Some circumscriptional thoughts on SBL

G. Nuñez
U. Cortés

Report LSI–89–20

Abstract:

We illustrate how some of the Similitude Based Learning (**SBL**) paradigms can be reformulated using some logical formalisms as circumscription, predicate completion and the close world assumption. Our approach shows that is possible to use these logical tools in order to obtain a formal unified vision of **SBL** paradigms, and it also suggests some kind of improvements on the current implementations. We introduce a hitherto unmentioned direct link between machine learning and circumscription. We believe that this new framework and the obtained results are of practical interest.

Resumen: Se ilustra como algunos de los paradigmas de Aprendizaje Basado en Similitudes (**SBL**), pueden reformularse usando los formalismos logicos de Circumscripción, Completación de predicados y la Hipotesis del mundo cerrado. Esta aproximación muestra que es posible usar estas herramientas logicas para obtener una visión formal unificada de paradigmas de **SBL**) y sugiere la posibilidad de algunas mejoras en las implementaciones existentes. Este trabajo establece una notable relacion directa, que no habia sido notada anteriormente, entre Aprendizaje y Circumscripción. Nosotros creemos que este es un nuevo framework y que los resultados obtenidos son de interes practico.

Some circumscriptional thoughts on SBL

Gustavo Nuñez *
Ulises Cortés
Facultat d'Informàtica de Barcelona
Departament de Llenguatges i Sistemes Informàtics (it)
Pau Gargallo, 5. Barcelona. 08028. **SPAIN**
ia@fib.upc.es

Abstract

*We illustrate how some of the Similitude Based Learning (**SBL**) paradigms can be reformulated using some logical formalisms as circumscription, predicate completion and the close world assumption. Our approach shows that is possible to use these logical tools in order to obtain a formal unified vision of **SBL** paradigms, and it also suggests some kind of improvements on the current implementations. We introduce a hitherto unmentioned direct link between machine learning and circumscription. We believe that this new framework and the obtained results are of practical interest.*

1. Motivation

La beauté est un vice, merveilleux, de la forme.
César Moro

One of the defining features of intelligence is the ability to learn. Thus, machine learning is a central concern of AI. Learning concepts from examples is the most widely studied problem in machine learning. One of the two major learning techniques the **Similitude Based Learning (SBL)**, involves comparing several examples of a concept, searching for common features in order to define the concept to be learned. In this area, however, only a few formal approaches have been proposed (*e. g.* **[VALI84]**,**[AMST88]**,**[NATA88]**, etc.). These approaches prove to be fruitful and result in several learning algorithms, but they only work for fairly small sets of concepts. Our aim is to show how some of the learning paradigms can be reformulated using the logical formalism of circumscription which has been proposed by McCarthy to deal with the problems related with non-monotonic reasoning **[McCA80]**. Our approach shows that is possible to use these logical tools in order to obtain a formal unified vision of **SBL** paradigms, and it also suggests some kind of improvements on the current implementations. We introduce a hitherto unmentioned direct link between machine learning and circumscription. We believe that this new framework and the obtained results are of practical interest. In particular, we study Winston's well-known algorithm for learning structural descriptions **[WINS75** and Vere's inductive learning of concepts algorithm **[VERE78]**, and show how our approach allows an unified formal vision of both methods. We use a typical example of each method and reformulate them in terms of our approach to show its central ideas. From this it is easy to see that our method subsumes both approaches and could be applied to others learning paradigms.

2. Introduction

The **Closed World Assumption (CWA) [REIT78]**, the **Predicate completion Theory (PCT) [CLAR78]** and the **Circumscription Theory (CT) [McCA80]**

* Partially supported by scholarships from UABCS, BANXICO & Instituto Tecnológico de la Paz,B.C.S. **MÉXICO**

have been proposed as different approaches to the formalization of non-monotonic reasoning. Although these formalisms were presented independently, Reiter had showed that *sometimes* it is possible to consider the predicate completion as a special case of circumscription **[REIT82]**. Lifschitz also derived general conditions relating these two augmenting conventions, and had showed that when the **CWA** can be applied consistently to fixed domain theories, it produces the same result than parallel circumscription of all predicates in the theory **[LIFS85b]**. Those classes of theories have been also studied by Reiter in **[REIT84]**. On other hand, Gelfond have suggested that some versions of the **CWA** are equivalent to some particular forms of circumscription **[GELF89]**.

In sections 2.1, 2.2 and, 2.3 we introduce the main results of these formalisms to help in the comprehension of the examples of section 3.

2.1 The Closed World Assumption

The **CWA** is an important convention, usually implicit on data base design, that was studied by Reiter **[REIT78]** and appears to be a powerful logic tool for commonsense reasoning. Suppose that we have in a data base the list of people that have to paid their income tax. For such a data base, to be able to answer queries such as:

Did Ton Sales pay his income tax?

it would be useful to adopt the following convention: Persons whose name does not appear on the database did not pay their income tax this year. Without such convention, we would have to list explicitly all persons who have not yet paid every time we want to find out whether someone had paid his tax. This assumption, that seems to be – and is – fairly natural is an example of the **CWA**, and is used each time that we address these kind of queries to a conventional data base.

The **CWA** augments a theory $\mathbf{T}[\mathbf{\Delta}]$, which is the closure of a database $\mathbf{\Delta}$ (or a set of beliefs expressed as a finite set of first-order formulas) under logical entailment, adding to $\mathbf{\Delta}$ a set of assumed beliefs $\mathbf{\Delta_{asm}}$ comprising the negation of any ground atom (*i.e.* a formula without bound or free variables) that cannot be logically entailed from $\mathbf{\Delta}$. This is the naive form of the negation as failure rule (NF-rule). The augmented theory, $\mathbf{CWA}[\mathbf{\Delta}]$, can be stated succintly as follows:

$$\mathbf{CWA}[\mathbf{\Delta}] = \{\varphi \mid \mathbf{\Delta} \cup \mathbf{\Delta_{asm}} \vdash \varphi\}$$

The **CWA** does not always result in a consistent augmented theory $\mathbf{CWA}[\mathbf{\Delta}]$ as shown in the following example:

Example 2.1

Assume that:

$\mathbf{\Delta} = P \vee Q$.

Then neither P nor Q is in $\mathbf{T}[\mathbf{\Delta}]$, so by the **CWA** their negations are both in $\mathbf{CWA}[\mathbf{\Delta}]$ and therefore:

$$\mathbf{CWA}[\mathbf{\Delta}] = \{P \vee Q, \neg P, \neg Q, P, Q\}$$

which is not a consistent theory.

