The Complexity of Model Checking for Propositional Default Logics

Paolo Liberatore and **Marco Schaerf**[1]

Abstract. Default logic is one of the most widely used formalisms to formalize commonsense reasoning. In this paper we analyze the complexity of deciding whether a propositional interpretation is a model of a default theory for some of the variants of default logic presented in the literature. We prove that all the analyzed variants have the same complexity and that this problem is in general Σ_2^p complete, while it is coNP complete under some restrictions on the form of the defaults.

1 INTRODUCTION

Among all the formalisms proposed to model commonsense reasoning, default logic [15] is one of the most successful ones. Many aspects of default logic have been analyzed in the literature: semantics, algorithms, complexity, relationship with other formalisms, and so on.

In this paper we focus on a specific usage of default logic, that is, on using it as a tool to represent information in the form of sets of models. Any knowledge representation formalism is a formal tool to represent information about the world. Here we focus on default logic as a way to represent sets of models and, in particular, on the computational complexity of deciding the following problem:

> Given a default theory $\langle D, W\rangle$ and an interpretation M, decide if there exists an extension E of $\langle D, W\rangle$ such that $M \models E$.

This problem is better known as model checking. There are several reasons why model checking is of interest in AI. First of all, as convincingly suggested by Levesque [13] and advocated by Halpern and Vardi in [11], model-based representations are considered a viable alternative to the standard approach of representing knowledge in terms of formulae. In model-based representations the basic computational task is model checking, not inference. In this setting it is also very important to study the computational complexity of model checking.

While the computational complexity of inference and model checking are related, there is no way to automatically derive the results for model checking from those already known for inference.

The computational complexity of model checking is also strictly related to another computational aspect of knowledge representation formalisms: their *representational compactness*. Some recent papers [5, 8, 4] have shown that the compactness of a knowledge representation formalism is strictly related to the complexity of model checking.

Due to the lack of space not all proofs are presented. We present those that are, in our opinion, the most interesting ones and sketch out or completely omit all the others.

2 DEFINITIONS

Default logic has been defined by Reiter [15]. In default logic knowledge about the world is divided into two parts, representing certain knowledge and defeasible rules, respectively. For propositional default logic, certain knowledge (denoted with W) is a set of propositional formulae, while defeasible rules (denoted with D) are a collection of special inference rules called *defaults*. A default is a rule of the form

$$\frac{\alpha : \beta}{\gamma}$$

where α, β, and γ are propositional formulas. α is called the *prerequisite* of the default, β is called the *justifications*, and γ is the *consequence*.

Default rules of the form $\frac{\alpha:\beta}{\beta}$ are called *normal* (i.e. the justification is equal to the consequence of the default). Defaults of the form $\frac{\alpha:\beta\wedge\gamma}{\gamma}$ are called semi-normal. If a default has no precondition (i.e. it has the form $\frac{:\beta}{\gamma}$) it is called prerequisite-free.

A default theory is a pair $\langle D, W\rangle$, where D and W are as above. The semantics of a default theory $\langle D, W\rangle$ is based on the notion of *extension*, which is a possible state of the world according to the knowledge base. Formally, an extension is a fixpoint of the operator Γ defined as follows. Let A be a set of propositional formulas; $\Gamma(A)$ is the smallest set such that:

1. $W \subseteq \Gamma(A)$;
2. if $\Gamma(A) \models \alpha$ then $\alpha \in \Gamma(A)$;
3. if $\frac{\alpha:\beta}{\gamma} \in D$, $\alpha \in \Gamma(A)$, and $\neg\beta \notin A$ then $\gamma \in \Gamma(A)$.

A set of formulae E is an extension of $\langle D, W\rangle$ iff $E = \Gamma(E)$. Note that an extension is a deductively closed set of formulae. We say that a default $\frac{\alpha:\beta}{\gamma}$ is applicable w.r.t. a set of formulas A if and only if $A \models \alpha$ and $A \not\models \neg\beta$.

Given a set of defaults D, we denote with $CONS(D)$ the set of consequences of the defaults in D, that is $CONS(D) = \{\gamma \mid (\frac{\alpha:\beta}{\gamma}) \in D\}$. Each extension E of $\langle D, W\rangle$ is identified by a subset of D, called the set of *generating defaults* of E, defined as: $GD(E, \langle D, W\rangle) = \{\frac{\alpha:\beta}{\gamma} \in D \mid \alpha \in E \text{ and } \neg\beta \notin E\}$.

