PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 02/18

NOTES ON ARGUMENTS: FROM SYLLOGISM TO NETWORK ORGANIZED REASONING

Antonio L. Furtado

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900
RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 02/18 ISSN 0103-9741
Editor: Prof. Carlos José Pereira de Lucena July, 2018

Notes on Arguments: from Syllogism to Network Organized Reasoning

Antonio L. Furtado

PUC-Rio, Departamento de Informática, Rio de Janeiro, Brasil
furtado@inf.puc-rio.br

Abstract: This work is a study on how *arguments* can be formulated to achieve a better understanding of the *factual description* provided by a given information system, and to discover alternative sequences of *narrative events* able to lead to a desired target state of the system. Starting from Aristotle's syllogism, we proceed by examining problem solving strategies associated with four *semiotic relations*, which constitute the determinants of information spaces. Next, as a preliminary step to help passing from arguments about facts to arguments concerning events, a prototype is introduced which is able to translate a conceptual specification – involving facts, events and agents – into a relational database implementation equipped with a **Log** of the executed events. Developed as a logic programming tool with access to Oracle via an ODBC interface, the prototype follows a *plan recognition / plan generation paradigm.* Within the environment created by the prototype, it becomes possible to extract alternative *typical plans*, as a process mining task. A *network organized reasoning* approach is finally proposed as a powerful instrument to analyze the typical plans thus obtained, and trace new plans by traversing event sub-sequences taken from different original alternatives. The same approach is claimed to be applicable to the universe of storytelling, in special to interactively compose new genre-consistent stories by combining existing variants of a folktale.

Keywords: Arguments, Syllogism, Abduction, Problem Solving, Semiotic Relations, Conceptual Schemas, Story Bases, Process Mining, Network Organized Reasoning.

Resumo: Este trabalho é um estudo sobre como se pode formular *argumentos* para atingir uma compreensão melhor da *descrição factual* fornecida por um dado sistema de informação, e para descobrir sequências alternativas de *eventos narrativos* capazes de conduzir o sistema a um desejado estado-alvo. Partindo do silogismo de Aristóteles, prosseguimos examinando estratégias de resolução de problemas associadas a quatro *relações semióticas*, que constituem os determinantes de espaços de informações. Em seguida, como um passo preliminar para ajudar a passar de argumentos sobre fatos a argumentos voltados para eventos, é introduzido um protótipo capaz de traduzir uma especificação conceitual – envolvendo fatos, eventos e agentes – em uma implementação em banco de dados relacional equipada com um **Log** dos eventos executados. Desenvolvido como ferramenta de programação em lógica com acesso a Oracle via uma interface ODBC, o protótipo segue um *paradigma de reconhecimento de planos / geração de planos*. Dentro do ambiente criado pelo protótipo, torna-se possível extrair *planos típicos* alternativos, como tarefa de mineração de processos. Uma abordagem de *raciocínio organizado em rede* é finalmente proposta como um instrumento poderoso para analisar os planos típicos assim obtidos, e para traçar novos planos percorrendo subsequências de eventos tiradas de diferentes alternativas. A mesma abordagem é tida como aplicável ao universo de narração de estórias, especialmente para compor interativamente novas estórias gênero-compatíveis através da combinação de variantes existentes de um conto folclórico.

Palavras-chave: Argumentos, Silogismo, Abdução, Resolução de Problemas, Relações Semióticas, Esquemas Conceituais, Bancos de Estórias, Mineração de Processos, Raciocínio Organizado em Rede.

In charge of publications

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br

1. Introduction

Information systems specified over a given mini-world involve, first of all, a database component whose *state* at a given time **T** consists of the set of *facts* holding at **T**. But *events* can happen, in particular with the intervention of *agents*, through which new states are reached. Thus, while interacting with a given system, we need to understand both its factual *description* and what *narratives* can emerge during its lifetime. In special, we need to find what permissible actions to perform in order to reach states whereat our objectives are fulfilled. And we expect that methodical *arguments* should give us an effective guidance.

The present work reviews some basic notions about how arguments can be formulated, starting from Aristotle's *syllogism*, a simple scheme that allows to extend our knowledge of the current facts by applying *inference rules*. Next, we shall consider somewhat more complex schemes, and shall examine different ways to reason with facts and rules.

For problem solving, more than one application of such schemes may be needed, and suitable reasoning strategies are necessary to organize these applications towards a solution. The effectiveness of the four strategies to be examined here is shown to derive from the characterization and exploitation of four *semiotic relations* (associated with the *four master tropes* [Burke] of rhetorical tradition) according to which information systems can be viewed as bounded three-dimensional spaces. This view leads to superimpose the ampler notion of *semiotic completeness* over and above the classic notion of database *relational completeness*.

To ease the transition from descriptions to narratives, i.e. from *data bases* to *story bases*, a logic programming prototype was developed, which converts, along three successive stages, an executable conceptual specification – defining the existing classes of facts, the restricted repertoire of event-producing operations, and the predicted behavior of the prospective agents – into a relational database implementation augmented with a **Log** that registers the execution of the operations. Conceived in the mold of a *plan-recognition / plan-generation paradigm*, the prototype supports both workspace simulation and regular execution over the database files. Also, having access to the **Log** and keeping available online the conceptual schemas, the prototype is particularly well equipped for data mining and, more importantly, for *process mining* tasks.

One most relevant process mining task enabled by the prototype consists of extracting *traces* from the **Log**, aligning any number of executed event-producing operations ultimately aiming at a given goal. Filtering these traces, one is able to discover *typical plans* to achieve the goal. Further, by combining such sequences into a network it is possible to condense coinciding subsequences, as well as to use the network's branching structure to represent where the alternative plans converge or diverge. We claim that this *network organized reasoning* approach is most useful, not only for analyzing the alternatives but also for devising new plans by choosing a path that traverses subsequences of different original plans. The generality of this approach is confirmed by its suitability to the seemingly alien universe of storytelling, serving, for example, to condense in the form of a network different variants of a folktale, thereby allowing to interactively compose new stories that remain consistent with the intended genre.

The remainder of the text is organized as follows. Section 2 is devoted to Aristotle and to extensions of his momentous creation. Section 3 looks at problem solving, trying to capture the intuition of mathematicians and then to apply it to information system domains. Section 4 treats our proposed semiotic relations. Section 5 moves the discussion from facts to events and story bases. Section 6 stresses network organized reasoning as a powerful process mining and plan composition resource. Concluding remarks are presented in section 7.

2. Reasoning from facts and rules

2.1. Aristotle's syllogism and Toulmin's extension

The Greek philosopher Aristotle (384–322 BC) is famously credited as the proposer of *syllogism* [Aristotle] as a sound form of reasoning. Given a commonsense *rule* such as "All humans are mortal" and given as *antecedent* the fact "Socrates is human", one is justified to accept, as a *consequent*, the fact that Socrates is mortal.

In first-order predicate logic notation, this simple syllogism can be expressed as follows:

<u>rule</u>: $\forall X\ (\text{human}(X) \rightarrow \text{mortal}(X))$
<u>antecedent</u>: human(Socrates)
<u>consequent</u>: mortal(Socrates)

However not all rules are that simple. For certain rules the consequent is not necessarily entailed by the occurrence of the antecedent; the degree of uncertainty can then be expressed by words such as “often”, “sometimes”, “presumably”, etc., or by assigning a statistical probability value. In addition, the rule may admit exceptions whose occurrence would render the consequent inapplicable. More fundamentally, the rule’s validity itself may not be so obvious as is the mortality rule that plagues humanity; thus, when invoking a rule, one may be called to explain on what the rule is based. Judicial laws and enterprise policies, for example, must have been promulgated somehow.

Such considerations led Stephen Edelston Toulmin (1922 – 2009), a British philosopher, to expand the syllogism by adding three new components [Toulmin]: the *qualifier*, to express the degree of certainty; the *rebuttal*, to expose the exceptions; the *backing*, to supply the basis of the rule. The three Aristotelian components are of course maintained, with different names: *warrant* – rule; *grounds* – antecedent; *claim* – consequent. Figure 1 shows an example, with an arrow to associate the grounds with the claim, and appropriate connectives to introduce the other components: **since** – warrant; **on account of** – backing; **so** – qualifier; **unless** – rebuttal.

