Some Computational Aspects of Circumscription

Phokion G. Kolaitis
Department of Computer Science
Stanford University
Stanford, CA 94305

Christos H. Papadimitriou
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093

Abstract

We explore the effects of circumscribing first-order formulae from a computational standpoint. First, extending work of V. Lifschitz, we show that the circumscription of any existential first-order formula is equivalent to a first-order formula After this, we establish that a set of universal Horn clauses has a first-order circumscription if and only if it is bounded (when considered as a logic program); thus it is undecidable to tell whether such formulae have first-order circumscription. Finally, we show that there are first-order formulae whose circumscription has a coNP-complete model-checking problem

1 Introduction

Circumscription, introduced by McCarthy [McC80, McC86], has turned out to be an influential formalism for common-sense reasoning Circumscription transforms logical formulae by adding a requirement of minimality, so that the circumscription of a formula $\phi(P)$, where P is a predicate symbol, asserts not only that P satisfies ϕ, but also that *no* proper subset of P satisfies ϕ The circumscription of a first-order formula is, on the face of it, a second order formula (a second-order quantifier ranging over predicates P' is needed to express minimality of P) This increase in logical complexity has important adverse consequences; for example, the inclusion of second-order formulae in a first-order set of beliefs rules out in principle a complete deductive system. Some of these difficulties have been pointed out before by several researchers, including Davis [Da80] and Schlipf [Sc86]

There have been numerous attempts to contain these adverse effects of circumsciption One fruitful direction has been to identify classes of first-order formulae whose circumscription is equivalent to a first-order formula Several such sufficient conditions for first-orderness have been proved in the recent past [Li85]; for a detailed exposition see also [GN87] These conditions are not necessary, since, as Lifschitz [Li85] observed, the circumscription of the formula $\exists x P(x)$ is first-order, although this formula is not contained in the classes identified by [Li85] In this paper we give a new such sufficient (although not necessary) condition, by establishing that *every existential first-order formula has a first-order circumscription*, thus generalizing the above example

In view of the above positive results for the existential formulae, it is natural to ask whether or not similar conclusions hold for the dual class of *universal first-order formulae* — of interest in Artificial Intelligence, because quite often belief statements or common-sense facts are expressed as universal sentences Lifschitz [Li85], however, pointed out that the circumscription of a universal first-order formula need not be equivalent to a first-order formula Indeed, if $\phi(P)$ is a universal first-order sentence asserting that P is a transitive binary relation containing the binary relation E, then the circumscription $\phi^*(P)$ states that P is the *transitive closure* of E (cf example 8 in [Li85]). Thus, $\phi^*(P)$ is *not* equivalent to a first-order sentence, because, using the compactness theorem of logic, it is easy to see that transitive closure is not first-order definable. Notice that $\phi(P)$ is actually a *conjunction of function-free Horn clauses*, since it can be written as

$$(\forall x \forall y)(E(x,y) \rightarrow P(x,y)) \wedge$$
$$(\forall x \forall y \forall z)(P(x,y) \wedge P(y,z) \rightarrow P(x,z)).$$

Recall that formulae that are conjunctions of (universally quantified) Horn clauses can also be thought of as *logic programs* and are, therefore, of special relevance to Artificial Intelligence Our second result characterizes precisely those formulae in this class (equivalently, logic programs) that have a first-order expressible circumscription More specifically, we show that the conjunction of function-free Horn clauses has a first-order circumscription if and only if the corresponding logic program is *bounded.* Boundedness of logic programs has been recently studied by several researchers in database theory ([GMSV87], [Va88], [CGKV88]) From these works and the above characterization, it follows that it is an undecidable problem to determine whether or not the circumscription of such a sentence is expressible in first-order logic

Finally, we point out, we believe for the first time, another adverse effect of passing from a first-order formula to its circumscription *Model-checking becomes computationally intractable* As is well-known, any set of first-order formulae has a polynomial-time (in fact, logarithmic-space) algorithm for testing whether a finite structure satisfies it In contrast, we exhibit a first-order formula whose circumscription is coNP-complete (and thus cannot be checked in polynomial time, unless P=NP).