[1] Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, Via Salaria 113, I-00198, Roma, Italy {liberatore|schaerf}@dis.uniroma1.it

ECAI 98. *13th European Conference on Artificial Intelligence*
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

The set $GD(E, \langle D, W\rangle)$ of generating defaults has the property (see [15]) that each extension E of $\langle D, W\rangle$ is the deductive closure of

$$W \cup CONS(GD(E, \langle D, W\rangle))$$

The set of generating defaults gives a compact representation of an extension of a default theory, which by definition is a deductively closed set of formulae, hence infinite.

Many variants of default logic have been proposed in the literature. In this paper we also take into account those proposed by Lukaszewicz [14], Brewka [1] and Delgrande, Schaub and Jackson [6]. For more details on these systems we refer the reader to the original papers.

There are two forms of inference w.r.t. a default theory that have been defined in the literature: skeptical and credulous. In this paper we only concentrate on the first form because it is the only one that can be characterized in terms of sets of models. We say that a formula α is a skeptical consequence of a default theory $\langle D, W\rangle$, denoted as $\langle D, W\rangle \models \alpha$, if for all extensions E of $\langle D, W\rangle$ we have that $\alpha \in E$. There is a semantic counterpart to this definition: The set of models of a default theory is the union of the set of models of all the extensions. We can now define the problem of model checking: An interpretation M is a model of a default theory $\langle D, W\rangle$ if and only if M is a model of at least one extension of the theory. If this is the case, we write $M \models \langle D, W\rangle$. Notice that the above definition corresponds to saying that $\langle D, W\rangle \models \alpha$ if and only if for all models M of $\langle D, W\rangle$, $M \models \alpha$.

2.1 Computational complexity

We assume that the reader is familiar with the basic concepts of computational complexity. We use the standard notation of complexity classes that can be found in [12]. Namely, the class P denotes the set of problems whose solution can be found in polynomial time by a *deterministic* Turing machine, while NP denotes the class of problems that can be resolved in polynomial time by a *non-deterministic* Turing machine. The class coNP denotes the set of decision problems whose complement is in NP. We call NP-hard a problem G if any instance of a generic problem NP can can reduced to an instance of G by means of a polynomial-time (many-one) transformation (the same for coNP hard).

Clearly, P $\subseteq$ NP and P $\subseteq$ coNP. We assume, in line with the prevailing assumptions of computational complexity, that these containments are strict, that is P $\neq$ NP and P $\neq$ coNP. Therefore, we call a problem that is in P *tractable*, and a problem that is NP-hard or coNP-hard *intractable* (in the sense that any algorithm resolving it would require a super polynomial amount of time in the worst case).

We also use higher complexity classes defined using oracles. In particular P^A (NP^A) corresponds to the class of decision problems that are solved in polynomial time by deterministic (nondeterministic) Turing machines using an oracle for A in polynomial time (for a much more detailed presentation we refer the reader to [12]). All the problems we analyze reside in the *polynomial hierarchy*, that is the analog of the Kleene arithmetic hierarchy. The classes Σ_k^p, Π_k^p and Δ_k^p of the polynomial hierarchy are defined by

$$\Sigma_0^p = \Pi_0^p = \Delta_0^p = \mathrm{P}$$

and for $k \geq 0$,

$$\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}, \quad \Pi_{k+1}^p = \mathrm{co}\Sigma_{k+1}^p, \quad \Delta_{k+1}^p = \mathrm{P}^{\Sigma_k^p}.$$

Notice that $\Delta_1^p = \mathrm{P}$, $\Sigma_1^p = \mathrm{NP}$ and $\Pi_1^p = \mathrm{coNP}$. The definitions of hardness and completeness for all these classes are similar to those of NP-hardness and completeness.

3 COMPLEXITY: GENERAL CASE

We first focus on the most general situation where no constraints are imposed on the syntactic format of the defaults. In particular, we show that deciding whether a propositional interpretation is a model of a default theory under the skeptical semantics is Σ_2^p complete, even if the defaults are prerequisite-free and semi-normal.

In the following section we show that the problem is $\Delta_2^p[\log n]$ hard and in Δ_2^p, if all the defaults are normal (when prerequisites are allowed) and that model checking is coNP complete if all the defaults are both prerequisite-free and normal.

