Efficient Reasoning using the Local Closed-World Assumption

Patrick Doherty[1*], Witold Łukaszewicz[2], and Andrzej Szałas[2]

[1] Department of Computer and Information Science,
Linköping University,
S-581 83 Linköping, Sweden,
patdo@ida.liu.se,
[2] Institute of Informatics,
Warsaw University,
ul. Banacha 2, 02-097 Warsaw, Poland,
{witlu,szalas}@mimuw.edu.pl

Abstract. We present a sound and complete, tractable inference method for reasoning with localized closed world assumptions (LCWA's) which can be used in applications where a reasoning or planning agent can not assume complete information about planning or reasoning states. This *Open World Assumption* is generally necessary in most realistic robotics applications. The inference procedure subsumes that described in Etzioni et al [9], and others. In addition, it provides a great deal more expressivity, permitting limited use of negation and disjunction in the representation of LCWA's, while still retaining tractability. The approach is based on the use of circumscription and quantifier elimination techniques and inference is viewed as querying a deductive database. Both the preprocessing of the database using circumscription and quantifier elimination, and the inference method itself, have polynomial time and space complexity.
In Processdings of the 9th International Conference on Artificial Intelligence: Mehtodology, Systems, Applications, 2000, (AIMSA 2000).

1 Introduction

Traditionally, classical reasoning and planning techniques have been developed for environments in which the reasoning agent is assumed to have complete information about the world in which it is embedded and the only changes to the world are the effects which result from the agent's invocation of actions. Under this assumption, an efficient means of representing negative information about the world in each planning or reasoning state is to apply the *Closed World Assumption* (CWA) [1, 16]. In this case, information about the world, absent in a state, is assumed to be false.

* The authors are supported in part by a basic research grant from the Wallenberg Foundation, Sweden.

In many realistic applications, in particular robotics applications, the assumption of complete information is not feasible and the CWA can not be used. For example, an unmanned aerial vehicle flying over a region can not have a complete model of the region. New objects are continually sensed or encountered and agents other than the UAV agent cause change in the region. In applications such as this, an *Open World Assumption* (OWA), where information not known by the agent is assumed to be unknown, is the ontologically right choice to make, but complicates both the representational and implementational aspects associated with inference mechanisms and the use of negative information.

The CWA and the OWA represent two extremes. Quite often, a reasoning agent has information which permits the application of the CWA *locally.* If the UAV agent has a camera sensor, the agent can assume complete information about objects in the focus of attention (FOA) of the camera; for example, the only cars in the FOA are those identified by the image processing module.

The research issue then, is to find maximally expressive, but tractable inference mechanisms for local closed world reasoning which can be integrated with deliberative components, such as planning algorithms, used in applications where the OWA applies. An additional issue is to be able to dynamically modify the degree of closed-worldness relative to the dynamics of the application at hand.

We approach the problem as follows. The starting point is the approach to LCWA described in [9], where the authors present a sound, but incomplete, tractable algorithm for LCWA intended for use in the XII Planner. Briefly, their approach works as follows: Assume an actual world w which can be represented by a complete logical theory. Since the reasoning agent only has incomplete information about that world, but that information is assumed correct, the agent's knowledge can be represented as a set of possible worlds S, where $w \in S$. For reasons of tractability, the approach approximates S by representing it as a set of ground literals, M, where negative information about w known to the agent is represented explicitly. M can be viewed as the agent's knowledge database. Localized closure information is represented in another database, $\mathcal{L}$, as a set of formulas restricted to be conjunctions of literals (not necessarily grounded). For example, $M = \{parent.dir(\mathsf{ecai.tex}, \mathsf{/ecai00}), size(\mathsf{kr.tex}, \mathsf{100})\}$, $\mathcal{L} = \{LCW(parent.dir(f, \mathsf{/ecai00}))\}$. Although a reasoning agent could not infer that it knows about all the files in all directories and their sizes, it can infer that it knows about all the files in the directory ecai00. In [9], the authors describe an algorithm which encodes a sound, but incomplete inference relation, $M, \mathcal{L} \models_{\mathcal{E}} \alpha$, where given M and $\mathcal{L}$, they can determine whether a conjunction of positive ground literals, α, is inferable under partial closure of the theory. Since both $\mathcal{L}$ and α are restricted to be positive conjunctive formulas, the algorithm and its efficiency are based on the use of matching conjunctive queries against a conjunctive database. Note that due to the OWA, for a specific query α, the algorithm may return true, false, or unknown. In the following, we use QLCW to refer to the query language of which α would be an instance.