This extended abstract is organized in five sections (this is the end of the first). In Section 2 we show our positive result for the circumscription of existential first-order formulae. Section 3 contains the characterization theorem for the circumscription of logic programs and the computational consequences of this result. In Section 4 we study the complexity of model-checking for circumscription, and we conclude in Section 5 by discussing some questions left open by this work Detailed proofs will be provided in the full paper

2 The Circumscription of Existential Formulae

Let $\phi(\mathbf{P})$ be a first-order or a second-order formula with equality, involving the sequence $\mathbf{P} = (P_1, \ldots, P_k)$ of predicate symbols and possibly other predicate symbols from a fixed underlying vocabulary σ. Following McCarthy [McC80, McC86] and Lifschitz [Li85], we define the *circumscription of* $\mathbf{P}$ *in* $\phi(\mathbf{P})$ to be the following second-order formula $\phi^*(\mathbf{P})$:

$$\phi(\mathbf{P}) \wedge (\forall \mathbf{P}')[(\mathbf{P}' < \mathbf{P}) \rightarrow \neg\phi(\mathbf{P}')],$$

where $\mathbf{P}' = (P_1', \ldots, P_k')$ is a sequence of predicates and $\mathbf{P}' < \mathbf{P}$ means that $P_i' \subset P_i$, $1 \leq i \leq k$, and there is a $j \leq k$ such that P_j' is a proper subset of P_j.

Several interesting cases have been pointed out in the recent past, in which the circumscription of a first-order formula collapses to a first-order formula. Lifschitz [Li85] showed that this holds for the class of *separable* formulae, a natural and fairly wide class that includes all quantifier-free formulae. Such results, reducing the logical complexity of circumscription from second-order to first-order, are potentially valuable, in view of the intractability of second-order logic on the one hand and the completeness theorem for first-order logic on the other.

We show below that the same holds for *all existential formulae*.

Theorem 1. Suppose that $\phi(\mathbf{P})$ is an existential first-order sentence of the form $\exists \mathbf{x}\psi$, where $\mathbf{x} = (x_1, \ldots x_l)$ is a sequence of variables and ψ is quantifier-free formula. Then the circumscription $\phi^*(P)$ of $\mathbf{P}$ in $\phi(\mathbf{P})$ is equivalent to a first-order formula. □

The proof of Theorem 1 constructs a first-order formula equivalent to the circumscription. We start by bringing ψ in its *complete disjunctive normal form*, that is, ψ is written as the disjunction of several formulae θ_i, where each θ_i is the conjunction of literals, where a literal can be either an atomic formula, or its negation, or an equality between two variables, or an inequality ($\neq$) between two variables; moreover, each disjunct contains at least one of the literals $x_i = x_j$ or $x_i \neq x_j$ for any two variables x_i, x_j (that is, it determines *an equality type*). Next, we distribute the existential quantifiers over the disjunction, and thus we have to show that each disjunct of the form $\exists \mathbf{x}\theta_i \wedge (\forall P')[(P' < P) \rightarrow \neg(\bigvee_{j=1}^m \exists \mathbf{x}\theta_j)]$ is first order. However, since only existential quantifiers occur in this disjunct and θ_i has a fixed equality type (in other words, the mapping from variables to constants is fixed up to renamings), the assertion concerning P above can be replaced by a first-order formula stating that P is a certain finite set and no proper subset of it satisfies $\bigvee_{j=1}^m \exists \mathbf{x}\theta_j$ (the latter statement can be expressed by an exponentially long first-order formula, ranging over subsets of the set of constants determined by the equality type). This completes the construction.