Minker was the first who note that the source of difficulties in example 2.1 is the fact that $\mathbf{\Delta}$ contains a disjunction of positive ground literals and no one of them could be deduced from $\mathbf{\Delta}$**[MINK82]**. To guarantee closure consistence he introduces a more restrictive assumption, called the **Generalized Closed-World Assumption (GCWA)** which applied to $\mathbf{\Delta}$ derives the following set of assumed beliefs:

$$\mathbf{\Delta}'_{\mathbf{asm}} = \{\ \neg P \mid P \text{ is a ground atom and there is no ground disjunction } B \text{ of atoms such that } \mathbf{\Delta} \vdash P \vee B \text{ and } \mathbf{\Delta} \not\vdash B\}$$

defining in this case the closure $\mathbf{GCWA}[\mathbf{\Delta}]$ of $\mathbf{\Delta}$ as the logic closure of the set $\mathbf{\Delta} \cup \mathbf{\Delta}'_{\mathbf{asm}}$. Minker, in this work, proves the consistence of the augmented theory $\mathbf{GCWA}[\mathbf{\Delta}]$, in the case of a consistent data base $\mathbf{\Delta}$.

Shepherdson proves the following consistency theorem for the $\mathbf{CWA}[\mathbf{\Delta}]$ from which it is easy to show the consistency of $\mathbf{GCWA}[\mathbf{\Delta}]$ in the cases where $\mathbf{\Delta}$ is consistent **[SHEP84]**:

Theorem 2.1

$\mathbf{CWA}[\mathbf{\Delta}]$ is consistent iff, for every positive ground-literal clause $L_1 \vee \ldots \vee L_n$ that follows from $\mathbf{\Delta}$, there is at least one ground literal L_i also entailed by $\mathbf{\Delta}$ that subsumes it. Equivalently, the $\mathbf{CWA}[\mathbf{\Delta}]$ augmentation of a consistent data base $\mathbf{\Delta}$, is inconsistent iff there are positive ground literals $L_1, \ldots, L_n$ such that

$$\mathbf{\Delta} \vdash L_1 \vee \ldots \vee L_n$$

but, for $i = 1, \ldots, n$, $\mathbf{\Delta} \not\vdash L_i$

It is easy to show that if $\mathbf{\Delta}$ only contains Horn clauses (*i.e.* clauses with at most one positive literal) then the conditions of the theorem hold, and the augmented theory is consistent for this important special case. On other hand, the consistency of $\mathbf{CWA}[\mathbf{\Delta}]$ is strongly dependent on the domain of the language terms. For example, by the application of the previous theorem,it is easy to show that the **CWA** augmention of

$$\mathbf{\Delta} = \{\forall x(P(x) \vee Q(x),)\, P(a),\, Q(b)\}$$

is consistent in the case that A and B are the only objects in the language, but if there is any other object then it is inconsistent. This problem has been treated by restricting the $\mathbf{CWA}[\mathbf{\Delta}]$ to the so called fixed-domain theories. In these kind of theories, the constant objects of language are restricted to a finite number of terms (**Domain Closure Assumption**), each one of them represents an unique object (**Unique Names Assumption**). When there are no constant functions in the language, these assumptions might be formally stated as:

~ **The Domain Closure Assumption (DCA)**

$$\forall x\, (x = t_1 \vee \ldots \vee x = t_n)$$

~ **The Unique Names Assumption (UNA)**

$$t_i \neq t_j \qquad \forall i\, \forall j \;\; i \neq j$$

where the t_i are all the constants objects of the language

The **CWA** is too strong for many applications. You do not always want to assume that *any* ground atom not provable from $\mathbf{\Delta}$ is false.

2.2 The Predicate Completion Theory

Another research direction motivated by non-monotonic reasoning and related problems, particularly those of taxonomic hierarchies, was the predicate completion of a set of clauses, described by Clark [**CLAR78**]. This approach consists of a kind of *minimization* principle which permits us to express, plainly, the assumption that the only objects that satisfy a property, or a set of properties in the case of parallel completion, are those that must be so deduced from our beliefs. It could be interpreted as special case of **CWA** which express that the only positive information of predicate P is that available in the theory (*i.e.* the information about P in the theory is complete). In this case, to avoid circular definitions that do not satisfy the mimimality assumption, the completion on predicate P is limited to the clauses with the following property : if they have a positive occurrence of P this occurrence is unique (solitary clauses). These clauses must be expressed in the following normal form:

$$\forall x\,(\exists y\,(x = t) \wedge Q_1 \wedge \ldots \wedge Q_n \Longrightarrow P(x))$$

where x is a tuple of variables not occurring in t, $(x = t)$ is an abbrevation for $(x_1 = t_1 \wedge \ldots \wedge x_n = t_n)$, and the variable y occurs only in the antecedent of the implication.

If there are exactly n solitary clauses in P, with $n > 0$, on normal form:

$$\forall x\,(E_i \Longrightarrow P(x)) \text{ for } i = 1, \ldots, n \tag{2.1}$$

$$\text{(Equivalently, } \forall x\,(E_1 \vee \ldots \vee E_n \Longrightarrow P(x)))$$

then the completion $\mathbf{COM}[\boldsymbol{\Delta}; P]$ of P in $\boldsymbol{\Delta}$ is defined by adding to $\boldsymbol{\Delta}$ the formula:

$$\forall x\,(P(x) \Longrightarrow E_1 \vee \ldots \vee E_n)$$

that is

$$\mathbf{COM}[\boldsymbol{\Delta}; P] = \boldsymbol{\Delta} \wedge (\forall x(P(x) \iff E_1 \vee \ldots \vee E_n))$$

Since all the E_i are solitary clauses, the composition $E_1 \vee \ldots \vee E_n$ does not contain P, and therefore it can be considered as the sufficient part of a definition of P. The completion consists essentially in assuming this composition as necessary for P. By this process we obtain a complete definition of this predicate in $\boldsymbol{\Delta}$, therefore, it can be considered as a formal procedure that leads a data base $\boldsymbol{\Delta}$ to the *learning* of the definition of P. This kind of learning is the most conservative that can be realized with the present information.

The **PCT** can be used in the formalization of non-monotonic reasoning, particularly for *default reasoning*. This kind of reasoning assumes as true (by default) all those typical properties that are not explicitly declared as false. The inheritance of characteristics from a class to a subclass, widely used in AI in hierarchical taxonomy systems, is an example of this type of reasoning. In these systems the inheritance class/subclass is implemented by default, and does not include the case of exceptions (qualifications). At the design phase the problem is to consider all the possible exceptions of a given rule, not only by their number but also because could exist a number of unknown exceptions.

McCarthy in [**McCA86**], proposed a technique to attack this problem, namely the qualification's problem, which consists in the consideration of the exceptions as *abnormalities*, that is as instances of an Abnormal predicate *(Ab)* whose satisfaction by any given object cancels the applications of the correspondent rule, namely the *inheritance cancellation rule.* The application of this technique gives a monotonic solution to the qualification's problem, in the sense that new exceptions are incorporated by means of additional instances of the Ab instead of having to modificate the previous rules. The classical ornithological example is introduced to ilustrate this technique.