Our analysis takes advantage of the results obtained by Gottlob in [9], where he shows that the complexity of deciding whether $\langle D, W\rangle \models \alpha$ is Π_2^p complete even in the most restricted cases. This result holds even if all the defaults are prerequisite-free and normal.

Our results, that apply to all considered variants of default logic, are summarized in Table 1.

	General	Semi-Normal	Normal
General	Σ_2^p complete	Σ_2^p complete	$\Delta_2^p[\log n]$ hard, in Δ_2^p
Prerequisite-Free	Σ_2^p complete	Σ_2^p complete	coNP complete

Table 1. Complexity of Model Checking.

The membership to the class Σ_2^p, in the general case, is easy to prove.

Theorem 1 *For all the considered variants of default logics (classical, constrained, cumulative, justified default logics) deciding whether $M \models \langle D, W\rangle$ is Σ_2^p, in the most general case.*

Proof (sketch). We show this only for Reiter's original definition of default logic, for all other systems the proof is similar. In order to decide whether $M \models \langle D, W\rangle$, we guess a set $D' \subseteq D$ of defaults, check if D' is a generating set, compute the formula $T = W \wedge CONS(D')$ and check whether $M \models T$. The guess requires a nondeterministic choice while all other steps can be accomplished in polynomial time if we an NP oracle. Thus, the problem belongs to $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^p$. □

While the proof of membership is almost straightforward, the proof of hardness is a bit more complex.

Theorem 2 *Deciding whether $M \models \langle D, W\rangle$ is Σ_2^p hard, even if all the defaults are prerequisite-free and semi-normal.*

Proof. We prove the hardness of the problem by reduction from the problem QBF. Let $\exists X \forall Y.F$ be a quantified boolean formula. We prove that this formula is valid if and only if

$$M \models \langle D, W\rangle$$

where

$$\begin{aligned} W &= \emptyset \\ D &= \left\{ \frac{: w \wedge \neg F}{w} \right\} \bigcup_{x_i \in X} \left\{ \frac{: w \wedge (w \rightarrow x_i)}{w \rightarrow x_i}, \frac{: w \wedge (w \rightarrow \neg x_i)}{w \rightarrow \neg x_i} \right\} \\ M &= \emptyset \end{aligned}$$

The idea is that in all the extensions in which the first default is applicable, w holds, thus M is not a model of that extension. For example, if $\neg F$ is consistent, it is always possible to apply the first default, but the extension obtained in this manner does not have M as a model.

1. Let us assume that there is a truth evaluation $X_1 \subseteq X$ such that, for all the evaluations $Y_1 \subseteq Y$ it holds that $X_1 \cup Y_1$ is a model of F. Consider the following set of formulas.

$$A = \bigcup_{x_i \in X_1} w \rightarrow x_i \cup \bigcup_{x_i \notin X_1} w \rightarrow \neg x_i$$

The set $E = Cn(A)$ is an extension since it contains W, it is closed under deduction, and for each default either the default is not applicable or the consequence is in E. Let us prove that last point. Consider the first default. We prove that $\neg(w \wedge \neg F) \in E$, which is equivalent to $A \models \neg(w \wedge \neg F)$. Applying the properties of propositional calculus we obtain

$$\begin{aligned} A \models \neg(w \wedge \neg F) \quad & \text{iff} \quad A \models \neg w \vee F \\ & \text{iff} \quad A \models w \rightarrow F \\ & \text{iff} \quad A \cup \{w\} \models F \\ & \text{iff} \quad \bigcup_{x_i \in X_1} x_i \cup \bigcup_{x_i \notin X_1} \neg x_i \models F \end{aligned}$$

The last formula is true, because for each Y_1 the interpretation $X_1 \cup Y_1$ is a model of F. As a result, the justification of the first default is false in E.

Consider the default $\frac{: w \wedge (w \rightarrow x_i)}{w \rightarrow x_i}$. If $w \rightarrow x_i \in A$, then the justification is consistent with A, and the consequence of the default is in E. On the other hands, if $w \rightarrow \neg x_i \in A$, then the justification of the default is false in E.

Summarizing, the set E is an extension since it contains W, it is closed under deduction, and for any default either the justification is false or the consequence is in the set. Since the interpretation M is a model of E, it holds that M is also a model of the default theory $\langle D, W\rangle$.

2. Let now assume that there exists an extension E such that $M \models E$. This implies that $w \notin E$, as a result the justification of the first default must be false in E, that is $E \models \neg(w \wedge \neg F)$. This condition is equivalent to $E \models w \rightarrow F$, which is in turns equivalent to $E \cup \{w\} \models F$.