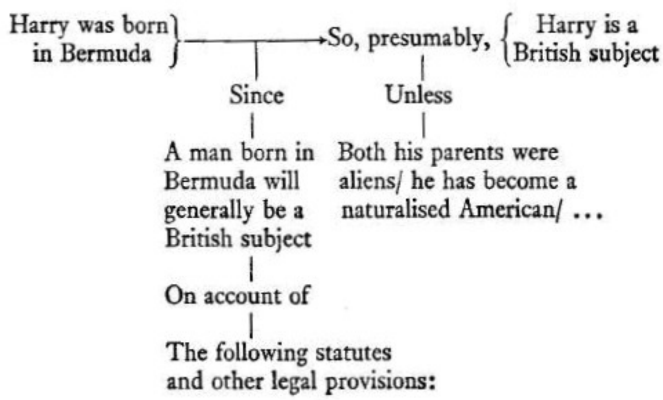


Figure 1: Toulmin’s expanded syllogism

2.2. Aristotle’s false inference rehabilitated and renamed: abduction

Now let us wonder about what would look like an inverted form of reasoning. Suppose we learn that the consequent of some rule has occurred. Could we conclude that the antecedent has also occurred? It has been a tradition among logicians to mark that as a *fallacy* to be avoided. Aristotle himself anticipated this judgement, calling it a “false inference”, and yet, when treating literary rather than logic matters, he was able to point out an intriguing use of the fallacy, attributing it to Homer’s wily Odysseus (Ulysses) [Aristotle]:

> Above all, Homer has taught other poets how to tell untruths as they ought to be told, that is by **false inference**. If one thing exists or happens because another thing exists or happens, people think that, if the **consequent** exists or happens, the **antecedent** also exists; but this is not the case. Thus if a proposition were untrue, but there was something else which must be true or must happen if the proposition were true, the poet should supply the latter; for because we know that this is true, our soul falsely infers the truth of the original proposition. There is an example of this in the bath scene in the *Odyssey*.” (Note: In the bath scene the disguised hero tells his wife, Penelope, that he has seen Odysseus, and she believes him because he describes his appearance accurately).

A more recent literary character created by the English writer Arthur Conan Doyle (1859-1930), the self-proclaimed “consulting detective” Sherlock Holmes, revealed that this form of reasoning was the secret of his professional success. This is how he explains it to his friend and chronicler, Dr. Watson, in *A Study in Scarlet*[1]:

> “Most people, if you describe a train of events to them, will tell you what the result would be. They can put those events together in their minds, and argue from them that something will come to pass.

[1] http://www.gutenberg.org/files/244/244-h/244-h.htm

There are few people, however, who, if you told them a **result**, would be able to evolve from their own inner consciousness **what the steps were which led up to that result**. This power is what I mean when I talk of **reasoning backwards**, or analytically."

One of the initiators of Semiotics, the American philosopher Charles Sanders Peirce (1839-1914), contributed decisively to put this notion in a truly rational perspective [Peirce]. He distinguished three fundamental ways of relating an antecedent **A**, a consequent **C**, and a rule **R**. In an early publication he affirmed that, although the third way, which coincides with Sherlock's preferred method, could not lead to a necessary conclusion, it should not be discounted as a fallacy, since it allowed to formulate a likely *hypothesis*. Curiously, while tracing the discussion back to Aristotle, Peirce did not refer to the "false inference" attributed to Homer in the passage that we cited, but to the Greek term *apagogue* (απαγωγή), which figures in one of the philosopher's logic treatises. Peirce himself later coined for this hypothesis-generator scheme the now universally consecrated term *abduction*, and stressed its importance as a major source of creativity, worthy to occupy a prominent position among three complementary forms of reasoning:

Deduction: given a rule **A → C**, if **A** is known to be true, one concludes that **C** is also true.
Induction: if **C** is observed to be true whenever **A** is true, one may tend to assert **A → C** as a rule.
Abduction: given a rule **A → C**, if **C** is known to be true, one can postulate **A** as a hypothesis.

In fairness to Sherlock Holmes, we must recognize that his "reasoning backwards" remark already provided a visual impression of how abduction works, by an inverse traversal of the arrow in the rule **A → C**, thus proceeding from **C** to **A**. Indeed, criminal investigators usually start from the observed **clues** (consequent), left by the culprit, towards their prior undisclosed origin, namely the **crime** (antecedent) of which the detective is led to suspect.

Medical doctors are similarly forced to reason from consequent to antecedent, when they rely on observed **symptoms** (consequent) to discover the causing **illnesses** (antecedent). Their daily work with *differential diagnosis* provides a good illustration of how hypotheses can be checked in an attempt to eventually reach a trustworthy conclusion. A crucial task in differential diagnosis is to eliminate from the set of possibilities any illnesses causing symptoms that are *not* confirmed. This equally legitimate form of deductive reasoning, called *modus tollens*, can be thus expressed:

Modus Tollens: given a rule **A → C**, if **C** is known to be false, one concludes that **A** must also be false.

For example, letting a1 and a2 be illnesses and c1 and c2 be symptoms, assume as valid the rules below:

a1 → c1
a2 → c1
a2 → c2

Now, if symptom c1 is observed:
- its cause can possibly be a1 or a2 (two hypotheses obtained by abduction)
- a medical examination is ordered to check the occurrence of symptom c2
- if symptom c2 does not occur, then illness a2 can be excluded (by *modus tollens*)
- but, even then, the occurrence of a1 would be certain **only if** the rule c1 → a1 also held

In practice, physicians often end up deciding tentatively for one single illness, even in the absence of if-and-only-if rule pairs, when all but one hypotheses are invalidated. Doing that, they may perhaps be obeying Sherlock's injunction in *The Sign of the Four*[2]: "when you have eliminated the impossible, what remains, however improbable, must be the truth."

2.3. Exploring the three ways to associate facts with rules

Turning to computer science applications, all these three ways to reason over facts and rules have multiple uses, and bring several subjects to mind that are of interest to our present research, such as:

[2] http://www.gutenberg.org/files/2097/2097-h/2097-h.htm

- **Deduction** – logic programming, Prolog, Datalog, intelligent systems
- **Induction –** statistic correlation, knowledge discovery, data and process mining
- **Abduction –** formulation and test of hypotheses, speculation, conjectures

About induction, whose purpose is to find the applicable rules, it is opportune to recall Toulmin's *backing* component, providing the basis of the rule. The tendency to assert a rule **A** → **C** merely because one has noticed that occurrences of **A** are repeatedly followed by occurrences of **C**, without offering any sort of justification, corresponds to the *post hoc ergo propter hoc* ("after this, therefore because of this") so common fallacy. Causality must be based on some non-arbitrary determinant, such as a law of nature, or a judicial law, or a business policy, or at the very least on a regularity that admits some commonsense explanation. And the frequency of the joint occurrences of **A** and **C** should be measured in order to assess whether it is significant enough to justify the rule (by reaching some minimum threshold), typically in terms of statistical probability.

As to abduction, academic research that does not go beyond hypothetical formulations, alleging that, at the present state of the art, a rigorously proven conclusion cannot be achieved, is usually criticized under the derogatory label of "speculation". But speculation has frequently happened to be the first step in a successful quest for knowledge.

Under the more respectable label of "conjecture", logicians and mathematicians do not hesitate to publicize what they believe to be true, provided that, until then, no counter-examples have been found. Some may even risk to prematurely announce a result, moved by what psychologists call *confirmatory* or *confirmation bias* [Oswald]. The wish to be celebrated as the first to prove, for instance, Fermat's conjecture or the four colors conjecture, could easily predispose a naïf scholar to believe, uncritically, in what would turn out to be an erroneous or incomplete "proof". Fortunately, both these classic conjectures have ultimately been upgraded to theorems, after many years of obstinate research effort. These two examples, besides a few others also taken from the domain of mathematics, will be treated next, as an introduction to a more general discussion of problem-solving.

3. Problem-solving strategies

3.1. Theorem proving

Proving a theorem may require more than the application of a single rule. We claim that there exist only four distinct proof strategies. As illustration, consider the following mathematical theorems:

Pythagoras theorem[3]: In a right triangle, the square of the hypotenuse is equal to the sum of the squares of the other two sides. The demonstration uses what we may call a ***chaining*** strategy, i.e. a sequence of axioms and lemmas are applied to deduct successive intermediate results, until a final result is reached which completes the proof of the theorem. For this particular theorem, the required propositions include lemmas to measure the area and determine the congruence of triangles. A mathematician usually exposes how the theorem can be proved as a series of deductions. However, this *forward* chaining process may not reflect how the proof was discovered – possibly the final derivation was the first to occur in the mind of the discoverer, whereas the last to be worked out were those derived from the postulated basic axioms. Indeed, one method particularly suitable for automatic theorem-proving and logic programming (as in Prolog implementations), is *resolution* [Chang], which applies *backward chaining*.