We substantially extend the approach of [9] by:

- providing a semantics for the case where LCW constraints in $\mathcal{L}$ and queries in QLCW are expressed by arbitrary first-order formulas. The semantics is based on the use of circumscription. The new semantics and the one given in [9] agree on the special case where conjunctions of positive literals are used in $\mathcal{L}$ and QLCW.
- isolating a more expressive language for LCW constraints in $\mathcal{L}$ which subsumes that used in [9], permits limited use of negation and disjunction, and still retains tractability.
- providing a sound **and** complete, tractable deduction method for the more expressive language. Observe that in [9] completeness is not guaranteed even in the case of a language with conjunctions of positive literals only.

Our approach to the problem is based on the use of circumscription to minimize formulas in $\mathcal{L}$ in the context of the theory $\mathcal{L} \wedge M$. Using quantifier elimination techniques, the original circumscribed theory can be reduced to a 1st-order or fixpoint formula. Viewing the reduced theory as a database query, inference relative to M can be viewed as a query to a database. Restricting the expressivity of $\mathcal{L}$ to what we call semi-Horn formulas, M to a conjunction of positive and negative ground literals, and queries in QLCW to semi-Horn formulas, we can show that both the theory reduction technique and querying technique remain tractable and safe. Tractability means that the method allows for efficient (PTIME) computations. Safety means that no inconsistencies are introduced by the method no matter what logical dependencies are used in $\mathcal{L}$.

Note that by first providing a general framework and semantics for structuring the problem in a classical setting and then isolating tractable combinations of fragments of the languages used in M, $\mathcal{L}$, and QLCW, we provide a methodology for generalization of the technique based on the use of results from the knowledge representation and deductive database communities.

2 Preliminaries

We deal with an ordinary first-order language with equality, L_1, over a fixed alphabet A without function constants. By L_2 we denote the second-order language based on an alphabet whose symbols are those of A, together with a denumerable set of n-ary predicate variables (for each $n \geq 0$). These will be denoted by the letters Φ and Ψ, possibly with subscripts and/or primes.

In the sequel, we shall use second-order circumscription. Our definition follows [13].

Definition 1. Let $\overline{P}$ be a tuple of distinct predicate constants, $\overline{S}$ be a tuple of predicate constants disjoint with $\overline{P}$, and let $T(\overline{P}, \overline{S})$ be a finite theory in a language L_1. The *second-order circumscription of* $\overline{P}$ *in* $T(\overline{P}, \overline{S})$ *with variable* $\overline{S}$, written $CIRC(T(\overline{P}, \overline{S}); \overline{P}; \overline{S})$, is the sentence (in the language L_2)

$$T(\overline{P}, \overline{S}) \wedge \forall \overline{\Phi} \overline{\Psi} [T(\overline{\Phi}, \overline{\Psi}) \wedge [\overline{\Phi} \leq \overline{P}] \supset [\overline{P} \leq \overline{\Phi}]],$$

where $\overline{\Phi}$ and $\overline{\Psi}$ are tuples of predicate variables similar to $\overline{P}$ and $\overline{S}$, respectively[1], and $\overline{\Phi} \leq \overline{P}$ (resp. $\overline{P} \leq \overline{\Phi}$) stands for $\bigwedge_{i=1}^{n} [\forall \overline{x}.\Phi_i(\overline{x}) \supset P_i(\overline{x})]$ (resp. $\bigwedge_{i=1}^{n} [\forall \overline{x}.P_i(\overline{x}) \supset \Phi_i(\overline{x})]$). ■

In the following, we shall often write $CIRC(T; \overline{P}; \overline{S})$ instead of the formula $CIRC(T(\overline{P}, \overline{S}); \overline{P}; \overline{S})$.