Now, the last default has not been applied. About the other defaults: for each x_i either $w \rightarrow x_i$ or $w \rightarrow \neg x_i$ is in E. As a result, $E \cup \{w\}$ gives a complete evaluation of the atoms in X. Since $E \cup \{w\} \models F$, it follows that there exists a truth evaluation of the atom in X such that F is valid for each evaluation of the atoms in Y. □

The following theorem shows the complexity of model checking for constrained, cumulative and justified default logic [14, 1, 6].

Theorem 3 *Deciding whether $M \models \langle D, W\rangle$ is Σ_2^p hard for constrained, cumulative and justified default logic. The results hold even if all the defaults are prerequisite-free and semi-normal.*

Proof (sketch). For constrained and justified default logic the proof follows the same lines of the proof of Theorem 2. For cumulative default logic, the result immediately follows from the proof for constrained default logic and [6, Theorem 5.10] where it is shown an equivalence with cumulative default logic where formulae in W have empty support. □

4 COMPLEXITY: NORMAL DEFAULTS

Both constrained and justified default logic have been proposed to overcome some shortcomings of classical default logic in the treatment of the justifications and their mutual consistency. When we restrict to normal defaults all the differences disappear and these three systems coincide. Moreover, they also coincide with cumulative default logic when the initial set of facts W has empty support. Therefore, our lower bounds for classical default logic immediately apply to all systems, while the upper bounds applies to all systems except cumulative default logic for which we provide a different proof.

We first show an upper bound on the complexity of model checking for normal default logic with prerequisites:

Theorem 4 *Deciding whether $M \models \langle D, W\rangle$ is Δ_2^p, if all the defaults are normal. This also holds for cumulative default logic.*

Proof (sketch). Let G be a subset of D defined as follows.

$$G = \left\{ \frac{\alpha : \beta}{\beta} \,\middle|\, M \models \beta \right\}$$

The set $D \backslash G$ has an important property: if any of the defaults in $D \backslash G$ is in the set of generating defaults of an extension, then M is not a model of that extension. This holds because, by construction, $\frac{\alpha' : \beta'}{\beta'} \in D \backslash G$ implies $M \not\models \beta$.

The algorithm to verify whether $M \models \langle D, W\rangle$ is defined as follows. Given the model M, we construct the set $G = \{d_1, \ldots, d_k\}$, initialize a variable E to W and set an index i to 1. This index ranges over $[1..k]$. We denote, for each i, $d_i = \frac{\alpha_i : \beta_i}{\beta_i}$.

The generic step i goes as follows: verify whether $E \models \alpha_i$ and $E \wedge \beta_i$ is consistent. If this is the case, let $E = E \cup \{\beta_i\}$, $i = 1$ and $D = D \backslash \{d_i\}$. Otherwise, let $i = i+1$. The reason we reset i to 1 is that, if a default i is discovered to be applicable, its consequence β_i comes to be true. It may be the case that there is a default d_j with $j < i$ such that $E \not\models \alpha_j$ but $E \cup \{\beta_i\} \models \alpha_j$. As a result, the defaults $d_1, \ldots, d_{i+1}$ have to be checked again. This is why we reset i to 1.

The algorithm ends when i has reached the last default of G, and has to be increased again.

This way, E is obtained by applying some defaults of G. In order to build an extension, we have to verify whether the defaults of $D \backslash G$ are applicable. This is done by verifying whether $E \models \alpha'$ and $E \wedge \beta'$ is consistent for each default $\frac{\alpha' : \beta'}{\beta'} \in D \backslash G$. If one of those default is applicable, then β' should be put in E. Since $M \not\models \beta'$, if follows that $M \not\models \langle D, W \rangle$, that is, M is not a model of the default theory.

Summarizing, there are two consistency checks for each default. The first phase of the algorithm analyzes at most $k^2/2$ defaults, thus there are at most k^2 consistency checkings. The second phase requires at most $2(n-k)$ consistency checking, where n is the total number of default in the theory. Thus, the whole algorithm can be executed with a polynomial number of calls to an NP oracle.