Fermat theorem [Faltings, Wiles]: No three positive integers a, b, and c can be found to satisfy the equation $a^n + b^n = c^n$ for any integer value of n greater than 2. Fermat himself formulated the theorem in 1637 without registering a proof, which was established by Andrew Wiles in 1994. Since early attempts in the original domain of number theory had been tried with no result, Wiles invested on a line of work initiated by other researchers that involved a different domain. So the successful strategy was problem solving by ***analogy***, in this case by proving a demonstrably analogous modularity theorem for semistable elliptic curves. To apply analogy between the original problem domain (called the *source* domain) and another conveniently chosen domain (the *target* domain), it is usually necessary as a preliminary task to characterize *mappings* between some of their comparable elements.

[3] https://en.wikipedia.org/wiki/Pythagorean_theorem

Four colors theorem [Appel]: The vertices of every planar graph can be colored with at most four colors so that no two adjacent vertices receive the same color. The theorem was proved in 1976 by Kenneth Appel and Wolfgang Haken. Adopting a divide-and-conquer approach, they first demonstrated that it would suffice to verify the property over what they called "reducible configurations". So their strategy consisted of ***detailing*** the original general problem, by decomposing it into cases, to be analyzed one by one. Having identified 1936 of those configurations (later reduced to 1476), they had to resort to computer support to perform the checking task. A subspecies of this strategy is the well-known finite induction method, based on the axiom scheme $\varphi(0) \land \forall x[\varphi(x) \rightarrow \varphi(x')] \rightarrow \forall x\varphi(x)$, where x' stands for the successor of natural number x.

Euclid's theorem [Euclid]: There are infinitely many prime numbers. A surprisingly straightforward way to prove this is to assume the contrary and show that this leads to a contradiction. Suppose, then, that there exists a finite set of primes. Let $P = (p_1, p_2, \ldots, p_n)$ be this set, containing all existing primes, and consider a number q formed by the product of these primes plus 1. If q (which is by construction larger than any $p_i \in P$) is prime then we already have a contradiction to the assumption. If q is not prime, then it must be divisible by at least one prime p_j, which cannot be any of the primes in P, which would yield 1 as remainder – again contradicting the assumption. This strategy, based on ***negation***, is known as proof by contradiction (also called reduction to absurdity).

It should be mentioned that proof by contradiction is only admissible if one accepts the *principle of excluded middle* – which prescribes that, for any proposition P, either that proposition or its negation ~P must be true, there being no third possibility (*tertium non datur*). The principle is accepted in classical logic systems, being rejected however in *intuitionistic logic*.[4]

To support our layman's above suggestion that theorem-proving starts as an abductive process, it is appropriate to bring in the opinion of a highly reputed mathematician [Polya]:

> Finished mathematics presented in a finished form appears as purely demonstrative, consisting of proofs only. Yet mathematics in the making resembles any other human knowledge in the making. You have to guess a mathematical theorem before you prove it; you have to guess the idea of the proof before you carry out the details.

3.2. Applying analogy outside the domain of mathematics

The four strategies are not confined to the proof of mathematical or logical theorems. From this point on, we shall argue for their applicability to other domains. Consider for example a riddle (see figure 2), proposed by Fauconnier and Turner, that seems to call for complex calculations but turns out to necessitate none of that[5]:

> A Buddhist monk begins at dawn one day walking up a mountain, reaches the top at sunset, meditates at the top overnight until, at dawn, he begins to walk back to the foot of the mountain, which he reaches at sunset. Make no assumptions about his starting or stopping or about his pace during the trips. Riddle: is there a place on the path that the monk occupies at the same hour of the day on the two trips?

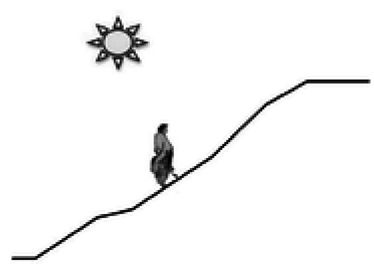


Figure 2: the Buddhist Monk puzzle

[4] https://plato.stanford.edu/entries/logic-intuitionistic/
[5] http://markturner.org/blending.html

Note that we are not asked to find a formula to compute a precise place and hour. It suffices to prove: “that there is indeed such a place, occupied at exactly the same time going up and going down”. And that can be done through the analogical problem solving strategy. It is easy to understand that the following is an analogous problem whose obvious solution serves to answer the original question, as shown by those authors, who invite us “to imagine the Buddhist monk walking both up and down the path on the same day. Then there must be a place where he meets himself, and that place is clearly the one he would occupy at the same time of day on the two separate journeys”.

The analogical problem solving strategy is the key concept in a most productive computer science approach, called *Case Based Learning* (**CBR**) [Kolodner], already with useful results in an ample variety of domains. Once a new problem is given, a **CBR** system performs a pattern-matching task to retrieve an analogous case from a case library, and then attempts to adapt its solution so as to reuse it in the context of the new problem’s specific characteristics. Successfully solved new problems are retained in order to enrich the case library.

The adaptation task is far more complex than the pattern-matching retrieval task. Heuristic methods may be required, as well as user interaction. According to Fauconnier and Turner, adaptation often utilizes a *blending* operation with inputs coming from the two problem cases, from which an adapted solution is expected as output. The essence of the operation (cf. [Fauconnier]) is “to construct a partial match between two inputs, to project selectively from those inputs into a novel 'blended' mental space, which then dynamically develops emergent structure”.

It has been noted that both deduction and abduction are relevant to **CBR**. Returning to the medical domain, consider a case library keeping information about a number of patient consultations. When a new patient arrives, a doctor, after observing the patient’s symptoms, would try to locate in the library one or more records of previous patients with similar symptoms. If the registered diagnostic is identical for all of them (or for a significant majority), the doctor might surmise that the illness affecting the new patient could well be the same – which is abductive reasoning, *from symptoms to illness*. But explaining the symptoms as the likely consequence of such illness is still part of the first stage of **CBR**, namely the characterization of the real problem: the illness to be cured. The second **CBR** stage, finding a solution to that problem, would involve retrieving and adapting to the new patient’s case the treatment thanks to which former patients recovered – and this is deduction, since it proceeds forward *from illness to treatment*.

3.3. The four strategies and database relational completeness

Having listed four strategies, we shall argue that they are in an important sense *complete*. As an introduction, we shall look at a well-accepted claim by E. F. Codd in the database area, the so-called *relational completeness* of his proposed set of algebraic operations. This set, originally containing 8 operations, later reduced to 5 basic or primitive ones – namely **product**, **projection**, **union**, **selection** and **difference** – should be enough to perform all conceivable manipulations over the columns and rows of database tables in *first normal form*. To provide a formal justification, Codd demonstrated that any expression in his relational calculus (a first-order logic theory) could be translated into a sequence of those algebraic operations.

Codd’s model was later extended to cope with *part-of* hierarchies. For this purpose, tables in non-first-normal-form (NF^2), containing structured cells, as in figure 3, were introduced. This extension, involving the notion of *granularity*, should permit to group information at different levels of detail. To preserve completeness, two new inverse basic operations were required: **nest** and **unnest**.

S_NAME	COURSE	
	C_NAME	GRADE
Jones	Math	A
	Science	B
	Physics	B
Smith	Math	A
	Physics	C
	Science	A
	Chemistry	A
	English	B

Figure 3: an NF^2 table

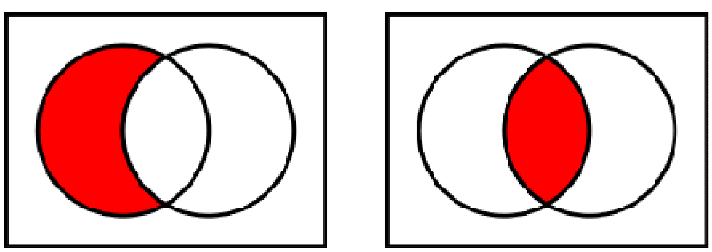
A more intuitive view of relational completeness comes from regarding the data represented by the tables as a *three-dimensional information space*, over which this set of seven basic algebraic operations can be shown to be sufficient to perform all necessary actions. Information spaces are usually bounded by *integrity constraints*, which *negate* whatever would lead to an incorrect database state. In Codd's algebra, the relational logic notion of negation is translated into the **difference** operation, employed for removal in table manipulation. A visual image can thus be brought to mind, wherein the role of the operations is indicated:

Horizontal axis – to construct: **product**; to project: **projection**
Vertical axis – to construct: **union**; to project: **selection**; to remove: **difference**
Depth axis – to zoom out: **nest**; to zoom in: **unnest**
Space bounds – to preserve integrity: **difference**

A relation table properly derived from an entity-relationship conceptual schema corresponds to either an entity class or a relationship class, and its rows represent the current instances of the class. The columns, in turn, serve to establish separate positions for the various attributes of the class. The orthogonal structure of the tables forces the instances of the class to be minimally similar, in that all instances must share the same properties (possibly with different attribute values, a *null* value being sometimes permitted). Finally, in NF^2 tables, the row cells are allowed to be subdivided hierarchically. Thus, three rhetorical strategies can be associated with moves along each of the three axes: **chaining** by a column-wise *horizontal* move to align the attribute values and relationship connections of individual instances of the class; **analogy** by a row-wise *vertical* move alternating among the homogeneous instances of the class; **detailing** by moving in *depth* to vary the view perspective, going downward to the elementary components or, inversely, going upward by grouping. The fourth rhetorical strategy, **negation**, actuates in the *space bound* constraints.