Let us now quote the fixpoint theorem formulated and proved in [15] for second-order quantifier elimination.

Theorem 1. Let P be a predicate variable, and $\Psi'(P)$,$\Psi(\neg P)$ be formulas without second-order quantification. Let $\Phi(P)$ be positive w.r.t. P, $\Psi(\neg P)$ be negative w.r.t. P and $\Psi'(P)$ be positive w.r.t. P, then

$$\exists P \forall \bar{y} [\Phi(P) \supset P(\bar{y})] \wedge [\Psi(\neg P)] \ \equiv \ \Psi[P \leftarrow \mu P(\bar{y}).\Phi(P)], \tag{1}$$

and

$$\exists P \forall \bar{y} [P(\bar{y}) \supset \Phi(P)] \wedge [\Psi'(P)] \ \equiv \ \Psi'[P \leftarrow \nu P(\bar{y}).\Phi(P)], \tag{2}$$

where the above substitutions exchange the variables bound by fixpoint operators by the corresponding actual variables of the substituted predicate. (1), ((2)) is used to minimize (maximize) P. ■

The definition of semi-Horn formulas, for which Theorem 1 is applicable, has been introduced in [5]. In what follows we shall consider a restricted version of semi-Horn formulas, where the *recursive* part of the semi-Horn formula is restricted as to the use of universal quantifiers.

Definition 2. By a *semi-Horn formula (w.r.t. Q)* we understand a conjunction of formulas of the form

$$[\Phi(\bar{x}) \supset Q(\bar{x})] \wedge \Psi(\neg Q), \tag{3}$$

and

$$[Q(\bar{x}) \supset \Phi(\bar{x})] \wedge \Psi(Q), \tag{4}$$

where $\Phi(\bar{x})$ is any classical first-order formula positive w.r.t. Q and $\Psi(\neg Q)$ ($\Psi(Q)$) is any first-order formula negative (positive) w.r.t. Q. Formula $\Phi(\bar{x}) \supset Q(\bar{x})$ ($Q(\bar{x}) \supset \Phi(\bar{x})$) is called the *recursive part* of (3) ((4)) and $\Psi(\neg Q)$ ($\Psi(Q)$) is called the *negative* (*positive*) *part* of (3) ((4)).

By a *semi-Horn formula* we understand a semi-Horn formula w.r.t. all predicate symbols occurring in the formula. ■

---

[1] A tuple of predicate expressions $\overline{X}$ is said to be similar to a tuple of predicate constants $\overline{Y}$ iff $\overline{X} = (X_1, \ldots, X_n)$, $\overline{Y} = (Y_1, \ldots, Y_n)$ and, for all $1 \leq i \leq n$, X_i and Y_i are of the same arity.

3 Representing an Agent's Knowledge

Suppose W is a complete logical theory formalizing what is true in an actual world state w. Suppose also, that T is a finite first-order theory, i.e. a finite set of sentences from L_1, formalizing an agent's *knowledge* about w. Following [9], we say that an agent has *local closed-world information* w.r.t. a formula α and T iff

$$T \models \alpha\theta \quad \text{or} \quad T \models \neg\alpha\theta \quad \text{for each ground substitution } \theta.$$

It is assumed that any knowledge the agent infers from T is correct in the actual world w. Since T provides only incomplete information about w, not all facts about w are known to the agent. In other words, only some of the information is locally closed relative to T, other information is unknown.