A similar construction proves the same result for Brewka's cumulative default logic. We construct the set G as above, then we use the boolean function *ROBUST* presented by Gottlob and Mingyi in [10, pg. 339] to decide whether G is in fact a set of generating defaults. Since Gottlob and Mingyi have shown that this boolean function can be decided with a polynomial number of calls to an NP oracle, the thesis follows. □

Notice that the complexity has decreased by almost one level (from Σ_2^p to Δ_2^p) by restricting to normal defaults. We want to point out that this phenomenon did not arise with the complexity of inference [9] where the complexity for general and normal default theories is the same.

In order to determine the lower bound of the complexity for normal default with prerequisites, we need the following Lemma:

Lemma 5 *Given a propositional formula F built on an alphabet of propositional letters $\{x_1, \ldots, x_n\}$, deciding whether the letter x_k ($k = \log n$) is not in the minimal lexicographic model of a formula F is $\Delta_2^p[\log n]$ complete.*

Theorem 6 *Deciding whether $M \models \langle D, W \rangle$ is $\Delta_2^p[\log n]$ hard, if all the defaults are normal.*

Proof. We prove the claim by reduction from the problem of the lexicographically minimal model above.

Consider the case $k = 2$. There are four possible cases.

1. $F \wedge \neg x_1$ inconsistent and $F \wedge \neg x_2$ inconsistent.
2. $F \wedge \neg x_1$ inconsistent and $F \wedge \neg x_2$ consistent.
3. $F \wedge \neg x_1$ consistent and $F \wedge \neg x_1 \wedge \neg x_2$ inconsistent.
4. $F \wedge \neg x_1$ consistent and $F \wedge \neg x_1 \wedge \neg x_2$ consistent.

The result should be "yes" in cases 2 and 4, and "no" in the other two cases. We use the defaults to capture each case. Namely, we prove that $M \models \langle D, W \rangle$ if and only if the lexicographically minimal model of F has not x_2, where

$$\begin{aligned} W &= \{w \rightarrow F\} \\ D &= \left\{ \frac{w \rightarrow (x_1 \wedge x_2) : w}{w} \,,\; \frac{w \rightarrow (x_1 \vee x_2) : w \wedge \neg x_1}{w \wedge \neg x_1} \right\} \\ M &= \emptyset \end{aligned}$$

The idea is: since $M \models W$, if no default is applicable, then M is a model of the extension. The first default is applicable only in case 1, while the second one is applicable only in case 3. The consequence of both defaults contains w. As a result, if any of the defaults is applicable then M is not in the extension.

Consider the first default. It is applicable only if $W \models w \rightarrow (x_1 \wedge x_2)$. This is equivalent to say that $F \wedge \neg x_1$ and $F \wedge \neg x_2$ are both inconsistent. The consequence of the default is w, thus M is not in the extension.

The second default is applicable only if $W \models w \rightarrow (x_1 \vee x_2)$, and $w \wedge \neg x_1$ is consistent with W. The first condition is equivalent to the inconsistency of $F \wedge \neg x_1 \wedge \neg x_2$, while the second one is equivalent to the consistency of $F \wedge \neg x_1$. This is exactly the case 3.

This way we proved that it is possible to determine the value of the second variable of the lexicographically minimal model of a formula. This idea can be extended to determine the value of the k-th variable. The idea is as follows. Since $k = \log n$, there are only $2^k = n$ possible cases. Half of these cases leads to a minimal model that contains x_k. For each of these cases we put a default with w among the consequence, so that M is excluded from the extension. Each case can be verified with (at most) k checks of inconsistency and a single check of consistency.

For each of these cases we use a single default. If the inconsistency checks are on the formulas $F \wedge R_1, F \wedge R_2, \ldots, F \wedge R_m$, the prerequisite of the default is $w \rightarrow (\neg R_1 \wedge \cdots \wedge \neg R_m)$. If the consistency check is on $F \wedge S$, the justification (and consequence) of the default is $w \wedge S$.

Since these are n defaults, this is a reduction from the problem of the minimal model to model checking of default logics. □

If all the defaults are prerequisite-free and normal, then the problem is only coNP complete.

Theorem 7 *The complexity of model checking of skeptical default is* coNP *complete, if all the defaults are prerequisite-free and normal.*

Proof (sketch). Hardness is proved by reduction from the problem of model checking for circumscription. In fact, Cadoli [2] has shown that model checking for circumscription is coNP complete and Etherington [7] has shown a reduction from circumscription into prerequisite-free normal default logic.