3.4. Some remarks on the use of negation

The completeness of the above mentioned set of seven operations implies that any other operation that may look useful in practice cannot be classified as basic, i.e. one should be able to define it in terms of the basic operations, which are only those belonging to the set. To this effect, **difference**, the operation that brings in the notion of *negation*, plays a major role. An obvious example is the **intersection** operation, which can be defined through a double negation (a notion disallowed in intuitionistic logic) achieved by applying difference two times. To see this, recall that the intersection of two sets A and B can be expressed as: $A \cap B = A - (A - B)$. Figure 4 uses Venn diagrams to illustrate this two-stage derivation, the circles standing, from left to right, for sets A and B respectively, with the red color highlighting the indicated results.



A - B A - (A - B)
Figure 4: intersection achieved by twice applying difference

Double negation also serves to translate universal into existential quantification. To say that everything that has property p must also have property q is equivalent to saying that nothing exists with property p that does not have property q, or more formally: $\forall x\ (p(x) \rightarrow q(x)) \leftrightarrow \neg\exists x\ (p(x) \wedge \neg q(x))$. Codd had originally included a rather intricate operation, that he called **division**, to cover expressions involving universal quantification, as happens with the popular example of "suppliers who supply all parts". But it was soon realized that this could be expressed by means of a chain of successive basic operations, including two applications of difference, and therefore there was no need to keep division in the base set.

In database theory, the question of how to handle negation has been treated in a practical way. A fundamental principle, known as the *closed world assumption*, establishes that a fact can be said to hold in the mini-world represented by a given database if and only if it is explicitly recorded in the database files. In the context of *deductive databases*, which is supplemented by inference rules, the set of facts is extended to include those that can be derived through the application of the rules. Whatever cannot be proven, i.e. facts that are neither recorded nor can be derived, is assumed to be false, complying to the principle of *negation as failure*, prevailing in logic programming systems. This means that negative facts need not be represented, since ~p is said to hold whenever p fails to be demonstrated.

According to the Entity-Relationship conceptual model, facts may assert the existence of instances of the specified classes of entities, as well as the value of their attributes and the presence of relationships connecting them. When comparing a pair of opposite values v1 and v2, negation can function in two different ways: either v1 necessarily implies ~v2, or else there are intermediate situations wherein both ~v1 and ~v2 are simultaneously admitted. An example of absolutely incompatible values is provided by v1 = 'white' and v2 = 'not white', whereas between the two extreme values v1 = 'white' and v2 = 'black' one can distinguish a 'gray' intermediate zone lying between two arbitrarily imposed thresholds. Figure 5 illustrates these two types of value opposition, put in contrast by the well-known *negation square* diagram shown in figure 6.

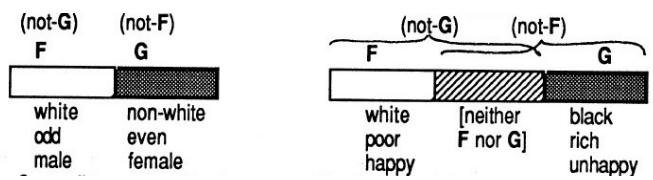


Figure 5: two types of opposed values

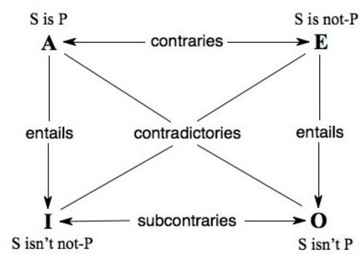


Figure 6: the negation square

3.5. On the diverse meanings of part-of

A seminal work about part-of hierarchies [Winston] has shown that the concept encompasses at least six different types. The authors use the term *meronymic relations* – which we have adopted from them – to cover all these types, claiming that they "differ in three main ways: "whether the relation of part to the whole is *functional* or not, whether the parts are *homeomerous* or not, and whether the part and whole are *separable* or not". 'Functional' parts serve restricted, separate purposes, as for instance the handle of a cup is the part to be used to hold the cup. 'Homeomerous' means to be the indistinguishable from the whole, as one can say that a slice of a pie is a pie. Finally, 'separable' parts are those that one can without undue difficulty take away from the whole, as happens with the cards from a deck, but not with a dose of gin after being mixed with other ingredients to produce martini. The six types are exemplified below, followed by **+** or **-** signs to indicate the presence or absence of each of the three characteristics just described:

Component / Integral Object – handle / cup	**+ - +**
Member / Collection – card / deck	**- - +**
Portion / Mass – slice / pie	**- + +**
Stuff / Object – gin / martini	**- - -**
Feature / Activity – paying / shopping	**+ - -**
Place / Area – Everglades / Florida	**- + -**

The authors show by means of examples that the part-of relation may fail to be transitive in arguments involving more than a single type. In one of these misbegotten examples an assertion of the first type listed above is unduly combined with an assertion of the second type, leading to a patently absurd conclusion:

Simpson's arm is part of Simpson.
Simpson is part of the Philosophy Department.
* Simpson's arm is part of the Philosophy Department.

4. From reasoning strategies to semiotic relations

When specifying any system, and when using it as well, some guidelines should be available. What properties are relevant to characterize an entity? What events should be observed? How do agents interact, either collaborating or competing? Is it possible to attain modularity, by setting the focus to different degrees of detail? Which integrity constraints should be enforced?

We based our proposal on studies [Ramus, Vico, Burke, Chandler] asserting the completeness as reasoning processes of the so-called *four master tropes* – metonymy, metaphor, irony and synecdoche. Their universality has been repeatedly emphasized, with the indication that they may constitute "a system, indeed *the* system, by which the mind comes to grasp the world conceptually in language" [Culler]. Reasoning about these tropes, we identified four types of semiotic relations that can exist not only between facts, but also between events and between agents, which we denominated, respectively, *syntagmatic*, *paradigmatic*, *antithetic* and *meronymic* relations. Informally speaking, syntagmatic relations refer to connectivity, paradigmatic relations to similarity and analogy, antithetic relations to negation, and meronymic relations to hierarchy. The intended meaning of the semiotic relations can be summarized as follows:

- **Syntagmatic** – chaining, connectivity - *and*
- **Paradigmatic** – analogy, similarity - *or*
- **Meronymic** – detailing, hierarchic - *part-of*
- **Antithetic** – negation, opposition – *not*

The term syntagmatic has been used by linguists [Saussure] to characterize a *horizontal axis* in the formation of sentences, whereas the term paradigmatic [Jakobson] introduces a *vertical axis*, along which are disposed, in a line perpendicular to the horizontal line upon which the sentence is represented, one or more words that can replace as acceptable alternatives the word at the intersection of the two lines. The term antithetic serves well to characterize the rules of grammar that establish whether a given sequence of words constitutes a well-formed sentence, thus imposing bounds to the space of language (as integrity constraints do to the information space). Linguists also use antithesis to help defining a concept "… not positively, in terms of their content, but negatively by *contrast* with other items in the same system. What characterizes each most exactly is being whatever the others are not" [Saussure, Chandler]. The term meronymic, which we borrowed from the already mentioned seminal article where six types of part-of were distinguished [Winston], would correspond to a *depth axis*, allowing to, so to speak, zoom in and out across hierarchical levels (in the language domain: sentence, word, syllable, etc.). Figure 7 is a standard illustration[6] of the syntagmatic and paradigmatic axes, as viewed by linguists.