Following [9], we approximate an agent's knowledge about T by a pair $M, \mathcal{L}$, where M is a finite set of positive or negative ground literals and $\mathcal{L}$ is a set of first-order formulas, representing local closed-world assumptions. We assume that if T formalizes the agent's knowledge about the world, then for each formula α

$$M \models \alpha \text{ implies } T \models \alpha.$$

Let $c_1, \ldots, c_n$ be all the constants from the alphabet under consideration. We write $DCA(M)$ to denote the *domain closure* axiom for a theory M. This is the formula

$$\forall x. \bigvee_{i=1}^{n} x = c_i.$$

We write $UNA(M)$ to denote the *unique name assumption* axiom for a theory M. This is the formula

$$\bigwedge_{1 \le i < j \le n} c_i \neq c_j.$$

We write $M, \mathcal{L} \models \alpha$ to denote that a formula α follows from a pair $M, \mathcal{L}$. This notion is defined as follows.

Definition 3. Let $M, \mathcal{L}$ be a finite set of ground literals and a set of formulas representing closed-world information, respectively. Suppose that $\mathcal{L}$ consists of formulas $\beta_1, \ldots, \beta_n$. Let $\overline{R} = R_1, \ldots, R_n$ be a set of new predicates symbols similar to $\beta_1, \ldots, \beta_n$.[2] By an *LCW-based extension of M*, denoted by M', we shall understand this to be the theory consisting of formulas of M, augmented by:

- $DCA(M)$ and $UNA(M)$
- the set of formulas $\forall \overline{x}. R_i(\overline{x}) \equiv \beta_i \ (i = 1, \ldots n)$. ■

The following definition provides us with the semantics of LCW as understood in this paper.

[2] A predicate symbol P is similar to a formula α iff the arity of P is equal to the number of free variables of α.

Definition 4. Let $\overline{S}$ be the set of all predicate symbols occurring in $\beta_1, \ldots, \beta_n$. Then

$$M, \mathcal{L} \models \alpha \text{ iff } CIRC(M'; \overline{R}; \overline{S}) \models \alpha,$$

where $\overline{R} = (R_1, \ldots, R_n)$. ■

Note that definition 4 provides the general case and semantics for reasoning under the LCWA. The rest of the paper considers restrictions on M, $\mathcal{L}$ and QLCW which make reasoning under the LCWA tractable.

In what follows we divide M' into three parts:

- a *positive part*, denoted by M_+, consisting of positive literals of M; the positive part is intended to gather positive information directly included in the database M
- a *negative part*, denoted by M_-, consisting of negative literals of M; the negative part is intended to gather negative information directly included in the database M
- an *LCW part*, denoted by M_c, consisting of equivalences $\forall \overline{x}.R_i(\overline{x}) \equiv \beta_i$ $(i = 1, \ldots n)$ introduced in Definition 3.

Observe that M_+ is just an extensional database as understood in the field of deductive databases (see, e.g. [1, 8]). Also M_- can be easily treated as a part of an extensional database. Now (deductive) queries are represented by M_c embedded in a tractable query language like fixpoint calculus or classical first-order logic (see e.g. [1]). Thus, whenever LCW is polynomially reducible to fixpoint or classical formulas, one has a tractable reasoning mechanism.

In what follows we often call $M_+ \cup M_-$ simply a database.

4 The Main Result

The following theorem provides us with a sufficient condition which guarantees that second-order quantifiers can be eliminated from $CIRC(M'; \overline{R}; \overline{S})$ using the fixpoint theorem (Theorem 1) and some syntactic transformations applied in the DLS algorithm [6].

The main result of this paper, formulated below, shows that the second-order formula resulting from circumscription can be reduced to a fixpoint formula. Thus the complexity of reasoning is polynomial in the size of M. This follows from the fact that the database part of M is not affected by the quantifier elimination process. Only LCW constraints can, in some cases, introduce additional complexity. However the size of the resulting formula is, in the worst case, not greater than $m + O(n^2)$, where n is the size of LCW constraints together with the query and m is the size of the database.

