Fundamenta Informaticae XX (1997) 1–16
IOS Press

Well-Founded Semantics for Default Logic

Gerhard Brewka
Institut für Informatik
Universität Leipzig
Augustusplatz 10-11, 04109 Leipzig, Germany
brewka@informatik.uni-leipzig.de

Georg Gottlob
Institut für Informationssysteme
Paniglgasse 16, Technical University of Vienna
1040 Vienna, Austria
gottlob@dbai.tuwien.ac.at

Abstract. Default logic is one of the most popular approaches to model defeasible reasoning. Nevertheless, there are a number of problems with Reiter's original semantics that have led to the investigation of alternative approaches. In particular, Baral/Subrahmanian and Przymusinska/Przymusinski have investigated generalizations of well-founded semantics for normal logic programs to default logic.

These generalizations have a number of interesting properties. Unfortunately, it turns out that in many realistic situations they are unable to draw any defeasible conclusions at all - which can hardly be viewed as satisfactory. We show how this difficulty can be solved by varying the fixed point operator underlying the semantics. We define a range of different semantics. All of them are correct wrt. safe conclusions under Reiter semantics, i.e. those conclusions with the same proof in all extensions. For the strongest semantics we have also completeness in the case of coherent default theories, i.e. default theories with at least one extension. The logics differ in the effort spent for determining potential conclusions. It turns out that they are at least as complex as original default logic.

We show that our approach also leads to new semantics for normal and extended logic programs. Moreover, we define prioritized versions of the logics.

Keywords: Nonmonotonic reasoning, default logic, well-founded semantics.

1. Introduction

We investigate in this paper semantics for Reiter's default logic [19] which are based on least fixpoints of monotone operators. Such semantics have their roots in logic programming, in particular in well-founded semantics [6, 16, 2], one of the by now standard semantics for logic programs. The success of well-founded semantics in logic programming suggests that its underlying techniques may prove fruitful in other areas of nonmonotonic reasoning as well. The goal of this paper is to answer the question whether these techniques can be generalized in an interesting and useful manner to full default logic.

The answer we will give has both a positive and a negative part. It turns out that we can indeed define a reasonable semantics for default logic based on least fixpoints of adequate monotone operators. This is good news, in particular for those who consider well-founded semantics a genuine semantics of its own determining the right meaning of a logic program, respectively a default theory, in the first place. On the other hand, our complexity analysis will show that the main reasoning tasks in our well-founded default logics are at least as complex as those in original default logic, and in an important case even harder. This is bad news, at least for those whose main interest in well-founded semantics stems from the fact that this semantics can be viewed as an approximation of the original semantics, that is, stable model semantics [7] in the case of logic programs and Reiter's semantics in the case of default logic. In logic programming well-founded semantics leads to polynomial algorithms. Our results imply that no corresponding gain in efficiency is to be achieved for full default logic.

The idea of extending well-founded semantics to default logic is not new. In fact, two such extensions are described in the literature, namely Baral and Subrahmanian's well-founded semantics for default logic [2] and Przymusinska and Przymusinski's stationary semantics [18]. Both approaches are closely related and agree on the set of skeptical conclusions. In comparison with original default logic they have the following advantages:

1. Since the definition of the semantics is based on a monotone operator the existence of a least fixpoint is guaranteed. Default theories that lack extensions thus have a reasonable consequence relation under the new semantics.
2. The least fixpoint can be approximated from below by iterating the monotone operator on the empty set. This gives rise to an iterative procedure for determining default conclusions.
3. The semantics are cumulative, i.e., adding a (skeptical) conclusion to the premises does not change the set of obtainable conclusions.
4. As shown in [10] computing skeptical conclusions for propositional default theories is on the first level of the polynomial hierarchy and thus more efficient than in Reiter's original approach.

Unfortunately, both approaches suffer from a serious weakness: it turns out that in many natural situations no defeasible conclusions at all are obtained. In these situations the approaches simply break down to monotonic reasoning from the available facts.

This can obviously not be viewed as a satisfactory treatment of default reasoning or, for that matter, as a reasonable generalization of well-founded semantics to default logic. We therefore investigate in this paper alternative generalizations which do not suffer from the mentioned problems. Our generalizations have the following two properties in common with Baral/Subrahmanian's and Przymusinska/Przymusinski's approaches:

1. they are based on monotone operators and define conclusions in terms of least fixpoints of these operators,
2. they are correct wrt. safe Reiter conclusions, that is, conclusions having the same proof in all extensions (a precise definition will be given in the next section).