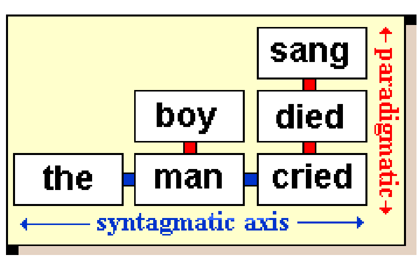
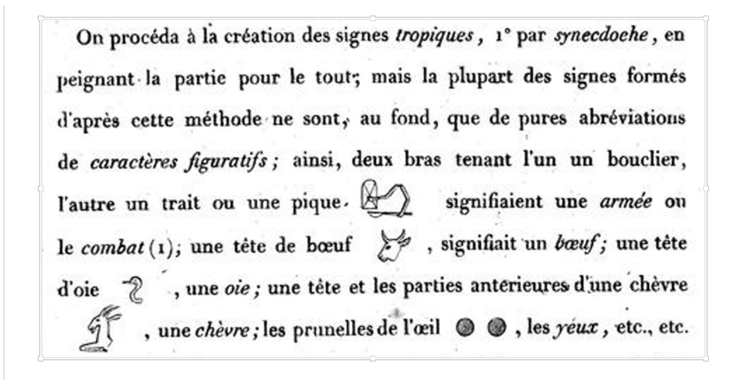


Figure 7: syntagmatic and paradigmatic axes

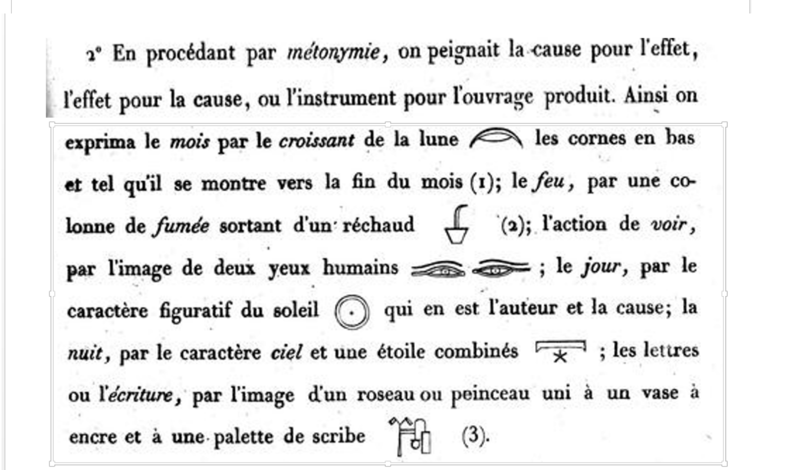
[6] http://visual-memory.co.uk/daniel/Documents/S4B/sem03.html

Still in the language domain, a striking testimony of the ubiquitous influence of the master tropes is provided by the French philologist Jean-François Champollion. In a posthumous book [Champollion], he affirmed that Egyptian hieroglyphs should be interpreted on the basis of three among the four master tropes, namely synecdoche, metonymy and metaphor, adding a catchall term, 'enigma', to cover whatever escaped these three cases. So, the head of an animal, such as the ox, would stand for the animal (synecdoche); the image of two human eyes for the act of seeing (metonymy); a falcon, in view of its high flight, for the abstract concept of sublimity (metaphor).

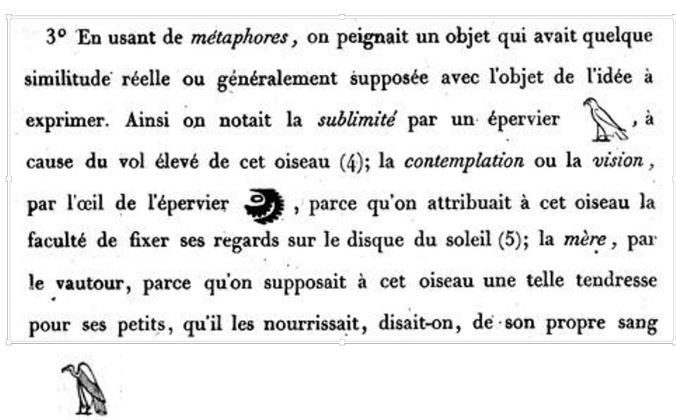
Proceeding by enigma meant, for Champollion, to employ the image of an object with distant, vague, to some extent occult relations with the intended concept. Curiously the very first example that he gives – an ostrich feather signifying 'justice' – involves negation, thus encompassing irony, the missing fourth master trope. He explains that an ostrich feather stood for 'justice' because, "as people used to say, all feathers of the wings of those birds are equal". Note that the notion of justice thus conveyed can be expressed by universal quantification (which, as we recalled earlier, is equivalent to double negation) or even by direct negation: *all* claimants are equally treated, *no* difference is permitted. But it must be added that the word 'enigma' has also the merit to recall the margin of doubt remaining in any model of reality. Four passages from [Champollion] are reproduced in figure 8, to illustrate his fourfold rhetorical analysis on the genealogy of hieroglyphs.



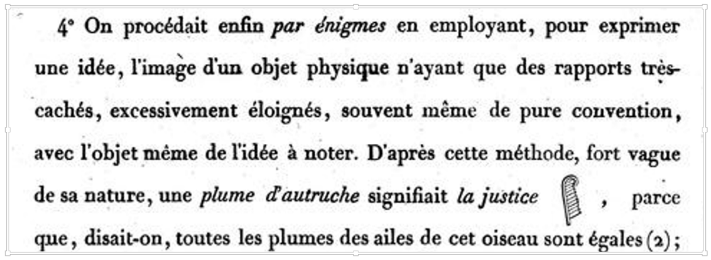
On procéda à la création des signes *tropiques*, 1° par *synecdoche*, en peignant la partie pour le tout; mais la plupart des signes formés d'après cette méthode ne sont, au fond, que de pures abréviations de *caractères figuratifs*; ainsi, deux bras tenant l'un un bouclier, l'autre un trait ou une pique. signifiaient une *armée* ou le *combat* (1); une tête de bœuf , signifiait un *bœuf*; une tête d'oie , une *oie*; une tête et les parties antérieures d'une chèvre , une *chèvre*; les prunelles de l'œil , les *yeux*, etc., etc.



2° En procédant par *métonymie*, on peignait la cause pour l'effet, l'effet pour la cause, ou l'instrument pour l'ouvrage produit. Ainsi on exprima le *mois* par le *croissant* de la lune les cornes en bas et tel qu'il se montre vers la fin du mois (1); le *feu*, par une colonne de *fumée* sortant d'un réchaud (2); l'action de *voir*, par l'image de deux yeux humains ; le *jour*, par le caractère figuratif du soleil qui en est l'auteur et la cause; la *nuit*, par le caractère *ciel* et une étoile combinés ; les lettres ou l'*écriture*, par l'image d'un roseau ou peinceau uni à un vase à encre et à une palette de scribe (3).



3° En usant de *métaphores*, on peignait un objet qui avait quelque similitude réelle ou généralement supposée avec l'objet de l'idée à exprimer. Ainsi on notait la *sublimité* par un épervier , à cause du vol élevé de cet oiseau (4); la *contemplation* ou la *vision*, par l'œil de l'épervier , parce qu'on attribuait à cet oiseau la faculté de fixer ses regards sur le disque du soleil (5); la *mère*, par le vautour, parce qu'on supposait à cet oiseau une telle tendresse pour ses petits, qu'il les nourrissait, disait-on, de son propre sang



4° On procédait enfin *par énigmes* en employant, pour exprimer une idée, l'image d'un objet physique n'ayant que des rapports très-cachés, excessivement éloignés, souvent même de pure convention, avec l'objet même de l'idée à noter. D'après cette méthode, fort vague de sa nature, une *plume d'autruche* signifiait *la justice* , parce que, disait-on, toutes les plumes des ailes de cet oiseau sont égales (2);

Figure 8: Champollion's statements concerning the trope-oriented generation of hieroglyphs

Passing from ancient Egypt to present day media, we continue to find the influence of the semiotic relations that we derived from the classic four master tropes. Even antithetic viewpoints are expressed, e.g. when a newspaper publishes "another opinion" side by side with an editorial that manifests the official position of the owners. Recommender systems provide another case in point. If you search for a book in Amazon, for instance, that book will be accompanied in the screen by a number of books of similar content (thus providing an example of paradigmatic relation). After that, if you select one of them, either your original choice or some other, you will be told about a number of somehow related books "frequently bought together" (supposedly because they cover a complementary topic, an example therefore of syntagmatic relations). To allow you to examine your choice in some detail (meronymic relation), it will display pointing to the cover a "look inside" invitation. And if you are still in doubt you can peruse top customer reviews, examining especially the most critical ones (antithetic relation). And for Internet search, in general, the habit of taking into consideration the four semiotic relations – never forgetting to look at what appears to go contrary to your expectations – is decidedly most rewarding.

5. Reasoning about events: from data bases to story bases

We have been working with the conceptual modeling of information systems with a *database component*, considering their static, dynamic and behavioral aspects. These three aspects, captured respectively in *static*, *dynamic* and *behavioral* conceptual schemas, were integrated through the application of a *plan-recognition / plan-generation paradigm* [Furtado 2000]. The *static schema* declares what are the classes of *facts* whose instances can hold at some database state, conveniently described in terms of the entity-relationship model. The *dynamic schema* employs a fixed repertoire of operations, defined in a STRIPS-like [Fikes] declarative style in terms of their pre-conditions and post-conditions (effects), to characterize the *events* whose occurrence is the only way to promote state transitions. The *behavioral schema* refers to the *agents* authorized to cause events by performing the operations.