Example 2.2

Let $\mathbf{\Delta}$ a database containing the following clauses:

$$\begin{aligned} \forall x\,(Bird(x) \wedge \neg Ab(x) &\Longrightarrow Flies(x)) \\ \forall x\,(Ostrich(x) &\Longrightarrow Ab(x)) \\ \forall x\,(Penguin(x) &\Longrightarrow Ab(x)) \end{aligned}$$

Expressing these clauses in the normal form (2.1), results:

$$\begin{aligned} \forall x\,(Bird(x) \wedge (\neg Flies(x)) &\Longrightarrow Ab(x)) \\ \forall x\,(Ostrich(x) \vee Penguin(x) &\Longrightarrow Ab(x)) \end{aligned}$$

therefore the completion of Ab in $\mathbf{\Delta}$ is:

$$\begin{aligned} \mathbf{COMP}[\mathbf{\Delta}; Ab] = \forall x (Ab(x) \iff & (Bird(x) \wedge (\neg Flies(x))) \vee \\ & (Ostrich(x) \vee Penguin(x))) \end{aligned}$$

(the only abnormal things are ostriches, or penguins, or birds that do not fly).

In the augmentation of a theory with respect to an Ab, used to manage the exceptions of a rule, the set of objects that satifies Ab is minimized, resulting in the maximization of the domain of the objects to which the rule applies. If you interpret this fact from the point of view of the **SBL**: the completion of the available information with respect to the negation of the target concept (the concept to be learned) results on the most general possible definition of it. This is the central idea applied in the examples of sections 3.1.1 and 3.2.1.

The parallel predicate completion, extends to a set of predicates the idea of considering the available information about a predicate P as complete. In this case, with the purpose of avoiding circularity and to assure the consistency of the augmented theory, it is assumed that the database *could be ordered in the set of predicates* $\{P_1, \ldots, P_n\}$ with respect to which the completion is made. This means that all clauses must be solitary for each P_i, and those clauses containing the predicates can be expressed as:

$$\forall x\,(E_i \Longrightarrow P_i(x))\ for\, i = 1, \ldots, n$$

where, for each i, E_i do not have any occurrence of the predicates P_i, P_{i+1}, $\ldots$, P_n, nor negative occurrences of P_1, $\ldots$, P_{i-1}. The completion is obtained by the addition of the n formulas:

$$\forall x\,(P_i(x) \Longrightarrow E_i)\ for\, i = 1, \ldots, n$$

2.3 The Circumscription Theory

Among the different approaches to handle the problems derived from incomplete information, specially of those problems on non-monotonic nature, the circumscription theory, initially proposed by J. McCarthy **[McCA80,86]** and developed by V. Lifschtiz **[LIFS85a,b,86]**, is considered by some authors as the most powerful tool to cope with those problems **[PRZY89, GELF89]**. One of its more important characteristics is that it is wholly based on the classical predicate logic.

In the same way as the predicate completion, the **CT** consists in augmenting a theory by the addition of a single formula that expresses the idea that the only available positive information about one predicate is that is present on the database. This formula is obtained by means of the concept of minimal models. We present a simplified version of **CT** in model-theoretic terms, following Genesereth's notation **[GENE88]**.

Let M and N be models of a theory T that only differ in the interpretation of a predicate P in T. Let $M \leq_P N$ mean that P's extension in M is a proper subset of its extension in N ($M <_P N$ means that $M \leq_P N$ and $N \not\leq_P M$).

The relation $\leq_P$ is a partial order over the set of all models of the theory. The minimal models with respect to this order (that does not necessarily exist) are called *P-minimals* or *minimals* in P. So a model M of the theory is *P-minimal* if a model N does not exist such that $N <_P M$.

Circumscription consists in obtaining of a single formula φ_P in terms of P and $\mathbf{\Delta}$, namely the *circumscription formula*, such that any augmented database's model ($\mathbf{\Delta} \wedge \varphi_P$) is a minimal model of $\mathbf{\Delta}$. This is a second-order formula and can be expressed as:

$$\varphi_P : \forall P^* \neg (\forall x(P^*(x) \implies P(x)) \wedge (\neg(\forall x(P(x) \implies P^*(x)))) \wedge \mathbf{\Delta}(P^*))$$

where P^* is a predicate of the same arity of P, and $\mathbf{\Delta}(P^*)$ is obtained substituting P^* by each occurrence of predicate P in $\mathbf{\Delta}$.

By means of the φ_P formula we formally express the fact that no predicate P^* exists that satifies $\mathbf{\Delta}$ when substituted by P, and whose extension is a proper subset of P. This fact, may be, becomes more evident if this formula is stated as:

$$\forall P^* ((\mathbf{\Delta}(P^*) \wedge (\forall x(P^*(x) \implies P(x)))) \implies \forall x (P(x) \implies P^*(x)))$$

As in the preceding sections **CT** does not always result in a consistent augmented theory. For example, an obvious case is when the database $\mathbf{\Delta}$ has not *P-minimal* models. For a consistent database $\mathbf{\Delta}$, Lifschitz shows the sufficiency of several sets of conditions for the consistency of **CT** **[LIFS85]**.

Theorem 2.2

The circumscription of a predicate P in a consistent database $\mathbf{\Delta}$ of separable formulæ is consistent.

This theorem will be used in the examples of the following sections. The *separable formulæ* in P are a generalization of the solitary clauses defined in section 2.2. They are a composition of conjuctions and disjunctions of formulæ of the type $\forall\, x\ (E(x) \implies P(x))$, with formulæ that do not have positive occurrences of P (here E is a formula without occurrences of P).

A special case of circumscription, of great interest in **SBL**, is that where all the $\mathbf{\Delta}'s$ formulæ could be expressed as:

$$N(P) \wedge (\forall x\, (E(x) \implies P(x)))$$

where $N(P)$ is a formula without positive occurrences of P and E do not have occurrences of P. Under this conditions the following theorem, formulated by Lifschitz **[LIFS85a]**, holds .

Theorem 2.3

$$\mathbf{CIRC}[\boldsymbol{\Delta}; P] = N(E) \wedge (\forall x\, (E(x) \iff P(x)))$$

where $N(E)$ is obtained by substituing E for each occurrence of P.