We consider these two properties as characteristic for the family of well-founded semantics.

The outline of the paper is as follows: Section 2 briefly reviews default logic and introduces some new terminology that will turn out to be useful later. Section 3 discusses Baral/Subrahmanian's well-founded and Przymusinska/Przymusinski's stationary semantics for default logic and analyzes their weaknesses. Section 4 introduces our new monotone operators and presents a number of results and examples. Section 5 analyzes the complexity of the main reasoning tasks. Section 6 shows that our approach yields new semantics also for normal and extended logic programs which can be viewed as special default theories. Section 7 shows how prioritized versions of the semantics can be defined. Section 8 concludes.

2. Default logic: a short review

A default theory T is a pair consisting of a set of facts W and a set of defaults D. The facts are standard first order sentences. Each default is of the form $a : b_1, \ldots, b_n/c$ where a, b_i, and c are sentences. The intuitive meaning of the default is: if a is derived and the b_i are consistent with what is derived then infer c. a is called prerequisite, b_i justification, and c consequent of the default. For a default d we use $pre(d)$, $just(d)$, and $cons(d)$ to denote the prerequisite, the set of justifications, and the consequent of d, respectively. We will sometimes use default schemata to represent the set of all ground instances of a default.

A default theory generates extensions which represent acceptable sets of beliefs a reasoner might adopt based on the given default theory (D, W). Extensions are defined as fixpoints of an operator Γ_T. Γ_T maps an arbitrary set of formulas S to the smallest deductively closed set S' that contains W and satisfies the condition: if $a : b_1, \ldots, b_n/c \in D$, $a \in S'$ and $\neg b_i \notin S$ then $c \in S'$. Default theories possessing at least one extension will be called coherent.

We will later use the following notion of a default proof. A similar notion was also used by Marek and Truszczynski [12]:

Definition 2.1. Let $T = (D, W)$ be a default theory, $D' \subseteq D$, and p a formula. A T-default proof for p from D' is a finite sequence $(d_1, \ldots, d_n)$ of defaults in D' such that the following conditions are satisfied:

1. $W \cup \{cons(d_1), \ldots, cons(d_{i-1})\} \vdash pre(d_i)$, for $i \in \{1, \ldots, n\}$,
2. $W \cup \{cons(d_1), \ldots, cons(d_n)\} \vdash p$.

As usual $\vdash$ denotes classical provability. We say a default $a : b_1, \ldots, b_n/c$ is defeated by a set of formulas S iff $\neg b_i \in S$ for some $i \in \{1, \ldots, n\}$. Using the notation

$$D^S = \{d \in D \mid \ d \text{ is not defeated by } S\}$$

we can obviously characterize $\Gamma_T(S)$ as follows

$$\Gamma_T(S) = \{p \mid p \text{ has a } T\text{-default proof from } D^S\}.$$

Extensions can be used to define different inference relations for default logic. Two of them, credulous and skeptical inference, are standard. We will define a third one which is somewhat less standard but relevant for our purposes. Some new terminology is convenient. Let $P = (d_1, \ldots, d_n)$ be a T-default proof and E an extension of T. We say P is valid in E iff for all $j \in \bigcup_{1 \leq i \leq n} just(d_i)$ we have $\neg j \notin E$.

Definition 2.2. Let $T = (D, W)$ be a coherent default theory. A formula p is called

credulous consequence of T	iff	p is in at least one extension of T,
skeptical consequence of T	iff	p is in all extensions of T,
safe consequence of T	iff	there is a T-default proof P for p valid in all extensions of T.

The set of credulous, skeptical, and safe conclusions of T will be denoted $Cred(T)$, $Skep(T)$, and $Safe(T)$, respectively. Simply extending the definition above to arbitrary default theories would lead to a somewhat strange situation where, for an incoherent theory T, $Cred(T) = \emptyset$ whereas $Skep(T) = Lang$, where $Lang$ denotes the set of all formulas. For an incoherent default theory T we therefore define $Cred(T) = Skep(T) = Safe(T) = Lang$. Now we obviously have, for all T, $Safe(T) \subseteq Skep(T) \subseteq Cred(T)$. Reasoning based on safe conclusions is thus even more cautious than skeptical reasoning. Intuitively, a safe reasoner does not accept a skeptical conclusion p unless p is believed in all extensions *for the same reason.* Whenever the arguments for p in different extensions are incompatible with each other this is taken to raise some doubt about p, and the formula is rejected.