Classical first-order logic concerns facts, but some temporal logics have been proposed to deal with events, notably *situation calculus* [McCarthy] and *event calculus* [Kowalski]. In our approach, we start with the notion of *situation*, which we define as a logic expression involving facts. An event, then, can be simply regarded, initially, as a *transition* from a state in which a situation S_c is true into a state in which some other situation S_g is true. In an information system, however, there may exist *transition constraints*, which must be checked in order to determine whether or not an intended event (i.e. intended transition) is to be considered *valid*. In addition, most systems must also abide by *static constraints* that characterize what states are valid. One orientation – which we have adopted – to enforce all sorts of constraints is to impose a strict *abstract data type* discipline, whereby transitions are only allowed to happen through the execution of a fixed repertoire of *event-producing operations*, whose pre-conditions and post-conditions have been mutually adjusted in such a way that validity is guaranteed.

For this purpose, we extend the notion of event by adding to the representation of the intended transition an event-producing operation O; situation S_c now expresses the pre-conditions of O and S_g its post-conditions. To realize what is implied by this operational notion of event, consider the over-simplified formula:

$S_c, O \rightarrow S_g$

which is to be read as follows: if in the current state S_c holds and O is executed, one can deduce that S_g will hold in an immediately following target state. Writing "deduce", in correspondence with the arrow, we mean to suggest, now talking about events, a comparison with the syllogism – the elementary device for reasoning about facts, mentioned at the beginning of this paper. In turn, crossing the arrow in the opposite direction offers a form of abduction to hypothesize what might lead to a state wherein S_g holds. Also, similarly to what we pointed out when discussing problem-solving strategies, it may be necessary to resort to *chaining,* here in the sense of applying a partially-ordered series P of two or more event-producing operations in order to, after a number of transitions through successive states, finally reach a state wherein Sg is true. Furthermore, if situation Sg is understood as a *goal* of some agent, then P should be interpreted as a *plan*.

The introduction of plans opens the way to four complementary tasks:

(1) generation, for which we have implemented backward-chaining (hence, abductive) algorithms;
(2) simulation, achieved by executing plans in workspace memory so as to *predict* possible future situations;
(3) discovery, by extracting from a **Log** (implemented as a relational table), and then filtering, sequences of records of executed event-producing operations that agents have frequently used in practice as alternative *typical plans* to reach their goals;
(4) recognition, which involves an attempt to *explain* the behavior of agents by finding out what plans, aiming at what goals, they are trying to pursue at a given instant; this is done by observing which event-producing operations they have executed until then, and matching these observations against a library of typical plans.

It has been remarked very early (cf. [Shanahan], for instance) that prediction, as in (2), is a form of deduction, whereas explanation, as in (4), corresponds to abduction. Discovery, as in (3), can obviously be categorized as induction.

Having adopted this plan-based paradigm, we were in a position to shift our attention from the repository of facts, which constitutes the database component of an information system, to the events whose occurrence during its lifetime could – if registered in some format as sequential plots – would enable us to view the system in terms of the *stories* emerging from its formal specification. In other words, we purported to have available two mutually dependent components to more fully exploit the potential of the specified information system, namely the *data base* itself and a *story base*.

To enable practical experiments in this direction, we developed a prototype, called IDB, described in detail in a technical report [Gottin 2015], which made possible to test our conceptually specified event-producing operations along three successive stages. The first stage deals with the logic programming (Prolog) definition of the three conceptual schemas. The operations, defined as already mentioned in a declarative style by their pre-conditions and post-conditions (i.e. effects, specified in terms of the facts to be added and the facts to be removed), can be tested by simulated execution in workspace memory, and can be chained together into plans by the implemented algorithms. At the second stage, at which Prolog communicates with Oracle via an ODBC interface, relational tables are created in correspondence to the entity-relationship static schema, and the declarative-style operations are compiled into a semi-procedural format, where predicates implementing `select` commands are generated to check the pre-conditions, and `insert`, `delete` and `update` commands to produce the effects concretely on the database tables. Generated plans can likewise be effectively executed, rather than merely simulated, being treated as database *transactions*, which are caused to backtrack if the pre-conditions of a constituent operation happen to fail. At the third stage, a second compiler converts the semi-procedural operations into independently executable Oracle storage procedures.

As noted, keeping the orientation adopted in our early work, our approach consistently relies on a strict abstract datatype discipline to be maintained throughout the three stages, by requiring that database manipulation be restricted to such sets of pre-defined operations, whose pre-conditions and effects are articulated so as to enforce all integrity constraints. Hopefully, possible inadvertent mistakes in a specification could be detected by simulated execution or anticipated by plan-generation. Apart from error-detection, planning should also help the designers to check whether each of the legitimate goals of the prospective user agents specified at the behavioral schema could be met, and, in contrast, whether there might exist unforeseen ways to reach inconsistent or undesirable situations.

But actual practice may still reveal previously ignored aspects. From an analysis of typical plans developed by resourceful users, the designers should be able to devise ways to introduce corrections and improvements to the original specification. For this purpose, the IDB prototype features a powerful facility to discover typical plans. Whenever any of the event-producing operations is executed upon the database tables, the execution has the side effect of inserting into the **Log** a record indicating the respective transaction number, the current time stamp, and the name and parameters of the event-producing operation. The discovery algorithm works on this special IDB environment that comprises the database tables, which represent the currently existing entity and relationship instances, and the time-stamped **Log** records, which register the execution of operations whose pre-conditions and effects definition is available to the algorithm. Given as input a goal that is true at a given state, it proceeds in a backward direction until reaching a goal-motivating situation, and yields as output one or more *traces* extracted from the **Log**, i.e. sub-sequences of the **Log** filtered to exclude operations not belonging to the same transaction or in no way contributing to the goal. Frequently occurring traces are suitable candidates for inclusion in a *library of typical plans* [Furtado 2001]. A plan-recognition algorithm (cf. task 4, above) was also implemented, as an extended version of the method described in [Kautz], to work on libraries of typical plans.

Users are also allowed to insert records, with the same composition as those of the **Log**, into an **Agenda** table, in order to register operations scheduled for execution at some future time. The presence of these two special tables, which register the past and the (possibly still non-committed) future occurrence of events, together with the plan-generation and plan-recognition algorithms, sets up a *temporal database* environment wherein, if periodical snapshots are taken from the database tables, it should become practically viable – at a reasonable cost – to recover the database state at any given time T. This would be achieved by locating and copying the snapshot which, by the time T_i of its creation, happens to be the closest one preceding T, and updating the copy through the execution of the event-producing operations registered in the **Log** between T_i and T.

6. Network organized reasoning

With the help of the IDB environment, we have been working on a *process-mining* project, which started with a preliminary investigation over the academic domain of our university [Gottin 2017]. The project extends our early work on what we called *plot-mining* [Furtado 2007]. A further extension is the combination of traces that aim at the same given goal into a *network structure*, wherein sub-sequences representing similar events are coalesced, and the convergence or divergence of sub-sequences is pictorially made explicit by join or fork nodes. By what we termed *network organized reasoning*, one can easily perceive what the traces have in common and in what they differ, and can also devise new plans by traversing the network along some path composed of parts taken from different traces. It should be noted that an important related research project on process mining [Aalst] also utilizes logs and network / workflow representations, but, contrary to our approach, is fundamentally directed to legacy databases, not assuming therefore the availability of formally defined conceptual database schemas.