The membership is harder to prove. Let M be an interpretation and $\langle D, W \rangle$ a default theory. First check if $M \models W$. If this is not the case then M is not a model of $\langle D, W \rangle$. Otherwise, consider the following set of defaults

$$G = \left\{ \frac{: \beta}{\beta} \;\middle|\; M \models \beta \right\}$$

This set has a property: the union of the consequence of all these defaults is consistent (notice that all of them have M as a model). Let A be

$$A = \bigcup_{\frac{:\beta}{\beta} \in G} \beta$$

If the deductive closure of $A \cup W$ is an extension, then M is a model of the default theory. The only thing that prevents $Cn(A \cup W)$ from being an extension is the fact that there is some other default $\frac{:\beta'}{\beta'} \in D \backslash G$ such that β' is consistent with $A \cup W$.

This can be verified with a single check of consistency. Since $M \models \langle D, W\rangle$ if and only if the formulas β' are not consistent with $A \cup W$, the problem of model checking is coNP complete. □

5 CONCLUSIONS AND FUTURE WORK

In this paper we have presented an analysis of the complexity of model checking for various variants of default logic. Not surprisingly all the considered variants of default logic have the same complexity and restrictions on the syntactic form of the defaults may decrease the complexity. This phenomenon does not arise in the complexity of inference [9].

Model-based representations of information are often advocated because they make the reasoning tasks simpler (model checking for propositional logic can be accomplished in linear time). These results may seem discouraging, since all computational tasks are polynomially intractable. However, it has been shown [8] that default logic makes it possible to represent information in a very compact way. Using our analysis, it is easy to exactly characterize, in the style of [3], the compactness of the various variants of default logic and of all the restricted cases here analyzed.

REFERENCES

[1] G. Brewka, 'Cumulative default logic: in defense of nonmonotonic inference rules', *Artificial Intelligence*, **50**(2), 183–205, (1991).

[2] M. Cadoli, 'The complexity of model checking for circumscriptive formulae', *Information Processing Letters*, **44**, 113–118, (1992).

[3] M. Cadoli, F. Donini, P. Liberatore, and M. Schaerf, 'Comparing space efficiency of propositional knowledge representation formalisms', in *Proc. of KR'96*, pp. 364–373, (1996).

[4] M. Cadoli, F. M. Donini, P. Liberatore, and M. Schaerf, 'The size of a revised knowledge base', in *Proc. of PODS'95*, pp. 151–162, (1995). Extended version available as Technical Report DIS 34-96, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", November 1996.

[5] M. Cadoli, F. M. Donini, M. Schaerf, and R. Silvestri, 'On compact representations of propositional circumscription', *Theor. Comp. Sci.*, **182**, 183–202, (1997).

[6] J. P. Delgrande, T. Schaub, and W. K. Jackson, 'Alternative approaches to default logic', *Artificial Intelligence*, **70**, 167–237, (1994).

[7] D. V. Etherington, 'Relating default logic and circumscription', in *Proc. of IJCAI'87*, pp. 489–494, (1987).

[8] G. Gogic, H. A. Kautz, C. Papadimitriou, and B. Selman, 'The comparative linguistics of knowledge representation', in *Proc. of IJCAI'95*, pp. 862–869, (1995).

[9] G. Gottlob, 'Complexity results for nonmonotonic logics', *J. of Log. and Comp.*, **2**, 397–425, (1992).

[10] G. Gottlob and Z. Mingyi, 'Cumulative default logic: finite characterization, algorithms and complexity', *Artificial Intelligence*, **69**, 329–345, (1994).

[11] J. Y. Halpern and M. Y. Vardi, 'Model checking vs. theorem proving: A manifesto', in *Proc. of KR'91*, (1991). Also in Lifshitz V. *Artificial Intelligence and Mathematical Theory of Computation. Papers in Honor of John McCarthy*, Academic Press, San Diego, 1991.

[12] D. S. Johnson, 'A catalog of complexity classes', in *Handbook of Theoretical Computer Science*, ed., J. van Leeuwen, volume A, chapter 2, 67–161, Elsevier, (1990).

[13] H. J. Levesque, 'Making believers out of computers', *Artificial Intelligence*, **30**, 81–108, (1986).

[14] W. Lukaszewicz, 'Considerations on default logic: an alternative approach', *Computational Intelligence*, **4**(1), 1–16, (1988).

[15] R. Reiter, 'A logic for default reasoning', *Artificial Intelligence*, **13**, 81–132, (1980).