Example: We compute the circumscription of P in the formula

$$\exists x_1 \exists x_2 (R(x_1, x_2) \wedge P(x_1) \wedge P(x_2))$$

using the procedure described above. After bringing the quantifier-free part in complete disjunctive normal form and distributing the existential quantifiers over the disjunction, this formula is transformed to

$$\exists x_1 \exists x_2 (R(x_1, x_2) \wedge P(x_1) \wedge P(x_2) \wedge (x_1 = x_2))$$
$$\vee\ \exists x_1 \exists x_2 (R(x_1, x_2) \wedge P(x_1) \wedge P(x_2) \wedge (x_1 \neq x_2)).$$

The circumscription of P in the above formula is equivalent to

$$\exists x_1 (R(x_1, x_1) \wedge P(x_1) \wedge ((\forall y)(P(y) \leftrightarrow y = x_1)) \vee$$
$$[\exists x_1 \exists x_2 (R(x_1, x_2) \wedge P(x_1) \wedge P(x_2) \wedge (x_1 \neq x_2) \wedge$$
$$((\forall y)(P(y) \leftrightarrow (y = x_1 \vee y = x_2))) \wedge$$
$$(\neg R(x_1, x_1)) \wedge (\neg R(x_2, x_2)))]. \quad \square$$

We notice that computing a first-order sentence equivalent to the circumscription of P in an existential first-order formula $\phi(P)$ seems to increase the size of $\phi(P)$ exponentially, a phenomenon not observed in the other known cases of first-order circumscription studied in [Li85]. It would be interesting to determine whether this is inherent to existential first-order formulae, or a particular creation of our proof.

In the full paper we shall also prove that Theorem 1 can be extended in several directions: It holds for formulae containing not only relation symbols, but also function and constant symbols. Also, it holds for *circumscription with variables* (a more general variant). Finally, it is also true of *existential second-order formulae*, that is, second-order formulae whose second-order and first-order quantifiers are all existential.

3 Circumscription and Boundedness

The positive results for the existential formulae in the preceding section suggest that one should examine next the class of universal first-order formulae. Other properties of the circumscription of universal formulae have been studied before and it is known, for example, that this class of formulae behaves nicely with respect to the *satisfiability* of circumscription (cf. [BS85], [EMR85], [Li86]). As mentioned in the introduction, however, Lifschitz [Li85] observed that there are universal formulae (actually conjunctions of function-free Horn clauses) whose circumscription is *not* first-order expressible. In view of this, the best possible result one could hope for is a computationally useful characterization of the universal sentences that have a first-order circumscription.

In this section we establish a connection between the circumscription of a conjunction of function-free Horn clauses and the convergence of the corresponding logic program. More specifically, we show that the circumscription of a conjunction of Horn clauses is first-order if and only if the corresponding program is bounded. Boundedness is a property of logic programs that has been showed recently by Gaifman et al. [GMSV87] to be an *undecidable problem*. Thus, it is *not* possible to give a computationally useful characterization of which universal first-order formulae possess a first-order circumscription. In spite of these negative consequences, our result suggests that it may be possible to identify wide subclasses of universal formulae on which there are algorithms that detect when circumscription is first-order. In the case of logic programs, algorithms detecting boundedness on fairly natural collections of logic programs have been discovered in recent years by researchers in database theory and logic programming ([Io85], [Sa85], [Na86], [NS87], [CGKV88]).

Logic Programs. Recall that a *Horn clause* $\psi(P)$ (with respect to the n-ary predicate P) is an expression of the form:

$$\forall \mathbf{x} \forall \mathbf{z}(\chi(\mathbf{x}, \mathbf{z}, P) \rightarrow P(\mathbf{x})),$$

where $\chi(\mathbf{x}, \mathbf{z}, P)$ is a conjunction of (positive) atomic formulas involving P and other predicate symbols (these latter symbols are called *database relations*). Finally, a *logic program* $\phi(P)$ is a conjunction

$$\bigwedge_{i=1}^{l} \forall \mathbf{x} \forall \mathbf{z}_i(\chi_i(\mathbf{x}, \mathbf{z}_i, P) \rightarrow P(\mathbf{x}))$$

of Horn clauses.