Consider, for example, an academic information system wherein a search is conducted over the **Log** to find out how students who started with zero credits managed to complete 3 credits. The course offerings in our trial consisted of Design (1 cr.), Art (2 crs.), Semiotics (3 crs.); however an event-producing operation was provided whereby a course might receive the benefit of a one-credit increment. It so happened that seven traces with a successful outcome were extracted. As a preliminary step required by our algorithm, generalization was applied to the traces, limited to the strictly necessary to allow the detection of similarities; in this case it was enough to substitute the word 'Student' for the different student names in the seven traces. When mounting the network, the node labels assigned by the algorithm were added as prefixes to the event-producing terms, thus enabling to easily follow the paths of the seven traces over the network (cf. figure 9):

*** Paths:

(1) [N1:ini,N2:enroll(Student,Semiotics),N3:drop(Student,Semiotics),N4:enroll(Student,Art),N5:change_cr_post(Art),N6:mark(Student,Art),N7:end]

(2) [N1:ini,N2:enroll(Student,Semiotics),N8:change_cr(Art),N9:transfer(Student,Art),N6:mark(Student,Art),N7:end]

(3) [N1:ini,N2:enroll(Student,Semiotics),N3:drop(Student,Semiotics),N10:enroll(Student,Design),N8:change_cr(Art),N9:transfer(Student,Art),N6:mark(Student,Art),N7:end]

(4) [N1:ini,N2:enroll(Student,Semiotics),N3:drop(Student,Semiotics),N10:enroll(Student,Design),N11:drop(Student,Design),N4:enroll(Student,Art),N5:change_cr_post(Art),N6:mark(Student,Art),N7:end]

(5) [N1:ini,N4:enroll(Student,Art),N5:change_cr_post(Art),N6:mark(Student,Art),N7:end]

(6) [N1:ini,N2:enroll(Student,Semiotics),N12:mark(Student,Semiotics),N7:end]

(7) [N1:ini,N10:enroll(Student,Design),N4:enroll(Student,Art),N6:mark(Student,Art),N13:mark(Student,Design),N7:end]

```
*** Events leading to nodes:
    N1:[ini]
    N2:[enroll(Student,Semiotics)]
    N3:[drop(Student,Semiotics)]
    N4:[enroll(Student,Art)]
    N5:[change_cr_post(Art)]
    N6:[mark(Student,Art)]
    N7:[end]
    N8:[change_cr(Art)]
    N9:[transfer(Student,Art)]
    N10:[enroll(Student,Design)]
    N11:[drop(Student,Design)]
    N12:[mark(Student,Semiotics)]
    N13:[mark(Student,Design)]
```

Once the network is obtained, users can interactively compose stories congenial to their own tastes, by traversing new paths that incorporate sub-sequences of different traces. The dialogue reproduced below, illustrates how a user was asked at each decision point to choose from the current alternatives, yielding at the end a story (represented by the green-colored path in figure 9) that is indeed new, since it incorporate nodes N1, N10, N4, as in trace (7), followed by N5, N6, N7, as in trace (5). The name 'Arthur' was supplied by the user.

To our surprise, when we personally tried this little experiment, intending to end up with 4 credits and thus surpassing the specified 3 credits goal, we were (deservedly…) frustrated in our attempt. We expected that, after the choice of N5, the dialogue would pause at the fork node N6, where we would be allowed to pick N13 from the options list [N13,N7] – but the "system" knew better. It had been told not to bother users with unnecessary choices, and after our choice of N5 it concluded that the 3 credits goal was guaranteed by just proceeding non-stop through N5-N6-N7.

```
3 ?- new_story.

>> Choose from: [N10,N2,N4] - N10
>> Choose from: [N11,N4,N8] - N4
>> Choose from: [N5,N6] - N5

*** Path:
[N1,N10,N4,N5,N6,N7]

*** where:
N1:ini
N10:enroll(Arthur,Design)
N4:enroll(Arthur,Art)
N5:change_cr_post(Art)
N6:mark(Arthur,Art)
N7:end
```

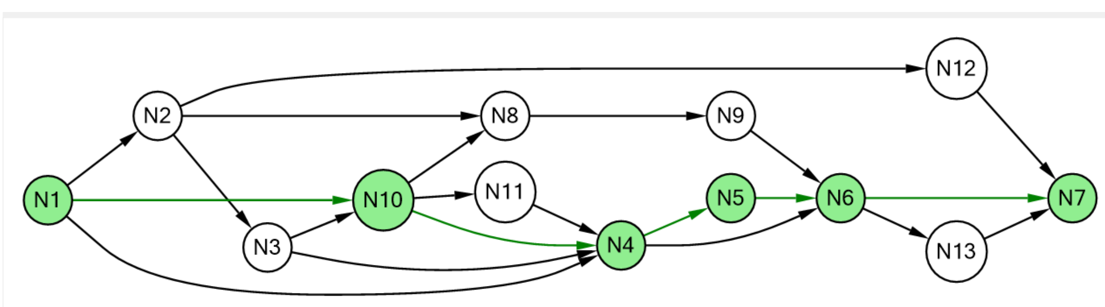


Figure 9: network for simple academic example

Since traces are plots of stories, we feel justified to regard and treat as a story base any data repository associated with records of events pertaining to any kind of domain, real or imaginary. Indeed, we have, in our long-term digital entertainment ***Logtell*** project [Ciarlini], extended our research to literary stories, applying computational narratology notions. In a recent paper [Lima] we demonstrated how to construct networks by combining different variants of a folktale into a network, and described a prototype to help users with no authorial background to interactively compose new variants according to their preferences. Figure 10 displays the network obtained from the condensation of four variants of the popular Little Red Riding Hood story. The green-colored path signals the successive choices that led to one user's own Little Red Riding Hood story, narrated in template-driven natural language on the upper left-hand side, and as comics strip images on the upper right-hand side. By entering a show_tell command, the user was then entitled to sit back and watch a frame by frame voice narrative.

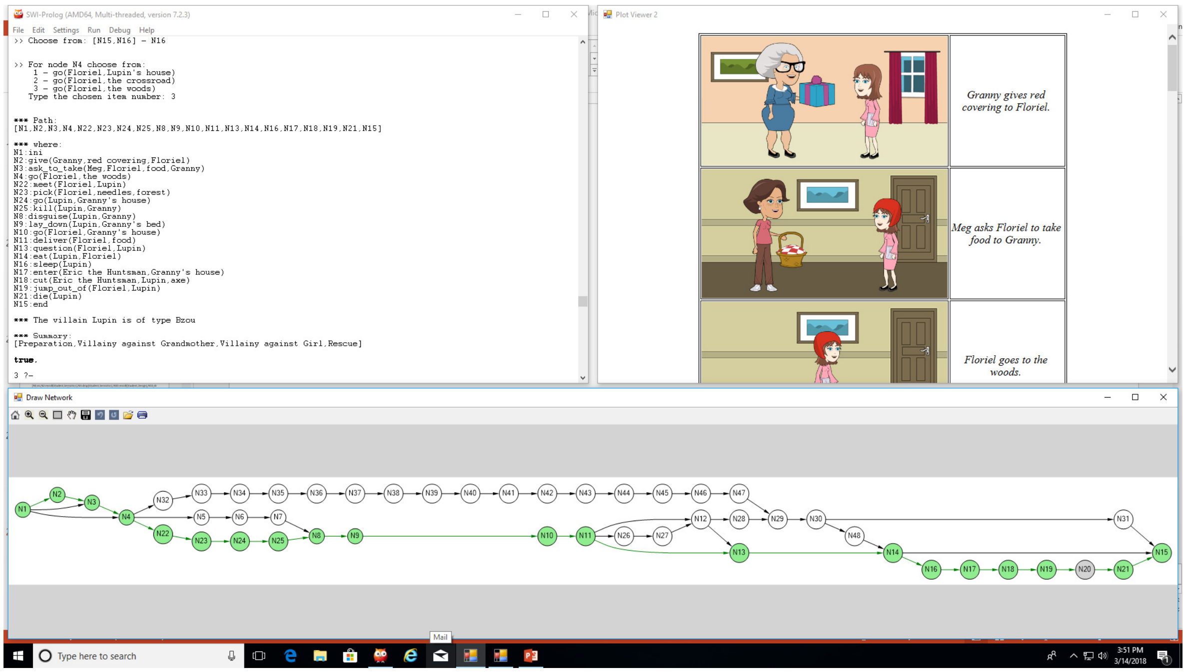


Figure 10: user-created Little Red Riding Hood story composed by network traversal

7. Concluding remarks

Writing this report, our objective was to offer a broad overview of argument as a resource essential to any sort of research work, expecting that the text would serve as a guideline prepared specifically for computer science projects. We tried to convey the intuitive notions involved, rather than the precise technical details of the various formalisms, which the interested reader is asked to look for in the specialized literature. In particular, we refer to [Ciarlini] for a formal description of our conceptual modeling approach.

As we proceeded from the *syllogism* to strategies and methods suitable for information systems applications, we stressed a number of aspects that seemed to deserve special attention. Even the use of that basic scheme, involving a single rule of the form Fact1 implies Fact2, may not be so simple as it looks: the rule must have a rational justification beyond the mere observation that Fact1 is often followed by Fact2; on the other hand, concluding Fact2 from the occurrence of Fact1 may be discredited as a *false inference*, but may instead, under the more dignified name of *abduction*, provide a plausible hypothesis and even lead to some major breakthrough.