In this case the augmentation is constructed by the addition of a first-order formula to $\boldsymbol{\Delta}$. In these cases the circumscription formula is called *collapsable*. Theorem 2.3 is an special instance of a more general result that was recently shown by Rabinov **[RABI89]**. There are more general forms of circumscription as the Parallel Circumscription, Prioritized Circumscription and forms of Circumscription allowing varying predicates, etc. **[McCA86] [LIFS85a]**, that produce stronger results which are also relevant for the learning paradigms. We now proceed to explore these lines. An interesting case is that which allows a larger extension for some others predicates, namely the variable predicates, with the purpose of obtaining a smaller extension for those being minimized. For these cases the following theorem could be applied **[GENE88]**.

Theorem 2.4

If $\boldsymbol{\Delta}$ is of the form $N(Z) \wedge (\forall x\, (E(x) \Longrightarrow Z))$, where N has no positive occurrences of Z, and E has no occurrences of Z, then:

$$\mathbf{CIRC}[\boldsymbol{\Delta}; P; Z] = N(Z) \wedge (\forall x\, (E(x) \Longrightarrow Z)) \wedge \mathbf{CIRC}[N(E); P]$$

where *N(E)* is obtained by substituting E for each occurrence of Z and E, P and Z can be tuples of predicates.

Example 2.3

Using the theorem 2.3, it is easy to see that in the example 2.2 the circumscription of Ab in $\boldsymbol{\Delta}$ produces the same results than the completion of Ab in $\boldsymbol{\Delta}$, that is:

$$\begin{aligned}\mathbf{CIRC}[\boldsymbol{\Delta}; Ab] = \forall x\, (Ab(x) \iff (Bird(x) \wedge (\neg Flies(x)))\vee \\ (Ostrich(x) \vee Penguin(x)))\end{aligned}$$

On the other hand, as $\boldsymbol{\Delta}$ could be expressed in the form:

$$N(Flies) \wedge (\forall x\, (E(x) \Longrightarrow Flies(x)))$$

where:

$$\begin{aligned}N(Flies) &= \forall x\, (Ostrich(x) \vee Penguin(x) \Longrightarrow Ab(x)) \\ E(x) &= Bird(x) \wedge (\neg Ab(x))\end{aligned}$$

and $N(Flies)$ has not positive occurrences of $Flies$, and $E(x)$ has no occurrences of $Flies$. Then if it is allowed the variation of predicate $Flies$, from theorem 2.4 we obtain:

$$\begin{aligned}\mathbf{CIRC}[\boldsymbol{\Delta}; Ab; Flies] &= \boldsymbol{\Delta} \wedge \mathbf{CIRC}[N(E); Ab] \\ &= \boldsymbol{\Delta} \wedge \mathbf{CIRC}[\forall x\, ((Ostrich(x) \vee Penguin(x)) \Longrightarrow Ab(x)); Ab] \\ &= \boldsymbol{\Delta} \wedge (\forall x\, ((Ostrich(x) \vee Penguin(x)) \iff Ab(x)))\end{aligned}$$

so we get, a more restrictive characterization of the abnormal predicate Ab. (The only abnormal things are either ostriches or penguins).

3. Two examples of SBL using circumscription.

3.1 Learning structural descriptions from examples. Winston's method

The well-known Winston's learning algorithm **[WINS75]** has as its starting point a set of primitive relations to describe the relationships between objects, and a set of basic objects (the toy world). In this world the events are scenes which are represented by a labeled graph where objects are represented as the nodes and the arrows represent the primitive relations. The system learns, with the assistantship of a "Professor", by means of examples and *near-miss* examples.

We use the classical example of learning the concept of *arch*, to show how the circumscriptional approach allows us to obtain a definition of this concept by considering only the *relevant* characteristics. This approach makes explicit all the logical assumptions that are implicit in Winston's approach.

We will assume implicit universal quantification, and use an abbreviated subset of relations used by Winston in **[WINS75]**, just to simplify the reading of the resulting expressions:

SUPPORTED-BY (**SPB**) A-KIND-OF (**AKO**) TOUCH (**T**)
STANDING (**S**) LYING (**L**)

Also B is used as the abbreviation for BRICK. The learning process begins by showing the system a picture of a scene (see fig. 3.1) which is a positive example of the target concept. The system represents this scene as labeled-graph, as shown in figure 3.2, which is the representation of an arch.

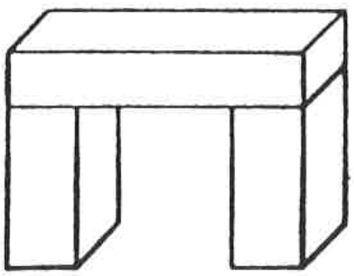

Figure 3.1

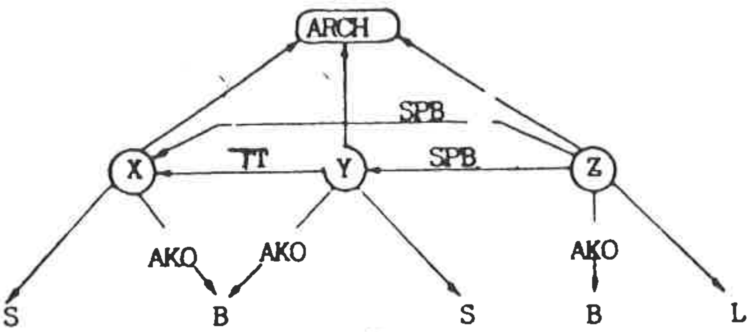

Figure 3.2

Then, the scene of figure 3.3 is presented to the system as the first *near-miss* of arch. By means of a comparison between the graph resulting of this negative example against the previous one, the systems indentifies as compulsory the fact that the bricks

X and Y must support Z; marking as necessaries both support relations in the graph of figure 3.2.

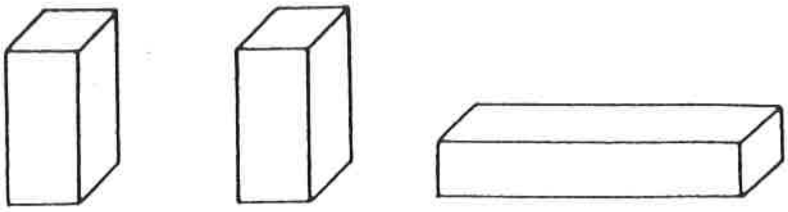

Figure 3.3

If is presented now to the system the scene of the new *near-miss* showed in figure 3.4, it repeats the comparison procedure between the graph produced from this figure and the graph obtained in the previous stage. In this case, it marks as necessary the arrow corresponding to the relation $\neg\mathbf{T}(X, Y)$.

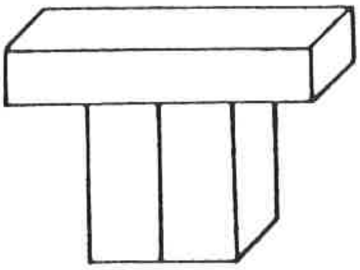

Figure 3.4

If the learning process finishes at this point, the system's representation of the arch concept is the graph of the figure 3.2, but with the difference that now, the relationships that have been marked as necessary could not be eliminated by further *near-misses.*

3.1.1. Circumscriptional approach

In this section we will show how the conclusions obtained by Winston's method can be reformulate using the circumscriptional approach; allowing the formalization of those conclusions.