In the rest of the paper we will often omit the index, respectively prefix T from Γ_T and T-default proof whenever T is clear from context. Moreover, we will apply the operator

Th also to default theories. If $T = (D, W)$ then $Th(T)$ denotes the smallest set of formulas which contains W, is closed under classical deduction and under the justification-free defaults in D. $Th(T)$ is thus equivalent to $Th(W)$ whenever D does not contain a default without justification, but may be larger in the general case.

3. Well-founded and stationary default logic

Baral and Subrahmanian [2] and Przymusinska and Przymusinski [18] have introduced alternative semantics for default logic, called well-founded respectively stationary semantics, which - as discussed in the introduction - have several advantages over Reiter's original semantics. In particular, the semantics are based on fixpoints of a monotone operator. This guarantees the existence of a least fixpoint. Default theories that lack extensions thus have a reasonable consequence relation under the new semantics. Moreover, the least fixpoint can be approximated from below by iterating the monotone operator on the empty set.

Both approaches differ in the fixpoints they consider for credulous reasoning. However, they use the same underlying monotone operator and thus obtain the same least fixpoint. Since we will not be concerned with credulous reasoning we will focus here on the parts the approaches agree upon and refer the reader to the original papers for the differences.

Well-founded semantics was originally defined for logic programs. The original formulation in [6] is based on a certain partial model. Przymusinski later reconstructed this definition in 3-valued logic [16]. Baral and Subrahmanian gave a reformulation based on the double application of an anti-monotone operator and extended this definition to default logic. We will only present the definition for default logic here which is also used in [18].

We should point out that, in order to simplify our presentation, we restrict ourselves throughout the paper to the literals which are true according to the three-valued formulation of the semantics. We call them well-founded conclusions. The literals which are false will be left implicit. They can be determined in a canonical way as follows: let T, the set of true literals, be defined as the least fixed point of a monotone operator composed of two antimonotone operators $op_1 op_2$. Then the literals which are false in the three-valued model are exactly those which are not contained in $op_2(T)$. Given this canonical extension to the full three-valued model we can safely leave the false literals implicit from now on.

The definition of the well-founded conclusions is based on the observation that Reiter's Γ-operator is anti-monotone, that is

$$S \subseteq S' \text{ implies } \Gamma(S') \subseteq \Gamma(S).$$

The twofold application of Γ thus defines a monotone operator. According to the Knaster-Tarski theorem [22] this monotone operator has a least fixpoint. The set of well-founded conclusions of a default theory T, denoted $WFS(T)$, is defined to be this least fixpoint of Γ^2. The fixpoint can be approached from below by iterating Γ^2 on the empty set. In case T is finite this iteration is guaranteed to actually reach the fixpoint.

The intuition behind this approach is as follows: assume S is a set of formulas known to be provable. The first application of Γ produces the set of all formulas which still are potentially derivable. It is clear that formulas not contained in $\Gamma(S)$ can never be derived. Therefore, formulas which have a default proof from $D^{\Gamma(S)}$ are certainly provable, that is, the set of formulas produced by Γ^2 can safely be assumed true.

It is well-known that well-founded semantics, WFS from now on, is correct wrt. skeptical inference under Reiter's original extension semantics, i.e., all well-founded conclusions are skeptical conclusions. Well-founded semantics can thus be viewed as an approximation of Reiter's semantics. We can strengthen this result somewhat:

Proposition 3.1. Let T be a default theory. We have $WFS(T) \subseteq Safe(T)$.

Proof:
Follows immediately from Proposition 2, see Section 4.

WFS has several desirable properties. As shown by Przymusinska and Przymusinski the semantics is cumulative, that is:

$$p \in WFS((D,W)) \text{ implies } WFS((D,W)) = WFS((D, W \cup \{p\})).$$

Moreover, its computational complexity is better than that of original default logic. Gottlob [10] has shown that for propositional default theories determining whether $p \in WFS((D,W))$ is on the first level of the polynomial hierarchy. For original default logic this task is on the second level.