Suppose that $\mathbf{A}$ is a structure (set of values for the database relations in $\phi(P)$). The *semantics of the program* $\phi(P)$ on $\mathbf{A}$ is the smallest n-ary relation P^∞ on $\mathbf{A}$ such that $\mathbf{A} \models \phi(P^\infty)$. The semantics P^∞ of the logic program can be alternatively viewed as *the least fixpoint* of a certain operator Θ on n-ary relations associated with $\phi(P)$. More precisely, notice first that the logic program

$$\bigwedge_{i=1}^{l} \forall \mathbf{x} \forall \mathbf{z}_i(\chi_i(\mathbf{x}, \mathbf{z}_i, P) \rightarrow P(\mathbf{x}))$$

is equivalent to

$$\forall \mathbf{x}[(\bigvee_{i=1}^{l} \exists \mathbf{z}_i \chi_i(\mathbf{x}, \mathbf{z}_i, P)) \rightarrow P(\mathbf{x})].$$

We define now the operator Θ from n-ary relations to n-ary relations as follows:

$$\Theta(P) = \{\mathbf{a} : \mathbf{A} \models \bigvee_{i=1}^{l} \exists \mathbf{z}_i \chi_i(\mathbf{a}, \mathbf{z}_i, P)\}.$$

In other words, Θ maps a relation P to the set of all those tuples that satisfy with $\mathbf{A}$ some χ_i. The operator Θ is *monotone*, that is

$$P \subseteq P' \rightarrow \Theta(P) \subseteq \Theta(P'),$$

and hence it has a least fixpoint, which can be easily seen to be equal to P^∞. The least fixpoint can also be obtained by the iteration

$$P^0 = \emptyset, \quad P^{m+1} = \Theta(P^m),$$

because $P^\infty = \bigcup_{m=1}^{\infty} P^m$. Intuitively, P^m is the relation resulting from at most m successive applications of the *rules* of the program, where we start by taking P to be empty. On finite structures this process converges after a finite number of steps, i.e for every finite structure $\mathbf{A}$ there is an integer m_0 such that $P^\infty = P^{m_0} = P^m$, for all $m \geq m_0$.

Example: Consider the logic program $\phi(P)$ which is the conjunction of the Horn clauses (omitting universal quantifiers)

$$(E(x, y) \rightarrow P(x, y)) \wedge ((P(x, z) \wedge P(z, y)) \rightarrow P(x, y)).$$

The associated operator Θ gives here

$$\Theta(P) = \{(x, y) : (E(x, y) \vee \exists z(P(x, z) \wedge P(z, y))\}$$

As a result, on any graph $\mathbf{G} = (V, E)$, the relation P^m consists of all pairs (x, y) such that there is a path of length m or less from x to y. It follows that P^∞ is equal to the *transitive closure* of E.

Lifschitz [Li85] noticed that the circumscription $\phi^*(P)$ of P in the logic program above asserts that P is the transitive closure of E, in other words the circumscription of P coincides with the semantics of the logic program. It turns out that this is a special case of the following:

Lemma 1. Let $\phi(P)$ be a logic program and let $\phi^*(P)$ be the circumscription of P in $\phi(P)$. Then, for any structure $\mathbf{A}$ and any relation S on $\mathbf{A}$

$$\mathbf{A} \models \phi^*(S) \text{ if and only if } S = P^\infty \quad \Box$$

A logic program $\phi(P)$ is *bounded* if there is a positive integer k such that on any structure $\mathbf{A}$ the logic program converges to its least fixpoint within k steps, i.e. $P^\infty = P^k$ on any $\mathbf{A}$. Notice that the logic program defining the transitive closure TC of E is unbounded Moreover, it is well known (cf [AU79]) that TC is not first-order definable This is not a coincidence; as a matter of fact, bounded programs are exactly those programs for which recursion can be eliminated:

Lemma 2 . A logic program $\phi(P)$ is bounded if and only if there is a first-order formula $\theta(\mathbf{x})$ which defines P^∞. $\Box$

The proof of Lemma 2 involves the compactness theorem of mathematical logic. We now have all the prerequisites to derive the main result of this section:

Theorem 2. The circumscription $\phi^*(P)$ of P in a logic program $\phi(P)$ is equivalent to a first-order sentence if and only if the logic program $\phi(P)$ is bounded.

Proof: Assume first that there is a first-order sentence $\vartheta(P)$ which is equivalent to the circumscription $\phi^*(P)$ on every structure $\mathbf{A}$. It follows from Lemma 1 that $\vartheta(P)$ *implicitly defines* P on every stucture, that is for any structure $\mathbf{A}$ and any n-ary relations S and S' on $\mathbf{A}$

$$\mathbf{A} \models (\vartheta(S) \wedge \vartheta(S')) \longrightarrow (S = S' = P^\infty)$$

The Beth definability theorem (cf [CK73]) implies then that P^∞ is *explicitly definable*, i.e there is a first-order formula $\theta(\mathbf{x})$ defining P^∞ on every structure $\mathbf{A}$ We apply now Lemma 2 to conclude that the logic program $\phi(P)$ is bounded.

Conversely, if there is an integer $k \geq 1$ such that $P^\infty = P^k$ on every structure, then the circumscription $\phi^*(P)$ is equivalent to the first-order sentence

$$\forall \mathbf{x}(P(\mathbf{x}) \longleftrightarrow \phi^k(\mathbf{x})),$$

where $\phi^k(\mathbf{x})$ is a first-order formula defining the k-th stage P^k of P^∞ on every structure. $\Box$

The question of when is a logic program bounded has been studied extensively in database theory The interest in this property is explained by Lemma 2, which reveals that bounded programs are exactly those logic programs for which recursion is not necessary Thus, testing a logic program for boundedness is a useful step in optimizing the program and obtaining efficient evaluation methods for it Several researchers, including [Io85], [Sa85], [Na86], [NS86], developed boundedness algorithms for fairly wide classes of logic programs. Gaifman, Mairson, Sagiv, and Vardi [GMSV87], however, showed that *no* such algorithm exists for the class of *all* logic programs, by establishing the following:

Theorem.([GMSV87]) The collection of bounded logic programs is a complete recursively enumerable set. □

As an immediate consequence of Theorem 2 and the above result, we have

Corollary 1. The collection of logic programs having a first-order circumscription is a complete recursively enumerable set. Consequently, it is an undecidable problem to tell whether or not, given a universal first-order formula, *its circumscription is expressible in first-order logic.* □

We should mention that Krishnaprasad [Kr88] showed that it is an undecidable problem to tell whether or not the circumscription of a first-order formula is expressible in first-order logic The formulae he constructed, however, involved function symbols and at least two alternations of quantifiers

4 Model Checking

Let ψ be a formula (first or second-order) The *model checking problem for* ψ is the following computational question (of obvious interest to AI): Given a finite structure, does it satisfy ψ? It is well-known that the model checking problem for first-order formulae can be carried out in logarithmic space (and thus in polynomial time) in the size of the given finite structure.