The information represented in the graph (see fig. 3.2) could be expressed, under our approach, as the following first-order expression:

$$\begin{aligned} &\mathbf{AKO}(B, X) \wedge \mathbf{AKO}(B, Y) \wedge \mathbf{AKO}(B, Z) \wedge (\neg\mathbf{T}(X, Y)) \\ &\wedge \mathbf{SPB}(Z, X) \wedge \mathbf{SPB}(Z, Y) \wedge \mathbf{S}(Y) \wedge \mathbf{S}(X) \wedge \mathbf{L}(Z) \\ &\Longrightarrow ARCH(X, Y, Z) \end{aligned} \tag{3.1}$$

As stated in the previous section Winston's algorithm concludes from the first *near-miss* (see figure 3.3) that both X and Y must support Z, in first order notation, this fact is represented as:

$$ARCH(X,Y,Z) \Longrightarrow \mathbf{SPB}(Z,X) \wedge \mathbf{SPB}(Z,Y) \tag{3.2}$$

or equivalently:

$$\neg(\mathbf{SPB}(Z,X) \wedge \mathbf{SPB}(Z,Y)) \Longrightarrow \neg ARCH(X,Y,Z) \tag{3.3}$$

The conclusion obtained by Winston's method from the second *near-miss* is expressed as:

$$ARCH(X,Y,Z) \Longrightarrow \neg\mathbf{T}(X,Y)$$

which is equivalent to:

$$\mathbf{T}(X,Y) \Longrightarrow \neg ARCH(X,Y,Z) \tag{3.4}$$

Therefore, if we consider that our database $\mathbf{\Delta}$ is formed by the formulæ 3.1, 3.3 and 3.4, then their combination could be expresed in the normal form, $N(P) \wedge (E \Longrightarrow P)$, required by theorem 2.3. Here:

$$P = (\neg ARCH(X,Y,Z))$$
$$E = \neg(\mathbf{SPB}(Z,X) \wedge \mathbf{SPB}(Z,Y)) \vee \mathbf{T}(X,Y)$$

and $N(P)$ is the clause:

$$(\neg\mathbf{AKO}(B,X)) \vee (\neg\mathbf{AKO}(B,Y)) \vee (\neg\mathbf{AKO}(B,Z)) \vee \mathbf{T}(X,Y) \vee (\neg\mathbf{SPB}(Z,X))$$
$$\vee(\neg\mathbf{SPB}(Z,Y)) \vee (\neg\mathbf{S}(Y)) \vee (\neg\mathbf{S}(X)) \vee (\neg\mathbf{L}(Z)) \vee (\neg(\neg ARCH(X,Y,Z)))$$

which is equivalent to the formula 3.1. Applying this theorem to obtain the circumscription of $\mathbf{\Delta}$ with respect to $\neg ARCH$ yields:

$$\mathbf{CIRC}[\mathbf{\Delta}; \neg ARCH] = \forall X \forall Y \forall Z\, (\neg ARCH(X,Y,Z) \iff$$
$$\neg(\mathbf{SPB}(Z,X) \wedge \mathbf{SPB}(Z,Y)) \vee \mathbf{T}(X,Y))$$

which allows to express the ARCH definition as:

$$ARCH(X,Y,Z) \iff (\mathbf{SPB}(Z,X) \wedge \mathbf{SPB}(Z,Y)) \wedge (\neg\mathbf{T}(X,Y)) \tag{3.5}$$

Since definition of $\neg ARCH$ has been obtained minimizing its extension, then the ARCH's definition 3.5 is the most general that can be obtained using the information given to the system.

Under this *circumscriptional* approach, learning consists of a process that goes from the most general definition of a concept, allowed by the first *near-miss*, to more specific characterizations, obtained using new *near-misses*, until getting an *adequate* specificity, which is not necessarily the same as that of the concept. In this case, it is not necessary to eliminate the added noise, as it is eliminated with the first *near-miss*. However, Winston's learning algorithm proceeds inversely. It is a generalization technique for learning concepts. The non compulsory relations are added by positive examples (noise), which produce excessive restrictions on the concept's characterization that have to be eliminated by the introduction of positive examples; if these new examples are not carefully chosen by the professor, they could introduce new disturbances. Another well-known limitation of this approach is that, as the number of examples increases, rules become overly specific and thus difficult to understand and debug. This process finishes when the professor decides that the system has an adequate characterization which, eventually, could match the concept.

On other hand, although Winston, in his work **[WINS75]** textually says that:

> "If two differences are found, either of them may be sufficient to be a *near-miss*, while the other difference may be equally sufficient or merely irrelevant"

Although, from the *near-miss* of figure 3.3 his system concludes, in the same work, that both support relations are crucial. However, the correct conclusion must be that one of them or both are necessary, which in our notation can be expressed as:

$$ARCH(X,Y,Z) \Longrightarrow \mathbf{SPB}(Z,X) \vee \mathbf{SPB}(Z,Y) \tag{3.6}$$

If this expression is used instead of expresion 3.3, the definition of arch becomes:

$$ARCH(X,Y,Z) \iff (\mathbf{S}(Z,X) \vee \mathbf{S}(Z,Y)) \wedge (\neg\mathbf{T}(X,Y))$$

Now, even expression 3.6 is not correct without the supposition that arch's concept can be completely characterized by a subset of the initial relations set presented in the initial example (see expression 3.1). The same holds for expression 3.4.

In general, for the logical validation of the conclusions obtained by Winston's algorithm, you must suppose that the examples contain enough information (relations) to result on a complete characterization of the target concept as a subset of those relations.

The following theorem makes explicit the logical assumptions implicit on Winston's method, and justifies the conclusions obtained from the graph comparation. The proof of this theorem is in Appendix A.