In the proof of the theorem we use second-order quantifier elimination (For surveys of approaches to second-order quantifier elimination consult [6, 14]). Because of the space limitations, the proof is not included in the current paper, but is available from the authors.

Theorem 2. Let $CIRC(M'; \overline{R}; \overline{S})$ be defined as in Section 3. If M consists of literals and LCW constraints in $\mathcal{L}$ are defined by means of semi-Horn formulas, then the following conditions hold:

- second-order quantifiers can be eliminated from $CIRC(M'; \overline{R}; \overline{S})$;
- if the size of the database M is m and the size of M_c together with the query is n then, in the worst case, the resulting formula has size $m + O(n^2)$. Moreover M is not affected by the quantifier elimination process.

The second-order quantifier elimination technique applied in the proof of Theorem 2 is based on Theorem 1 and provides us also with definitions of the eliminated predicates. As in [8], this feature is crucial for the approach we present in this paper. More precisely (for details see e.g. [8]):

- in the case of formulas of the form (1), one gets an explicit definition of the least relation P satisfying the first-order part of (1); and
- in the case of formulas of the form (2), one gets an explicit definition of the greatest relation P satisfying the first-order part of (2).

Observe that Theorem 2 can still be generalized using techniques of [6–8, 14].

Corollary 1. Consider a relational (or deductive) database in which the query language QLCW is the classical first-order logic or monotone fixpoint calculus[3]. If M consists of literals and the LCW constraints in $\mathcal{L}$ are defined by means of semi-Horn formulas, then:

- the time complexity of the quantifier elimination algorithm is polynomial in the size of the input query;
- the formula resulting from the quantifier elimination process is a monotone fixpoint formula, thus time and space data complexity of querying the database is polynomial in the size of the database;
- if all Φ's occurring in recursive parts of semi Horn formulas defined in definition 2 (i.e. in formulas of the form $\Phi(\bar{x}) \supset Q(\bar{x})$ and $Q(\bar{x}) \supset \Phi(\bar{x})$) do not contain Q's then the formula resulting from the quantifier elimination process is a classical first-order formula. Thus assuming that the query language is restricted to the classical first-order logic one obtains polynomial time data complexity and polylogarithmic space data complexity [1, 12].

Proof. The first item easily follows from the results provided in [5, 8] and from the proof of Theorem 2.

The second item just quotes results well-known from deductive databases (see e.g. [1, 12]).

The last item follows from the fact that for such formulas the Ackermann Lemma [2] is applicable - see also [6].

[3] I.e. calculus in which fixpoint are defined on monotone formulas - see e.g. [1].

5 The LCW Algorithm

Theorem 2 together with results in [8] provides us with a complete and tractable algorithm for deduction from a database M and LCW database $\mathcal{L}$, assuming that formulas in $\mathcal{L}$ and α are formulated as semi-Horn formulas and M consists of literals. An un-optimized abstract version of the algorithm is shown below:

Function LCWQuery$(\alpha, M, \mathcal{L})$: 3-boolean
  $M_{\mathcal{L}} := M \cup \mathit{Reduce}(\mathcal{L})$
  if $\mathit{RQuery}(M_{\mathcal{L}}, \alpha) \neq \emptyset$ **then return** T
  else if $\mathit{RQuery}(M_{\mathcal{L}}, \neg\alpha) \neq \emptyset$ **then return** F
  else return U;
end.

Reduce() is the quantifier elimination technique described in [8] extended with the result from Theorem 2. It is assumed that *Reduce*() provides us with definitions of the predicates eliminated from $CIRC(M'; \bar{R}, \bar{S})$.

RQuery() is based on [12] and returns a set of tuples satisfying α. However, CWA is not assumed by *RQuery*().