The inherent second-orderness of circumscription has yet another unpleasant side: in passing from a first-order formula to its circumscription, model-checking may become intractable Since circumscription is defined in terms of second-order logic, the complexity of model checking for $\phi^*(P)$ cannot surpass polynomial space. In fact, since the definition of circumscription uses only universal second-order quantifiers, the model checking problem for $\phi^*(P)$ is in coNP ([Fa74]). In the case of a logic program $\phi(P)$, the model checking for $\phi^*(P)$ is actually solvable in polynomial time (this follows from Lemma 1) In contrast to this, we give an example of a simple first-order formula, whose circumscription is *coNP-complete*, and thus most probably inherently intractable A similar phenomenon was also observed by Vardi [Va86] in passing from *physical databases* to *logical databases*

Example: We call an undirected graph *cubic* if all nodes have degree three (that is, exactly three edges incident upon each). Obviously, cubicity is a first-order property, that is, there is a first-order formula $\kappa(E)$ such that $\kappa(E)$ is satisfied by exactly the cubic graphs $\mathbf{G} = (V, E)$. A *circuit* of the graph is a closed path without repetitions of edges Call a circuit *long* if it contains at least twelve nodes. Finally, we call a graph *simple* if it is the disjoint union of long circuits. That is, simple graphs have all degrees two, and there are no circuits of length eleven or less in them. It is easy to see that simplicity can also be expressed by a first order formula $\sigma(E)$.

Let $\phi(E)$ be the formula $\kappa(E) \vee \sigma(E)$. It states the elementary fact that a graph $\mathbf{G} = (V, E)$ is either simple or cubic. Naturally, this is a very easy property to check. What is the circumscription $\phi^*(E)$, however? It proclaims that $\mathbf{G} = (V, E)$ is either cubic or simple, *and there is no proper subset E' of E such that $\mathbf{G}' = (V, E')$ is also either cubic or simple.* If $\mathbf{G}$ is simple, then clearly no proper subgraph (here we mean that edges, *but no nodes*, are deleted) can be simple. Similarly, if a connected graph is cubic, then no proper subgraph of it can be cubic *However, if a graph is cubic, it may or may not contain a simple subgraph*, and it is not clear how to tell those that do from those that do not (short of enumerating all subgraphs, an exponetially difficult task) Thus, it appears difficult to solve the model checking problem for $\phi^*(E)$ In fact, we shall next show that the problem is coNP-complete. □

Lemma 2. It is NP-complete to tell whether a cubic connected graph has a simple subgraph (on the same set of nodes).

Sketch of Proof: The proof is a variant of the reduction from 3SAT to the Hamilton circuit problem [PS82]. It turns out that the existence of a simple subgraph of the graph constructed there is equivalent to the existence of a Hamilton cycle Moreover, the graph can be made cubic by standard techniques □

Thus, we have established the following:

Theorem 4. There is a first order sentence, whose circumscription has a coNP-complete model checking problem. □

5 Open Problems

There are several open problems motivated from the results reported here The following is only a partial list:

1 Identify interesting classes of universal formulae on which either circumscription is first-order definable or there are efficient algorithms detecting which formulae in the class possess a first-order circumscription

2. From the results of section 3, it follows that it is an undecidable problem to determine if the cicumscription of a given first-order sentence is equivalent to a first-order sentence on *finite structures* This problem is, on the face of it, in Σ_2 (= the second level of the arithmetic hierarchy). Is it Σ_2-complete? Recall that a typical Σ_2-complete problem is the set of Gödel numbers of non-total recursive functions

3 Let C be the collection of all logic programs whose circumscription is equivalent to a first-order sentence on *finite structures* Is C a recursively enumerable set? This questions is closely related to the problem of whether or not Lemma 2 holds, when only finite structures are considered.

4. Assume that $\phi(P)$ is a universal first-order sentence. Is it true that the model checking for the cicumscription $\phi^*(P)$ is solvable in polynomial time? This is certainly true when $\phi(P)$ is in addition a logic program Notice also that the sentence $\phi(E)$ in section 4, whose circumscription has a coNP-complete model cheking problem, is actually equivalent to a universal-existential sentence

5. In view of the computational difficulties surrounding cirumscription, it is natural to raise again the question: is

there a different formalization of common-sense reasoning, which on the one hand is computationally more tractable than circumscription, and on the other retains most salient features of it ?

