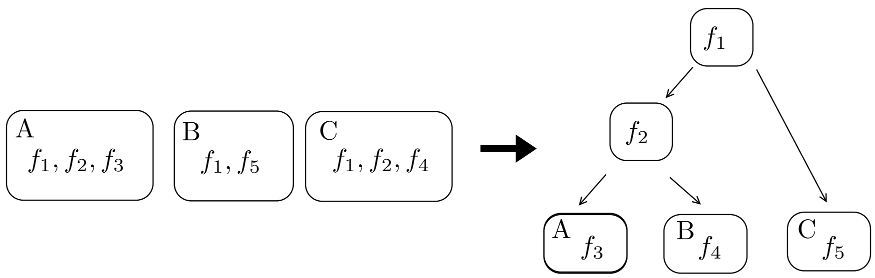


Fig. 1.5. From a collection of concrete objects to an inheritance hierarchy. A set of concrete objects (A, B and C) is given on the left. A possible hierarchy that accepts this set of objects is given on the right. The features linked to a class are the set of features defined for this class merged with the features inherited (recursively) from the parent classes. A class without name is called an *abstract class* in the object oriented programming terminology [Boo91] and corresponds to an internal node of the inheritance graph (hierarchies with multiple inheritance will be graphs rather than trees [dBY95]).

The five criteria used by Moore to constrain the class hierarchy are now *topological constraints* that have a simple and intuitive meaning. The corresponding topological constraints are respectively:

1. Every feature appears in a distinguished simplex.
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.
5. Concrete objects are simplex of maximal dimension.

The problem of inferring an inheritance hierarchy is now simply to find simplexes satisfying the previous properties in the complex made by the concrete objects.

1.5 Analogy Solving with the ESQIMO System

We explore now the possibility of a topological representation to support analogy [VGS98]. 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 problems, *non supervised* IQ-tests. 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.

1.5.1 Diagrammatic Representation of the problem

Usually, IQ-tests are given in terms of geometrical elements so that they can express many different properties at the same level and still stay simple. We chose a geometrical universe similar to the one investigated in [WS90] of twelve basic elements $E = \{e_1, \ldots, e_{12}\}$, as shown on figure 1.6(a). These elements are all the possible combinations of the seven properties (or predicates): $P = \{p_1, \ldots, p_7\} = \{$round, square, triangle, white, dark, big, small$\}$.

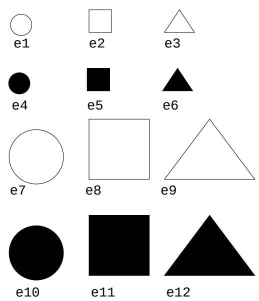
These two sets are the only knowledge used by ESQIMO to solve the tests. We can represent this knowledge with a simplicial complex $K(\Omega)$ or its conjugate $K'(\Omega)$ (see figure 1.6(b)) by representing the binary relation $\lambda \subset A \times P$ such that $(a_i, p_j) \in \lambda$ if $p_j(a_i)$ holds.

1.5.2 Algorithm based on a SC Representation

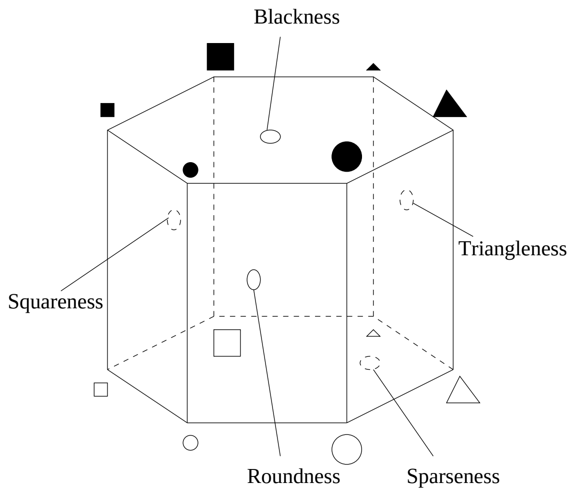
When a problem is presented, each figure A, B and C is composed of one or more elements $e_i \in E$. Each element e_i can be represented as a simplex of $K(\Omega)$, the properties p_j such that $p_j(e_i)$ holds, being its vertices. Thus, a simple figure (composed of only one element) will be represented as a simplex and a composed figure (more than one element) will be represented with a set of simplexes. The problem is now to find a relation between the (set of) simplex(es) representing A and the (set of) simplex(es) representing B and apply it to the (set of) simplex(es) representing C.