6 Related Work

In this section, we show that our approach subsumes the approaches proposed in [9], [3] and [10].

In [9] it is assumed that the LCW database, $\mathcal{L}$, consists of formulas that are conjunctions of atoms. We write $M, \mathcal{L} \models_{\mathcal{E}} \alpha$ to denote that a formula α follows from a pair $M, \mathcal{L}$ in Etzioni *et al.* [9] approach. The following theorem holds.

Theorem 3. For all M and $\mathcal{L}$

$$M, \mathcal{L} \models_{\mathcal{E}} \alpha \text{ implies } \mathcal{M}, \mathcal{L} \models \alpha. \blacksquare$$

Similarly, the ψ-forms considered in [3] are simply expressible in the language we deal with. Moreover, the semantics of both approaches is equivalent when restricted to the ψ-forms only.

In fact, the approach presented in [3] is subsumed by the one provided in [10]. In [10] Horn clauses, with additional built-in predicates, are used to express LCW constraints. These are easily expressible in our approach as we deal with semi-Horn formulas that are substantially more expressive than Horn clauses.

Note that the subsumption results are related to reasoning in a static state under the LCWA and not to sequences of dynamic states where updating the LCWA database is an additional issue considered in both [9] and [3].

7 Example

Example 1. The following example demonstrates the versatility of the approach by representing the UAV example in section 1. There are four cars with different signatures based on color. The UAV's focus of attention (FOA) is region r3. In $\mathcal{L}$, we assume complete information about the $ContainedIn()$ relation by minimizing it (6), and the $In()$ relation by maximizing it (5). (7) encodes the following LCWA by maximizing the relation $See()$:

After sensing region r3 *with a camera, we want to assume that we have seen* ***all*** *moving vehicles in the FOA (*r3*) except for those with signature* gray.

In querying the database using the LCWQuery algorithm, we can infer that $See(\mathsf{c1}, \mathsf{r3})$ holds, but it is unknown whether $See(\mathsf{c2}, \mathsf{r3})$, due to its signature; unknown whether $See(\mathsf{c3}, \mathsf{r3})$, because it is unknown whether it is moving; and unknown whether $See(\mathsf{c4}, \mathsf{r3})$ because it is not in the FOA. Note that the latter queries return unknown and not false due to the incompleteness of the database. In fact, other sensors may contribute to whether the unknown vehicles are seen.

$$\begin{aligned} \mathcal{L} = \{ In(x, r') \supset \neg(In(x,r) \wedge ContainedIn(r, r')), \quad (5) \\ ContainedIn(r, r'), \quad (6) \\ See(x, \mathsf{r3}) \supset \neg(InFOA(\mathsf{r3}) \wedge In(x, \mathsf{r3}) \wedge Sig(x, s) \wedge \\ s \neq \mathsf{gray} \wedge Moving(x))\} \quad (7) \end{aligned}$$

$$\begin{aligned} M = \{ In(\mathsf{c}_1, \mathsf{r}_1), In(\mathsf{c}_2, \mathsf{r}_2), In(\mathsf{c}_3, \mathsf{r}_1), In(\mathsf{c}_4, \mathsf{r}_4), \\ Moving(\mathsf{c}_1), Moving(\mathsf{c}_2), Moving(\mathsf{c}_4), \\ sig(\mathsf{c}_1, \mathsf{blue}), sig(\mathsf{c}_2, \mathsf{gray}), sig(\mathsf{c}_3, \mathsf{green}), sig(\mathsf{c}_4, \mathsf{yellow}), \\ ContainedIn(\mathsf{r1}, \mathsf{r3}), ContainedIn(\mathsf{r2}, \mathsf{r3}), InFOA(\mathsf{r3})\} \end{aligned}$$

In order to understand why the constraints for $In()$ and $See()$ in $\mathcal{L}$ are represented in the manner above, it is important to observe that the relations we want to *minimize* or *maximize* are in fact relations that are varied in the circumscriptive definition used for LCWA. Consequently, the minimization and maximization are achieved indirectly.