Theorem 3.1

Assume a set of predicates $\mathbf{\Pi} = \{P_1, \ldots, P_n\}$ whose conjuction is a sufficient condition for the complete characterization of a concept C, that is:

$$\bigwedge_{i \in N} P_i \Longrightarrow C \text{ where } (N = \{1, \ldots, n\}) \tag{3.7}$$

Also assume that the concept C can be completely characterized by the conjunction of a subset of $\mathbf{\Pi}$, that is:

$$\exists S \subset N \text{ such that } C \iff \bigwedge_{i \in S} P_i \tag{3.8}$$

If there exists a subset T of N and constant objects $o_1, \ldots, o_k$ such that:

$$((\bigwedge_{i \in T} P_i) \wedge (\bigwedge_{i \in N \setminus T} (\neg P_i)))\,(o_1, \ldots, o_k) \text{ is true} \tag{3.9}$$

and

$$(\bigwedge_{i \in T} P_i) \wedge (\bigwedge_{i \in N \setminus T} (\neg P_i)) \Longrightarrow \neg C \tag{3.10}$$

then

$$C \Longrightarrow \bigvee_{i \in N \setminus T} P_i \tag{3.11}$$

3.2. Inductive learning of concepts. Vere's method

Vere, [VERE78], has proposed a learning method which is an inductive generalization using examples and counter-examples. The concepts are characterized by conjuctions and disjunctions of predicates. The abstraction process is based upon an operation of *inductive substitution*, where some given occurrences of specific terms are substituted for variables. Those variables must not appear previously in the expression, and for different terms the use of different variables is necessary. For the sake of simplicity, we only consider those substitutions where all occurrences of the same term are substituted for the same variable.

Let E_1 and E_2 be first-order formulæ expressed in the normal conjunctive form. E_1 is *more general* than E_2, if there exists a substitution σ such that the set of clauses of E_1 is a subset of the set of clauses σE_2. Here, σE_2 is the resulting expression from the application of σ to E_2. The intersection and the difference of the set of clauses of E_1 and σE_2, are respectively called the *coupling* and the *remainder.*

Let us introduce some abbreviations for simplicity:

ON **(O)** SPHERE **(S)** CUBE **(C)** PYRAMID **(P)**
YELLOW **(Y)** RED **(R)** GREEN **(G)** BLUE **(B)**

Example 3.1

Let us consider the expression:

$$\mathbf{O}(A,B) \wedge \mathbf{S}(A) \wedge \mathbf{C}(B) \wedge \mathbf{G}(A) \wedge \mathbf{G}(B) \tag{3.12}$$

If we replace A by X and B by Y we obtain:

$$\mathbf{O}(X,Y) \wedge \mathbf{S}(X) \wedge \mathbf{C}(Y) \wedge \mathbf{G}(X) \wedge \mathbf{G}(Y)$$

As it has been explained, the formula:

$$\mathbf{O}(X,Y) \wedge \mathbf{S}(X) \wedge \mathbf{G}(X) \wedge \mathbf{G}(Y)$$

is one of the possible generalizations of (3.12).

The *Maximum common generalization* **(Mcg)** of two expressions is a generalization of both, with the additional characteristic that it cannot be more general than any other common generalization.

Vere's learning algorithm essentially consists in:

~ The professor gives the system a set of examples and counter-examples. Then the system builds, by means of an iterative process, the **Mcg** of all examples, taking the examples by pairs, and exploring all the possible couplings.

~ If the **Mcg** obtained includes some counter-examples, then the **Mcg** of the remainders of these counter-examples is introduced as restrictive term. These remainders are obtained through the elimination, from the clausal form of the counter-examples, of those clauses that are instances of any clauses present in the **Mcg**. In this step the learned concept is:

$$C_1 = \mathbf{Mcg} \wedge (\neg N_1)$$

If the concept C_1 excludes some examples then it is added to N_1 a new negative condition N_2, and successively. In general, the target concept C is expressed as:

$$C = \mathbf{Mcg} \wedge \neg(N_1 \wedge \neg(N_2 \wedge \neg(\ldots \wedge \neg N_k)))$$

We use the following example, which has been extracted from **[LOPE87]**, to ilustrate our approach in the case of inductive learning. In this case, the Vere's method begins obtaining the **Mcg** of four examples of figure 3.5; which is:

$$C_0 = \mathbf{O}(X, Y) \wedge \mathbf{C}(Y) \wedge \mathbf{G}(Y)$$

Examples

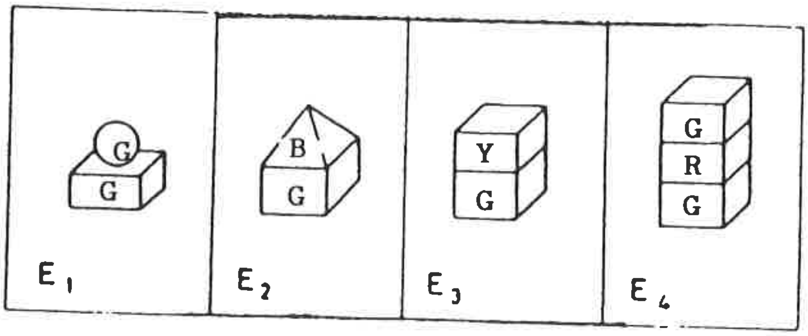

Figure 3.5

$$E_1 = \mathbf{O}(O_1, O_2) \wedge \mathbf{S}(O_1) \wedge \mathbf{C}(O_2) \wedge \mathbf{G}(O_1) \wedge \mathbf{G}(O_2)$$
$$E_2 = \mathbf{O}(O_3, O_4) \wedge \mathbf{P}(O_3) \wedge \mathbf{C}(O_4) \wedge \mathbf{B}(O_3) \wedge \mathbf{G}(O_4)$$
$$E_3 = \mathbf{O}(O_5, O_6) \wedge \mathbf{C}(O_5) \wedge \mathbf{C}(O_6) \wedge \mathbf{Y}(O_5) \wedge \mathbf{G}(O_6)$$
$$E_4 = \mathbf{O}(O_7, O_8) \wedge \mathbf{O}(O_9, O_7) \wedge \mathbf{C}(O_7) \wedge \mathbf{C}(O_8) \wedge \mathbf{C}(O_9) \wedge \mathbf{R}(O_7) \wedge \mathbf{G}(O_8) \wedge \mathbf{G}(O_9)$$

Counter-examples

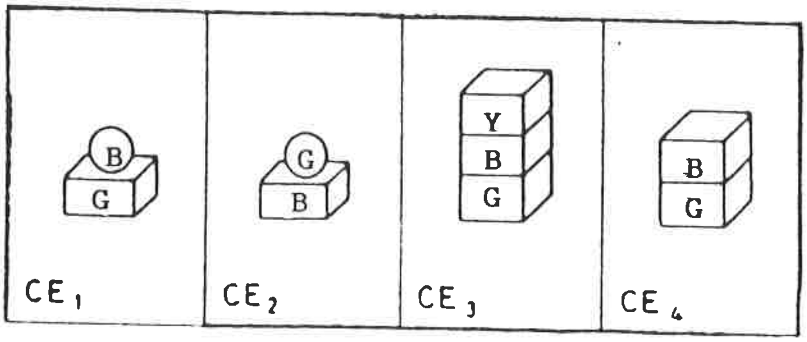

Figure 3.6

$$CE_1 = \mathbf{O}(P_1, P_2) \wedge \mathbf{S}(P_1) \wedge \mathbf{C}(P_2) \wedge \mathbf{B}(P_1) \wedge \mathbf{G}(P_2)$$
$$CE_2 = \mathbf{O}(P_3, P_4) \wedge \mathbf{S}(P_3) \wedge \mathbf{C}(P_4) \wedge \mathbf{G}(P_3) \wedge \mathbf{B}(P_4)$$
$$CE_3 = \mathbf{O}(P_5, P_6) \wedge \mathbf{O}(P_7, P_5) \wedge \mathbf{C}(P_7) \wedge \mathbf{C}(P_5) \wedge \mathbf{C}(P_6) \wedge \mathbf{Y}(P_7) \wedge \mathbf{B}(P_5) \wedge \mathbf{G}(P_6)$$
$$CE_4 = \mathbf{O}(P_8, P_9) \wedge \mathbf{C}(P_8) \wedge \mathbf{C}(P_9) \wedge \mathbf{B}(P_8) \wedge \mathbf{G}(P_9)$$