Case of simple figures. In the case of simple figures, the transformation T_{AB} is seen as a polygonal chain from S_A to S_B in $K(\Omega)$. 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: $T_{S_l,S_B} \circ \ldots \circ T_{S_A,S_1}$.

If there are several chains, then we say that there are several possible relations between A and B. We can select a *best* solution giving a higher



(a) Elements of the universe Ω of ESQIMO, respectively called e_1 to e_{12} starting from the top left element.



(b) A 2D view of the dual complex $K'(\Omega)$, the elements of E are the vertices and the properties $p_i \in P$ are simplexes of $K'(\Omega)$. Notice that the 6-simplex representing the property of blackness is normally 5-dimensional.

Fig. 1.6. Elements manipulated by ESQIMO and their representation as a simplicial complex.

priority to polygonal chains that are short and of higher dimension, to choose a transformation that requires less steps and that preserves more properties.

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 $T'_{AB}(S_A) = S_B$, T' is then a *simplicial application* [Hen94]. There are different possible strategies to determine the domain of $S(C)$ on which we can apply T_{AB}, and we implemented three of them, presented in [Val98] (unfortunately, we do not have enough space to develop them here).

Case of composed figures. 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 solution [Val98]. The only constraint we need is that all the vertices and faces of $S(B)$ are paired with vertices from $S(A)$.

1.5.3 Examples of Analogy Solving with ESQIMO

We give three examples of IQ-test solving with ESQIMO on figure 1.7. In the first example, we ask ESQIMO to solve the IQ-test with the call of the function `Resolve` with the pairing parameters `App2` and `AppApp2` (they are pairing strategies) as shown below (for more details see [Val98]). The three given figures A, B and C are defined in terms of e_i elements of E. As seen on the first example of figure 1.7, A is composed of a white small circle plus a white small square.

```
A={e1,e2};  B={e7,e5};  C={e3,e1};
Resolve[A,B,C,App2,AppApp2]
```

Here, A is a composed figure, its representation corresponds to the set of simplexes $S_A = \{\langle p_1, p_4, p_7\rangle, \langle p_2 p_4 p_7\rangle\} = \{\langle S_A^1, S_A^2\rangle\}$. Likewise, the representations of B and C are respectively, $S_B = \{\langle p_1, p_4, p_6\rangle, \langle p_2 p_5 p_7\rangle\} = \{\langle S_B^1, S_B^2\rangle\}$ and $S_C = \{\langle p_3, p_4, p_7\rangle, \langle p_1 p_4 p_7\rangle\} = \{\langle S_C^1, S_C^2\rangle\}$.

`App2` is a strategy for the pairing between the set of simplexes of A, and the set of simplexes of B that gives the following pairing: $(S_A^1 \to S_B^1), (S_A^2 \to S_B^2)$ ESQIMO gives output about intermediate results such as pairings, the result of applying strategy `App2` is given by the following output:

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

where an `AssocSet` is a set of pairings and `FromTo` is a pairing, which means also an elementary transformation `From` the first element of the pair `To` the second one. For each paring, an elementary transformation is proposed, depending on the heuristic used which is another parameter (that is internally settled until now [Val98]). We call them respectively T_1 and T_2. Then, the pairing strategy `AppApp2` is used to apply these elementary transformations to the elements of the set of simplexes representing C, it proposes to apply in parallel: $T_1(S_C^1)//T_2(S_C^2)$ The corresponding output is:

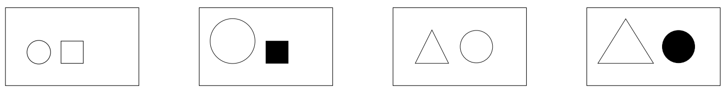
```
Par[Domain[1,Seq["D-elem"[SmallQ->0,BigQ->1]],{e3}],
    Domain[2,Seq["D-elem"[WhiteQ->0,BlackQ->1]],{e1}]]}
```

where `Par` means a parallel application and `Seq` a sequential application of the elementary transformation described in terms of change of properties (or predicates). Finally the solution is composed of two elements represented by the simplexes $S_D = \{\langle p_3, p_4, p_6\rangle, \langle p_1, p_5, p_7\rangle\} = \{e_9, e_4\}$ (see figure 1.7 first example), the corresponding output is:

```
Choose[{e9,e4}]
```

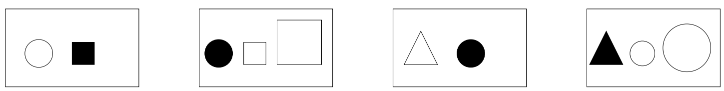
All along the solving process, ESQIMO uses the prefix `Choose` in all its outputs. That is because many different solutions are possible and acceptable for a psychological plausibility. ESQIMO can compute many solutions in parallel without selecting a *best* one, in that case there are many solutions that the user can `Choose` at the end.

The two other examples are solved with the same pairing strategies and are not detailed here.



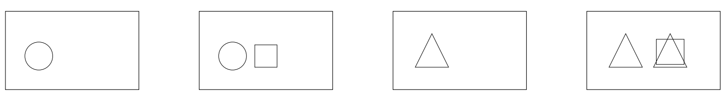
(a) (b) (c) (d)

The first element becomes bigger and the second becomes black.



(e) (f) (g) (h)

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



(i) (j) (k) (l)

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.

Fig. 1.7. Three examples of analogy resolution with ESQIMO.

Many choices made in ESQIMO's algorithm can be discussed. In fact, they can be seen as additional strategies parameterizing the ESQIMO kernel. For example:

- The description of the properties of each figure in terms of predicates can be a problem for properties such as position. We could give each possible position a predicate that could be true or false.
- 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 higher-order structures between the sets of simplexes.

- 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. Other measures of satisfaction can be tested.

Furthermore, note that our formalization of IQ-test problems does not depend on their geometrical nature. Indeed, only the representational level is based on topology while the objects manipulated by the system could have been non geometrical. We could, for example try ESQIMO on verbal IQ-tests more like in the Copycat system [Hof84].

Even if the ESQIMO system can be considered as very simple, we are convinced that a topological representational structure is well-adapted to support 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.

1.6 Conclusions

The idea developed here is that *combinatorial algebraic topology* is an adequate and unifying framework to specify and analyse diagrammatic representations and diagrammatic reasoning.

It is important to notice that we have only used *elementary* CAT notions: simplicial complexes generalise the concept graph and polygonal chains extend the concept of path. These two notions have an immediate and intuitive meaning, even in higher dimensions and are obviously diagrammatic. Future work must include the use of further CAT constructions (like simplicial applications, homotopy group, homology classes, etc.) to handle more sophisticated diagrammatic situations.

References

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

[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.

[Boo91] Grady Booch. *Object Oriented Design*. Benjamin/Cummings Publishing Company, 1991.

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

[dBY95] Jean daniel Boissonnat and Mariette Yvinec. *Gomtrie Algorithmique*. Ediscience International, 1995.

[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.

[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.

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

[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.

[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.

[Hir97] Stephen C. Hirtle. Representational structures for cognitive space : Trees, ordered trees and semi-lattices. In D.Lukose, H.Delugach, M. Keeler, L. Searle, and J. Sowa, editors, *Spatial Information Theory*, volume 1257 of *LNCS*. Springer-Verlag, 1997.

[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.

[Moo96] Ivan Moore. A simple and efficient algorithm for onferring inheritance hierarchies. In *TOOLS 96*. TOOLS Europe, Prentice-Hall, 1996.

[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.

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

[Val98] Erika Valencia. Hitch hiker's guide to esqimo. RR 1173, LRI, ura 410 CNRS, Universit Paris-Sud, 91405 Orsay, France, May 1998.

[VG98] Erika Valencia and Jean-Louis Giavitto. Algebraic topology for knowledge representation in analogy solving. In ECCAI, editor, *European Conference on Artificial Intelligence (ECAI98)*, pages 88–92, Brighton, 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.

[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.