Unfortunately it turns out that for many default theories the set of well-founded conclusions is intolerably small and provides a very poor approximation of the original semantics. Consider the following default theory T_1. The corresponding logic program has also been discussed in [1]:

$$\begin{array}{ll} 1)\ :b/b & 3)\ :\neg a/\neg a \\ 2)\ :a/a & \end{array}$$

The set of well-founded conclusions contains only the tautologies since $\Gamma(\emptyset) = Lang$ and $\Gamma(Lang) = Th(\emptyset)$. This is surprising since, intuitively, the conflict between 2) and 3) has nothing to do with the first default rule and we would thus expect b to be a well-founded conclusion.

This problem arises whenever the following conditions hold:

1. $\Gamma(\emptyset)$ is inconsistent, and
2. the derivation of the inconsistency does not depend on default rules defeated by $Th(T)$.

In this case well-founded semantics concludes p iff $p \in Th(T)$. In other words: no defeasible conclusion at all is obtained. It should be obvious that such a situation is not just a rare limiting case. To the contrary, it can be expected that most commonsense knowledge bases will give rise to such undesired behaviour. For instance, assume a knowledge base contains information that birds normally fly and penguins normally don't, expressed as the set of ground instances of the following default schemata:

$$\begin{array}{l} 1)\ bird(x) : fly(x)/fly(x) \\ 2)\ penguin(x) : \neg fly(x)/\neg fly(x) \end{array}$$

Assume further the knowledge base contains the information that Tweety is a penguin bird. Now if neither $fly(Tweety)$ nor $\neg fly(Tweety)$ is in $Th(T)$ then we are in the same situation as in the earlier example: well-founded semantics does not draw any defeasible conclusion at all, i.e. no single default rule (except degenerate ones without justification) - whether related to flying, penguins, birds or not - will be used to draw a conclusion. This can hardly be viewed as a satisfactory treatment of default reasoning.

4. The Framework

In this section we want to show that a reformulation of the fixpoint operator can overcome the weakness of well-founded and stationary default logic. More precisely, we will define a range of operators. All of them are monotone and define a semantics WFS_i which is correct wrt. safe Reiter conclusions. The least fixpoints (lfp's from now on) of the "interesting" operators will be supersets of the lfp of Γ^2 and the new semantics thus stronger in the sense that they yield more conclusions.

To motivate our approach let us try to identify the source of the weakness of WFS. The problem is that the use of Γ for establishing potential conclusions provides a very rough estimate only. Often far too many formulas are considered as potential conclusions. This is the case in particular when $\Gamma(S)$ is inconsistent. What we will do, therefore, is replace the first application of Γ by more refined operators, that is, instead of considering the monotone operator Γ^2 we will consider monotone operators composed of two different anti-monotone operators. More precisely, we investigate operators of the form

$$\Gamma\Gamma_i$$

where Γ_i is anti-monotone. A few observations before we introduce the new operators. For two operators op_i, op_j we use $op_i \leq op_j$ as an abbreviation for

$$op_i(S) \subseteq op_j(S) \text{ for all } S.$$

Now it is obvious that $\Gamma\Gamma_i \leq \Gamma\Gamma_j$ whenever $\Gamma_j \leq \Gamma_i$. In other words, if Γ_j produces fewer potential conclusions than Γ_i, then the lfp produced by the monotone operator $\Gamma\Gamma_j$ will be at least as big as the lfp of $\Gamma\Gamma_i$.

As discussed in Section 2 we can characterize Reiter's operator as follows:

$$\Gamma_T(S) = \{p \mid \text{ there is a } T\text{-default proof for } p \text{ from } D^S\}.$$

Our general strategy will be to impose certain additional conditions on default proofs. We first introduce special types of default proofs.

Definition 4.1. Let T be a default theory, $P = (d_1, \ldots, d_n)$ a T-default proof, S a set of formulas. We say P is

S-consistent	iff	$S \cup \{cons(d_1), \ldots, cons(d_n)\}$ is consistent,
S-justifiable	iff	$S \cup \{cons(d_1), \ldots, cons(d_n)\} \cup \{j_k\}$ is consistent, for each justification j_k of a default in P
S-extendable	iff	there is a consistent extension E of T such that $S \subseteq E$ and P is valid in E.