The counter-example CE_2 is eliminated by C_1 because the cube P_4 is not green. As CE_1, CE_3, CE_4 are included by C_0, it is necessary to introduce a corrective term N_1, which is the maximum common generalization of the counter-examples remainders R_1, R_3, R_4.

$$R_1 = \mathbf{S}(X) \wedge \mathbf{B}(X)$$
$$R_3 = \mathbf{O}(Z,X) \wedge \mathbf{C}(Z) \wedge \mathbf{C}(X) \wedge \mathbf{Y}(Z) \wedge \mathbf{B}(X)$$
$$R_4 = \mathbf{C}(x) \wedge \mathbf{B}(X)$$

From this expressions is easy to obtain $N_1 = \mathbf{B}(X)$, therefore the new characterization of the target concept is:

$$C_1 = C_0 \wedge (\neg \mathbf{B}(X))$$

as C_1 does not include example E_2, is obtained a new remainder with the E_2's clauses not covered by the C_1's clauses. This remainder is $\mathbf{P}(X)$, which is its own maximum common generalization, then the concept, finally, is characterized as:

$$C = C_0 \wedge (\neg(\mathbf{B}(X) \wedge (\neg \mathbf{P}(X)))) \tag{3.13}$$

3.2.1. Circumscriptional approach

Under our circumscriptional approach, the previous example can be reformulated as follows. The previous algorithm implicitly assumes that the **Mcg** of a set of examples is a necessary condition for the concept's characterization, that is: $C \Longrightarrow$ **Mcg**. In this case results that: $C \Longrightarrow C_0$, or equivalently:

$$\neg C_0 \Longrightarrow \neg C \tag{3.14}$$

On other hand, making the assumption that the generalized expression of each counter-example is a sufficient condition for $\neg C$, replacing the correspondent terms of each predicate of the counter-examples CE_1, CE_3, CE_4, by the same variables , and grouping the obtained expressions in a single one, results:

$$C_0 \wedge (R_1 \vee R_3 \vee R_4) \Longrightarrow \neg C \tag{3.15}$$

And besides:

$$\begin{aligned} R_1 \vee R_3 \vee R_4 &= \mathbf{B}(X) \wedge (\mathbf{S}(X) \vee \mathbf{C}(X) \wedge (\mathbf{O}(Z,X) \wedge \mathbf{C}(Z) \wedge \mathbf{Y}(Z) \wedge \mathbf{C}(X))) \\ &= \mathbf{B}(X) \wedge (\mathbf{S}(X) \vee \mathbf{C}(X)) \end{aligned} \tag{3.16}$$

From the last expression, and 3.15 results:

$$C_0 \wedge \mathbf{B}(X) \wedge (\mathbf{S}(x) \vee \mathbf{C}(x)) \Longrightarrow \neg C \tag{3.17}$$

Let us assume that our database $\mathbf{\Delta}$ is formed by the expressions 3.14 and 3.17, then the completion of $\neg C$ in $\mathbf{\Delta}$ is:

$$\mathbf{COM}[\mathbf{\Delta};\neg C] = \forall X\, ((\neg C_0) \vee (C_0 \wedge \mathbf{B}(X) \wedge (\mathbf{S}(X) \vee \mathbf{C}(X)))) \iff \neg C$$

From here the following characterization of concept C is obtained:

$$C \iff C_0 \wedge (\neg(\mathbf{B}(X) \wedge (\mathbf{S}(X) \vee \mathbf{C}(X)))) \tag{3.18}$$

This results shows that if the domain of variables X, Y, Z is not restricted to the set **Ob** = {Sphere, Cube, Pyramid}, our approach obtains a more restricted characterization of the concept C than the obtained with Vere's algorithm (see expression (3.13)). The explanation for this is due to the fact that we do not restrict the domain and then the characterization of $\neg C$ is wider; therefore the characterization of C is more limited.

It is important to remark that the restriction of the set of variables X, Y, Z to the set **Ob** is precisely the application of the Domain-Closure Assumption. If this assumption is applied then results the next expression:

$$\forall X(\mathbf{S}(X) \vee \mathbf{C}(X) \iff (\neg \mathbf{P}(X)))$$

So, if the **DCA** holds, then (3.18) is equivalent to:

$$C \iff C_0 \wedge (\neg(\mathbf{B}(X) \wedge (\neg(\mathbf{P}(X)))))$$

which is the same resulting expression obtained by Vere's algorithm.

On other hand, in the completion process we did not take in account the CE_2, just to compare our proccess with that of Vere. However, if we take all the counter-examples, it is easy to find out that the same characterization of concept C is obtained as when the CE_2 is not considered. Therefore, our approach consists only in the completion of the negation of the target concept in the database formed by the expression 3.14, and the generalized expressions whose state the sufficiency of the descriptions of the counter-examples for $\neg C$. This result can be generalized to the learning of any concept C.

4. Conclusions and Future Work

This work introduces a formal approach to the study of some of the implicit assumptions in the learning paradigms. We have concentrated on the analysis of two classical methods on **SBL**, and showed how the circumscriptional approach generalize those methods. We believe that this new framework and the obtained results are of practical interest.

Theorem 3.1 brings new light to the inside of the problem of structural description learning by making explicit all the logical implicit assumptions and proving their consistency (see appendix A).

In section 3.2.1, we present a strong demonstration of our approach including not only circumscription but the **DCA** applied to inductive learning of concepts. We think that this result could easily be generalized.

Our purpose is to extend our study to other major learning technique, like the **Explanation Based Learning (EBL)** and, to the **ID-3** family. Also we want to catch the possible links, under our approach, between **EBL** and **SBL**. All this without forget the implementation side of this approach.

Acknowledgements

We thank Ton Sales for his wise help and constant friendship and, to Celestí Roselló and José L. Balcázar TEXwizards.