The gist of the report was our characterization of four *semiotic relations* that we believe to be sufficient to determine the bounded three-dimensional structure of an information space. These syntagmatic, paradigmatic, meronymic and antithetic relations, emerging from the long tradition of the rhetorical *four master tropes*, have been claimed to provide a (for practical purposes) *complete* coverage in such widely distinct domains as mathematics and natural language. To add just one more domain as likely evidence of universality, the American historian Hayden White [Burke, White] indicated the master tropes as "part of the 'deep structure' underlying different historiographical styles" (cf. [Chandler], p. 137). Thinking of each of the four semiotic relations we are reminded, when searching for a piece of factual information F (no matter if over a conventional database or across the Internet), that it may also be worthwhile to search, respectively, for what is *connected*, or *similar*, or *hierarchically related*, or *contrary* to F. Recognizing and discussing contrary opinions always was, incidentally, a requirement of honest academic debate – as demonstrated by St. Thomas Aquinas, the greatest continuator of Aristotle, in his *Summa Theologica*[7], where, for every question examined, all objections that might be raised were duly cited and thoroughly discussed.

With the ample opportunities offered by today's open universe of linked data, expanding a search may lead to excessive *recall*, in detriment of *precision*. However the relatively new interest on *serendipitous* findings [Eichler], i.e. the accidental discovery of something of great interest that one is not currently looking for, is one more temptation to relax the precision requirement. We would suggest, as a compromise to avoid extra information attractive only as curiosity, that searches be directed not only at the current goal involved in the search, but also at the user's somehow declared or observed *latent* goals.

To show how to achieve running conceptual specifications following a *plan recognition / plan generation paradigm*, we briefly introduced our prototype tool which, passing from descriptive *facts* to narrative *events*, adds a *story base* component to information systems. This extension brought, among other consequences, the ability of the data repository of the system to start operating as a temporal database. The definition of event-producing operations in terms of pre-conditions and post-conditions was compared to syllogisms as a basic reasoning scheme applicable to causally related events. The interplay of the pre-conditions and post-conditions of the diverse operations provided, in turn, the basis to implement backward chaining plan-generation algorithms.

Besides automatically generated plans, useful for verifying by simulation the specified system's ability to achieve the goals of the prospective agents, as well as frustrate erroneous or fraudulent attempts, we stressed the importance of extracting from a **Log**, whereupon the execution of the event-producing operations has been recorded, the *typical plans* effectively employed by the user agents. This systematic extraction and analysis activity became for us a vital *process mining* task.

As the most elaborate instrument for comparing alternative plans, either automatically generated or extracted from a **Log**, we recommended *network organized reasoning* schemes, where coinciding, diverging and converging sequences are conveniently displayed, and over which new plans can be devised by combining parts of different alternatives. Each of these forms of argument, we finally remarked, has proved equally useful in digital entertainment applications.

Future work is of course needed to more fully correlate all these notions. However, as a sobering last thought, a disclaimer is in order: there is not and there can never be such thing as a complete argument model, since models are, by definition, an incomplete representation of reality. Thus, as a closing remark, we must admit that the notion of *completeness*, as asserted by Codd with respect to his algebraic formalism, and as attributed here to our semiotic relations, is necessarily relative and limited to the practical uses intended.

Acknowledgement. The author is grateful to his colleague Simone D. J. Barbosa for calling his attention to this topic and for numerous insightful conversations.

References

[Aalst] W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, 2011.

[Appel] K. Appel, W. Haken. Every planar map is four-colorable. *Bulletin of the American Mathematical Society*, 82, 5, 1976.

[Aristotle] Aristotle. *Poetics*. In *Classical Literary Criticism*. P. Murray and T.S. Dorsch (trans.). Penguin, 1965.

[Burke] K. Burke, K. *A Grammar of Motives*. University of California Press, 1969.

[Champollion] F.A. Champollion. *Grammaire égyptienne : ou Principes généraux de l'écriture sacrée égyptienne appliquée à la représentation de la langue parlée*. Firmin-Didot Frères, 1836.

[Chang] C-L. Chang, C-T. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.

[7] http://www.documentacatholicaomnia.eu/03d/1225-1274,_Thomas_Aquinas,_Summa_Theologiae_%5B1%5D,_EN.pdf

[Chandler] D. Chandler. *Semiotics: the Basics*. Routledge, 2002.

[Ciarlini] A.E,M. Ciarlini, M.A. Casanova, A.L. Furtado, P.A.S. Veloso. Modeling interactive storytelling genres as application domains. *Journal of Intelligent Information Systems*, 35, 3, 2010.

[Culler] J. Culler. *The Pursuit of Signs: Semiotics, Literature, Deconstruction*. Routledge, 1981.

[Eichler] J.S.A. Eichler, M.A. Casanova, A.L. Furtado, L. Ruback, L.A.P.P. Leme, G.R. Lopes, B.P. Nunes, A. Raffaetà, C. Renso. Searching linked data with a taste of serendipity. In *Proc. of the 29th International Conference on Advanced Information Systems Engineering (CAiSE)*, 2017.

[Euclid] Euclid. *The Thirteen Books of the Elements*. T.L. Heath (trans.). Dover Publications, 1956.

[Faltings] G. Faltings. The Proof of Fermat's Last Theorem by R. Taylor and A.Wiles. *Notices of the American Mathematical Society*, 42, 7, 1995.

[Fauconnier] G. Fauconnier, M. Turner. *The Way we Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books, 2002.

[Fikes] R.E. Fikes, N.J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* , 2, 1971.

[Furtado] A.L. Furtado, A.E.M. Ciarlini. The plan recognition/plan generation paradigm. In *Information Systems Engineering*. S. Brinkkemper, E. Lindencrona, A. Solvberg (eds.). Springer, 2000.

[Furtado] A.L. Furtado, A.E.M. Ciarlini. Constructing libraries of typical plans. In *Proc. of the 13th International Conference on Advanced Information Systems Engineering* (*CAiSE*), 2001,

[Furtado] A.L. Furtado, M.A. Casanova, S.D.J. Barbosa, K.K. Breitman. Plot mining as an aid to characterization and planning. *Technical Report MCC 07*. PUC-Rio, 2007.

[Gottin] V.M. Gottin, M.A. Casanova, E.S. Lima, A.L. Furtado. A story based approach to information systems. *Technical report MCC05*. PUC-Rio, 2015.

[Gottin] V.M. Gottin, H. Jiménez, A.C. Finamore, M.A. Casanova, A.L. Furtado, B.P. Nunes. An analysis of degree curricula through mining student records. In *Proc. of the 17h IEEE International Conference on Advanced Learning Techniques*, 2017.

[Jakobson] R. Jakobson. Closing statement: linguistics and poetics. In *Style in Language*. T.A. Sebeok (ed.). M.I.T. Press, 1960.

[Kautz] H.A. Kautz. *A formal theory of plan-recognition and its implementation*. Morgan Kaufmann, 1991.

[Kolodner] J. Kolodner, *Case-Based Reasoning*. Morgan-Kaufmann, 1993.

[Kowalski] R. Kowalski, M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4, 1, 1986.

[Lima] E.S. Lima, V. Gottin, B. Feijó, A.L. Furtado. Network traversal as an aid to plot analysis and composition. In *Proc. of the XVI Brazilian Symposium on Computer Games and Digital Entertainment,* 2017.

[McCarthy] J. McCarthy. Actions and other events in situation calculus. In *Proc. of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002)*, 2002.

[Oswald] M.E. Oswald, S. Grosjean. Confirmation bias. In *Cognitive Illusions: a Handbook on Fallacies in Thinking, Judgmement and Memory*. R.F. Pohl (ed.). Psychology Press, 2012.

[Peirce] C.S. Peirce. *The Essential Peirce*. Edited by the Peirce Edition Project. Indiana University Press, 1998.

[Polya] G. Polya. *Mathematics and Plausible Reasoning*. Martino Fine Books, 2014.

[Ramus] P. Ramus. *Arguments in Rhetoric Against Quintilian*. J.J. Murphy (ed.), C. Newlands (trans.). Southern Illinois University Press, 2010.

[Saussure] F. Saussure. *Cours de Liguistique Générale*. Payot, 1995.

[Shanahan] M. Shanahan. Prediction is deduction but explanation is abduction. In *Proc. of the 11th International Joint Conference on Artificial Intelligence* (*IJCAI*), 1989.

[Toulmin] S. Toulmin. *The Uses of Argument*. Cambridge University Press, 2003.

[Vico] G. Vico. *The New Science*. M.H. Finch (trans.). Cornell University Press, 1968.

[Wiles] A. Wiles. Modular elliptic curves and Fermat's Last Theorem. *Annals of Mathematics*, 142, 1995.

[Winston] M.E. Winston, R. Chaffin, D. Herrmann. A taxonomy of part-whole relations. *Cognitive Science*, 11, 4, 1987.

[White] H. White. *Metahistory: The Historical Imagination in Nineteenth-Century Europe*. John Hopkins University Press, 1973.