To be S-consistent a default proof thus has to satisfy a certain local consistency condition: the consequents of the used defaults have to be jointly consistent with S. S-justifiable default proofs additionally take justifications of the used defaults into account. However, this is still done in a local manner: it is only checked whether each justification is consistent with S together with all consequents of defaults in the proof. For S-extendable proofs finally a global check is performed: it is required that an extension E containing S actually exists in which the default proof is valid. We obviously have

$$P\ S\text{-extendable} \Rightarrow P\ S\text{-justifiable} \Rightarrow P\ S\text{-consistent}.$$

Based on these notions we define the following operators:

Definition 4.2. Let T be a default theory, S a set of formulas. For $1 \leq i \leq 4$ we define an operator Γ_i as follows:

$$\begin{aligned}
\Gamma_1(S) &= \{p \mid \text{ there is a } T\text{-default proof for } p \text{ from } D^S\}\\
\Gamma_2(S) &= \{p \mid \text{ there is an } S\text{-consistent } T\text{-default proof for } p \text{ from } D^S\}\\
\Gamma_3(S) &= \{p \mid \text{ there is an } S\text{-justifiable } T\text{-default proof for } p \text{ from } D^S\}\\
\Gamma_4(S) &= \{p \mid \text{ there is an } S\text{-extendable } T\text{-default proof for } p \text{ from } D^S\}
\end{aligned}$$

All defined operators are obviously anti-monotone. Operator Γ_i induces a semantics WFS_i through the equation $WFS_i(T) = lfp(\Gamma\Gamma_i)$. Γ_1 is just Γ, that is it induces WFS, respectively stationary semantics. The more interesting operators are the other ones. Γ_2, Γ_3, and Γ_4 are obtained by imposing increasingly stronger restrictions on the admitted default proofs. These operators are used to estimate potential conclusions. They thus offer a choice of how much effort - more precisely, how much consistency checking - we want to spend for this estimation.

Let us first reconsider the example used earlier to illustrate the weakness of original WFS and stationary semantics.

$$\begin{array}{ll} 1)\ :b/b & 3)\ :\neg a/\neg a \\ 2)\ :a/a & \end{array}$$

WFS and stationary semantics produce, as we saw, the set of all tautologies. Now consider WFS_2. We first apply Γ_2 to the empty set and obtain $\Gamma_2(\emptyset) = Th(\{a, b\}) \cup Th(\{\neg a, b\})$. Note that this set is not deductively closed and in particular does not contain $\neg b$. Default 1) is thus not "defeated" by $\Gamma_2(\emptyset)$ and therefore $\Gamma\Gamma_2(\emptyset) = Th(\{b\})$. This is also the least fixpoint of $\Gamma\Gamma_2$. We thus conclude b as intended. The same result is obtained in WFS_3 and WFS_4 as can easily be verified. Intuitively, the reason that our new semantics do not suffer from the weakness of original WFS and stationary semantics is that they are able to keep the effects of inconsistencies in the potential conclusions local.

Before discussing further examples let us establish a few results.

Proposition 4.1. WFS_4 is correct wrt. safe Reiter conclusions, i.e., for all T

$$WFS_4(T) \subseteq Safe(T).$$

Proof:
It suffices to show that $\Gamma\Gamma_4(S)$ is a set of safe formulas whenever S is. This is trivially true if either T has no extension at all or its single extension is inconsistent. Therefore assume T has at least one extension and all extensions of T are consistent.

If $p \in \Gamma\Gamma_4(S)$ then there is a default proof P for p from $D^{\Gamma_4(S)}$. We show that P is valid in all extensions, and hence p is a safe conclusion of T.

Assume there is a (consistent) extension E in which P is not valid. Then there is a justification q of one of the defaults in P such that $\neg q \in E$. But then, since S is a set of safe formulas and thus must be contained in E, there is an S-extendable default proof for $\neg q$ and thus $\neg q \in \Gamma_4(S)$. But then contrary to our assumption P cannot be a default proof for p from $D^{\Gamma_4(S)}$ and the proof is complete.

We obviously have

$$\Gamma_4 \leq \Gamma_3 \leq \Gamma_2 \leq \Gamma_1$$

and therefore:

Corollary 4.1. Let T be a default theory. We have

$$WFS_1(T) \subseteq WFS_2(T) \subseteq WFS_3(T) \subseteq WFS_4(T).$$

All semantics induced by one of our operators thus are correct wrt. safe conclusions.