Appendix A

Definitions

1. Let n be a natural number ($n > 0$) and, let $N = \{1, \ldots, n\}$. For each subset $X_m = \{x_{m1}, \ldots, x_{mm}\}$ ($m1 < \ldots < mm$), of the set $\{x_1, \ldots, x_n\}$ we will define $P(X_k) = P(x_{m1}, \ldots, x_{mm})$.

2. Let $P_1, \ldots, P_n$ be predicates of arities $a_1, \ldots, a_n$ respectively, and let:

$$a = \sum_{i=1}^{n} a_i$$

Also let k be a natural number such that $a_i \leq k \leq a$, for $i = 1, \ldots, n$. For each fixed n-tuple of subsets $X_{a_1}, \ldots, X_{a_n}$ of the set $\{x_1, \ldots, x_k\}$, of $a_1, \ldots, a_n$ elements respectively, we define formulæ in the variables $x_1, \ldots, x_k$, as follows:

$$\forall x_1 \ldots \forall x_k \, (\bigwedge_{i \in L} P_i)(x_1, \ldots, x_k) = \bigwedge_{i \in L} P_i(X_{a_i})$$
$$\forall x_1 \ldots \forall x_k \, (\bigvee_{i \in L} P_i)(x_1, \ldots, x_k)) = \bigvee_{i \in L} P_i(X_{a_i})$$

For each subset $L \in N$.

In the proof of theorem 3.1 we will assume that exists an specific n-tuple of subsets $X_{a_1}, \ldots, X_{a_n}$ and a natural number k, associated to the set of predicates. Also for each subset L of N, we will write respectively:

$$\bigwedge_{i \in L} P_i \quad \text{and} \quad \bigvee_{i \in L} P_i$$

instead of:

$$\forall x_1 \ldots \forall x_k \, (\bigwedge_{i \in L} P_i)(x_1, \ldots, x_k)$$
$$\forall x_1 \ldots \forall x_k \, (\bigvee_{i \in L} P_i)(x_1, \ldots, x_k)$$

Proof of theorem 3.1. (By *Contradiction*). Let us consider expression 3.11 to be false, then it follows:

$$C \wedge \neg(\bigvee_{i \in N \setminus T} P_i) \text{ is true for some } k\text{-tupla of objects}$$

From this and 3.8 it can be concluded that $S \cap (N \setminus T) = \emptyset$, this is, $S \subset T$. Therefore, from this fact, 3.8 and 3.10, it can be concluded that:

$$(\bigwedge_{i \in T} P_i) \wedge (\bigwedge_{i \in N \setminus T} (\neg P)) \Longrightarrow \neg(\bigwedge_{i \in T} P_i)$$

but also, obviously:

$$(\bigwedge_{i \in T} P_i) \wedge (\bigwedge_{i \in N \setminus T} (\neg P)) \Longrightarrow \bigwedge_{i \in T} P_i$$

From the last two expressions and 3.9 results a contradiction. ∎

5. Bibliography

[AMST88] Amsterdam, J. *Extending the Valiant Learning model.* In Proceedings of the Fifth International Conference on Machine Learning. (1988) pp. 381–394.

[BESN89] Besnard, P. *The Completeness Problem for First-Order Circumscriptions.* In Proceedings AISB–89 (1989). pp.71–78.

[CLAR78] Clark, K. *Negation as failure.* In H. Gallaire and J. Minker (Eds.), *Logic and Data Bases* (Plenum Press, New York, 1978) pp 293–322.

[GENE88] Genesereth, M. R., Nilsson, J. N. *Logical Foundations of Artificial Intelligence.* (Morgan Kaufmann Publishers. Los Altos, CA. 1988)

[GELF89] Gelfond, M., Przymusinka, H., Przymusinki, T. *On the Relationships between Circumscription and Negation as Failure.* Art. Int. 38. (1) (1989) pp. 75–94.

[LIFS85a] Lifschitz, V. *Computing Circumpscription.* In *Proceedings IJCAI–85*, Los Angeles, CA. (1985) pp. 121–127.

[LIFS85b] Lifschitz, V. *Close-World Databases and Circumscription.* Art. Int. 27 (2) (1985) pp. 229–235.

[LIFS86] Lifschitz, V. *On the Satisfiability of Circumscription.* Art. Int. 28 (1) (1986) pp. 17–27.

[LOPE87] López de Mántaras, R. & Agustí. *Técnicas de aprendizaje: aprendizaje simbólico.* In Mompín, J. (Coordinador). *Inteligencia Artificial. Conceptos, técnicas y aplicaciones.* (Marcombo, 1987).

[McCA80] McCarthy, J. *Circumscription -A form of non-monotonic reasoning.* Art. Int. 13, (1985) pp. 27–39.

[McCA86] McCarthy, J. *Applications of circumscription to formalizing Commonsense Knowledge* . Art. Int. 28, (1986) pp. 89–116.

[MINK82] Minker, J. *On indefinite data bases and the closed-world assumptions.* In *Proceedings Sixth Conference on Automated Deduction* (1982) pp. 292–308.

[NATA88] Natarajan, B. K. and Tadepalli, P. *Two New Frameworks for Learning.* In Proceedings of the fifth international conference on machine learning. (1988) pp. 402–415.

[RABI89] Rabinov, A. *A generalization of Collapsable Cases of Circumscription.* Art. Int. 38 (1) (1989) pp.111–117.

[REIT78] Reiter, R. *On Closed-world data bases.* In H. Gallaire and J. Minker (Eds.), *Logic and Data Bases* (Plenum Press, New York, 1978) pp. 55–76.

[REIT82] Reiter, R. *Circumscription implies predicate completion (sometimes).* In *Proceedings of the National Conference on Artificial Intelligence.* Pittsburgh PA. Morgan Kaufmann, Los Altos, CA. (1982). pp. 418–420.

[REIT84] Reiter, R. *Towards a logical reconstruction of relational data base theory.* In M. Brodie *et al* (Eds.), *On Conceptual Modeling.* Springer-Verlag. Berlin (1984).

[SHEP84] Shepherdson, J.C. *Negation as Failure: A Comparision of Clark's Completed Data Base and Reiter's Closed-World Assumption.* Logic Programming 1 (1) (1984) pp. 51–79.

[VALI84] Valiant, L. *A theory of learnable.* Comm. of the ACM, 27(11) (1984) pp. 1134–1142.

[VERE78] Vere, S. A. *Inductive Learning of Relational Productions.* In Waterman & Hayes-Roth (Eds.), *Pattern Directed Inference Systems* (Academic Press, 1978).

[WINS75] Winston, P. H. *Learning structural descriptions from examples.* In P. H. Winston (Ed.) *The psychology of computer vision.* (McGraw-Hill, New York, 1975).