Acknowledgments. We are grateful to Vladimir Lifschitz for several useful comments and suggestions on an earlier version of this paper. The research of the second author was partially supported by a NSF grant

6 References

- [AU79] Aho, A V., Ullman, J D.: Universality of data retrieval languages *Proc. 6th ACM Symposium on Priciples of Programming Languages*, 1979, pp 110-117.
- [BS85] Bossu, G , Siegel, P.: Saturation, non-monotonic reasoning and the closed world assumption.*Artificial Intelligence* 25 (1985), pp. 13-63
- [CGKV88] Cosmadakis, S.S , Gaifman, H , Kanellakis P.C , Vardi, M.Y : Decidable optimization problems for database logic programs *Proc 20th ACM Symposium on Theory of Computing*, Chicago, 1988, pp 477-490.
- [CK73] Chang, C C , Keisler, H J : *Model Theory*, North-Holland, 1973
- [Da80] Davis, M : The mathematics of non-monotonic reasoning *Artificial Intelligence* 13 (1980), pp 73-80
- [EMR85] Etherington, D , Mercer, R , Reiter, R : On the adequacy of predicate circumscription for closed-world reasoning *Computational Intelligence* 1 (1985), pp. 11-15
- [Fa74] Fagin, R : Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of Computations*, ed by R Karp, SIAM-AMS Proc 7 (1974), pp 43-74
- [GMSV87] Gaifman, H , Mairson, H , Sagiv, Y , Vardi, M Y : Undecidable optimization problems for database logic programs *Proc 2nd IEEE Symposium on Logic in Computer Science*, 1987, pp 106-115
- [GN87] Genesereth, M., Nilsson, N.J.: *Logical Foundations of Artificial Intelligence*, Morgan-Kaufmann, 1987
- [Io85] Ioannides, Y E : A time bound on the materialization of some recursively defined views *Proc. 11th International Conference on Very Large Data Bases*, 1985, pp 219-226
- [Kr88] Krishnaprasad, T : On the computability of circumscription *Information Processing Letters* Volume 27, Number 5 (1988), pp. 237-243
- [Li85] Lifschitz, V.: Computing circumscription *Proc. 9th International Joint Conference on Artificial Intelligence* 1985, pp 121-127.
- [Li86] Lifschitz, V.: On the satisfiability of circumscription *Artificial Intelligence* 28 (1986), pp. 17-27
- [McC80] McCarthy, J : Circumscription - a form of non-monotonic reasoning *Artificial Intelligence* 13 (1980), pp 27-39
- [McC86] McCarthy, J.: Applications of circumscription in formalizing common sense knowledge *Artificial Intelligence* 28 (1986), pp 89-116
- [Na86] Naughton, J.F : Data independent recursion in deductive databses *Proc. 5th ACM Symposium on Principles of Database Systems*, 1986, pp. 267-279.
- [NS87] Naughton, J.F., Sagiv, Y : A decidable class of bounded recursions.*Proc 6th ACM Symposium on Principles of Database Systems*, 1987, pp. 227-236.
- [PS82] Papadimitriou, C.H., Steiglitz, K : *Combinatorial Optimization*, Prentice-Hall, 1982.
- [Sa85] Sagiv, Y.: On computing restricted projections of representative instances.*Proc. 4th ACM Symposium on Principles of Database Systems*, 1985, pp. 171-180.
- [Sc86] Schlipf, J S.: How uncomputable is general circumscription *Proc. 1st IEEE Conference on Logic in Computer Science*, 1986, pp 92-95.
- [Va86] Vardi, M.Y.: Quering Logical Databases *J Computer and System Sciences* 33, No. 2 (1986), pp. 142-160
- [Va88] Vardi, M Y : Decidability and undecidability results for boundedness of linear recursive queries:*Proc. 7th ACM Symposium on Principles of Database Systems*, 1988