For WFS_4 we can even show completeness in the case of coherent default theories, i.e. theories with at least one Reiter extension.

Proposition 4.2. Let T be a coherent default theory. We have

$$Safe(T) \subseteq WFS_4(T).$$

Proof:
There are two cases: either T has the single inconsistent extension, in which case $Th(T)$ and thus $WFS_4(T)$ is also inconsistent, or T has at least one extension and all extensions are consistent.

Consider the latter case. If $p \in Safe(T)$ then there exists a proof P for p valid in all extensions. We show that P is also a default proof from $D^{\Gamma_4(\emptyset)}$ and thus $p \in WFS_4(T)$.

Assume P is not such a proof. Then there exists a justification q of some default in P such that $\neg q \in \Gamma_4(\emptyset)$. This can only be the case if there is a consistent extension E of T containing $\neg q$. Hence P is not valid in E, contrary to our assumption.

From this completeness result the following corollary follows immediately.

Corollary 4.2. Let T be a default theory with unique extension E. We have

$$WFS_4(T) = E.$$

Finally, we give a result for normal default theories, i.e. theories where all defaults are of the form $a : b/b$.

Proposition 4.3. Let T be a normal default theory. Then

$$WFS_2(T) = WFS_3(T) = WFS_4(T).$$

Proof:
$WFS_2(T) = WFS_3(T)$ is obvious since $\Gamma_2(S) = \Gamma_3(S)$ for all S whenever T is normal. To show $WFS_2(T) = WFS_4(T)$ we consider two cases:

a) T has the single inconsistent extension:
In this case $Th(T)$ is inconsistent and therefore $\Gamma(S)$ is inconsistent for all S, which means the result of the first anti-monotone operator is irrelevant.

b) all extensions of $T = (D, W)$ are consistent:
In this case the semi-monotony of normal default theories [19] guarantees that every S-consistent proof is also S-extendable, that is, we have $\Gamma_2(S) = \Gamma_4(S) = Cred(T)$, whenever S is a set of safe formulas containing W. Since W is contained in $\Gamma\Gamma_i(\emptyset)$ for $i = 1, \ldots, 4$ the result follows.

As we have seen earlier the three semantics differ from WFS even for normal default theories.

The next example illustrates the difference between the semantics in the non-normal case.

1) $: a/a$	3) $: b/c$	5) $\neg a : b/\neg b$
2) $: \neg a/\neg a$	4) $: a/d$	

The following table shows the results of applying Γ_i and $\Gamma\Gamma_i$ to the empty set. In each case already a single application of $\Gamma\Gamma_i$ produces the lfp which can be read off from the bottom line of the table.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
$\Gamma_i(\emptyset)$	$Lang$	$Th(\{a, c, d\}) \cup$ $Th(\{\neg a, c, d, \neg b\})$	$Th(\{a, c, d\}) \cup$ $Th(\{\neg a, c\})$	$Th(\{a, c, d\})$
$\Gamma\Gamma_i(\emptyset)$	$Th(\emptyset)$	$Th(\emptyset)$	$Th(\{c\})$	$Th(\{a, c, d\})$

The example nicely illustrates how more effort spent to establish potential conclusions can lead to more conclusions.

Remark 1: It should be pointed out that the semantics as they stand were developed to account for safe reasoning in coherent default theories. For incoherent default theories safeness is trivially satisfied by all formulas and the notion does not provide much guidance in determining reasonable conclusions. In fact, to handle incoherence adequately extra care must be taken: in incoherent theories it may happen that $WFS_i(T) \not\subseteq \Gamma_i(WFS_i(T))$, that is, contrary to intuition conclusions are obtained which are not potential conclusions. Consider, for instance, the incoherent default theory $T = (D, \emptyset)$ where D consists of the single default rule $true : a/\neg a$. Now $WFS_3(T)$ is $\Gamma\Gamma_3(\emptyset) = Th(\{\neg a\})$ whereas $\Gamma_3(WFS_3(T))$ does not contain $\neg a$.

We can avoid problems of this kind by forcing certain formulas to be considered as potential conclusions even if they do not have a derivation. Let F be the least fixpoint of $\Gamma\Gamma_i$. We call a formula p suspect (wrt. WFS_i) whenever p is in F but not in $\Gamma_i(F)$. A formula q is called a culprit (wrt. WFS_i) iff

1. $(d_1, \ldots, d_n)$ is a nonredundant default proof for a suspect formula p from $D^{\Gamma_i(F)}$,
2. $cons(d_1) \wedge \ldots \wedge cons(d_{n-1})$ is not suspect, and
3. q is a justification of d_n.