Another interesting observation is that the query generated by the quantifier elimination procedure results in a fixpoint formula due to the recursive definition of $In()$.

8 Conclusions

We have extended and subsumed the LCW querying techniques described in [9], [3, 4], and [10] and presented a tractable algorithm. The technique is based on the use of circumscription and results from the deductive database community and is consequently amenable to generalization. We have demonstrated the versatility of the approach by encoding a relatively complex UAV sensing scenario. We have not yet dealt with the LCW update problem associated with the query mechanism's integration with other planning and state sequential reasoning techniques considered in the other approaches, but are currently pursuing the problem.

References

1. S. Abiteboul, R. Hull, V. Vianu. *Foundations of Databases*, Addison-Wesley Pub. Co., 1996.
2. W. Ackermann. *Untersuchungen über das Eliminationsproblem der mathematischen Logik*, Mathematische Annalen, 110, 390-413, 1935.
3. T. Babaian, J. G. Schmolze. *PSIPLAN: Planning with ψ-forms over Partially Closed Worlds*, Technical Report.
4. T. Babaian, J: G. Schmolze. *PSIPLAN: Open World Planning with ψ-forms*, Proceedings of the 5th Int'l Conference on Artificial Intelligence and Planning (AIPS 2000), 2000.
5. P. Doherty, W. Łukaszewicz, A. Szałas. *A Reduction Result for Circumscribed Semi-Horn Formulas*, Fundamenta Informaticae, 28, 3-4, 261-271, 1996.
6. P. Doherty, W. Łukaszewicz, A. Szałas. *Computing Circumscription Revisited. A Reduction Algorithm.*, Journal of Automated Reasoning, 18, 3, 297-336, 1997.
7. P. Doherty, W. Łukaszewicz, A. Szałas. *General Domain Circumscription and its Effective Reductions*, Fundamenta Informaticae, 36, 1, 23-55, 1998.
8. P. Doherty, W. Łukaszewicz, A. Szałas. *Declarative PTIME Queries for Relational Databases using Quantifier Elimination*, Journal of Logic and Computation, 9, 5, 739-761, 1999.
9. O. Etzioni, K. Golden, D. S. Weld. *Sound and Efficient Closed-World Reasoning for Planning.* Artificial Intelligence, 89, 113-148, 1997.
10. M. Friedman, D. Weld. *Efficiently Executing Information Gathering Plans*, IJCAI-97.
11. J. Gustafsson. *An Implementation and Optimization of an Algorithm for Reducing Formulae in Second-Order Logic*, Technical Report LiTH-Mat-Ex-96-04, Linköping University, 1996.
12. N. Immerman. *Relational Queries Computable in Polynomial Time*, Information and Control, **68**, 86-104, 1986.
13. W. Łukaszewicz. (1990) *Non-Monotonic Reasoning - Formalization of Commonsense Reasoning*, Ellis Horwood Series in Artificial Intelligence. Ellis Horwood, 1990.
14. A. Nonnengart, H. J. Ohlbach, A. Szałas. *Elimination of Predicate Quantifiers*, In: *Logic, Language and Reasoning. Essays in Honor of Dov Gabbay, Part I*, H.J. Ohlbach and U. Reyle (eds.), Kluwer, 159-181, 1999.
15. A. Nonnengart, A. Szałas. *A Fixpoint Approach to Second-Order Quantifier Elimination with Applications to Correspondence Theory*, In: *Logic at Work. Essays Dedicated to the Memory of Helena Rasiowa*, E. Orlowska (ed.), Physica Verlag, 89-108, 1998.
16. R. Reiter. *On Closed World Databases*, in: *Logic and Databases*, H. Gallaire and J. Minker (eds.), Plenum Press, New York, 55-76, 1978.