Now we can invalidate the derivation of p by defining

$$WFS_i^*(T) = \Gamma(\Gamma_i(F) \cup \{\neg q \mid q \text{ is culprit wrt. } WFS_i\}).$$

Basically, the construction - which is somewhat reminiscent of dependency directed backtracking in truth maintenance systems [5] - blocks the derivation of formulas which are not potential conclusions. In our example formula $\neg a$ is suspect and $(true : a/\neg a)$ is the nonredundant default proof yielding $\neg a$. a is therefore a culprit and $WFS_3^*(T) = \Gamma(Th(\emptyset) \cup \{\neg a\}) = Th(\emptyset)$.

For coherent default theories WFS_i^* is equivalent to WFS_i since in that case we always have $F \subseteq \Gamma_i(F)$.

Remark 2: WFS_2, WFS_3 and WFS_4 are not cumulative, contrary to WFS. This is just to be expected since our semantics are much closer to default logic which itself violates cumulativity. We are not overly concerned about this and believe that for an adequate treatment of cumulativity consistency conditions need to be made explicit, as was done in [3]. We suspect that cumulative versions of our semantics can be defined along these lines without much difficulty.

5. Complexity analysis

In [9] it was shown that several forms of nonmonotonic reasoning are more complex than classical propositional reasoning. In particular, it was proven that skeptical default reasoning in Reiter's default logic is Π_2^P-complete and thus at the second level of the Polynomial Hierarchy, see also [20, 14]. In contrast, it was shown in [10] that skeptical reasoning in the well-founded default logics of Baral/Subrahmanian [2] and Przymusinska/Przymusinski [18], i.e., in WFS_1, is at the *first level of the Polynomial Hierarchy*, and thus computationally easier. This is a noticeable advantage of WFS_1 over classical default logic.

Unfortunately, as we show in this section, this advantage is not shared by our new versions $WFS_2 - WFS_4$. Skeptical reasoning in all these default logics is Π_2^P-hard and thus at least as hard as skeptical reasoning in classical default logic. This negative result holds even if we restrict our attention to coherent normal default theories. Moreover, we are able to show that skeptical reasoning in WFS_4 is Σ_3^P complete and thus *harder* than skeptical reasoning in classical default logic.

Recall that the complexity class Σ_2^P is the class of all problems solvable in polynomial time by a nondeterministic Turing machine having access to an oracle in NP and that its dual class Π_2^P is the class of all problems whose *complements* are in Σ_2^P; for more details, see [21, 11].

Proposition 5.1. Skeptical reasoning with coherent normal default theories according to the semantics WFS_2, WFS_3, and WFS_4 is Π_2^p hard.

Proof:
By Proposition 4.3 it suffices to prove the statement for WFS_4. The following problem $\mathcal{Q}$ is well-known to be Π_2^P complete [21]: Given a quantified propositional formula F of the form $\forall p_1, p_2, \ldots p_n \, \exists q_1, q_2, \ldots q_m \, \phi$, where ϕ is a propositional formula on propositional variables $p_1, p_2, \ldots, p_n$, $q_1, q_2, \ldots, q_m$, determine whether F is valid. F is valid iff for each truth assignment ν to the propositional variables $p_1, p_2, \ldots, p_n$ there is an extension ν' of ν covering also the propositional variables $q_1, q_2, \ldots, q_m$ such that ϕ is true under ν'.

We transform the problem $\mathcal{Q}$ as follows into a skeptical reasoning problem in WFS_4.

Let $T = (D, W)$ be the default theory defined by $W = \emptyset$ and

$$D = \{ :p_1/p_1,\ :\neg p_1/\neg p_1,\ \ :p_2/p_2,\ :\neg p_2/\neg p_2,\ \ldots, :p_n/p_n,\ :\neg p_n/\neg p_n\ :\phi/\phi \}.$$

Obviously T is a coherent normal default theory which is obtained from F in polynomial time. We claim that $\phi \in WFS_4(T)$ iff F is valid.

If. Assume F is valid. Then T has exactly 2^n Reiter extensions corresponding to the 2^n different choices between the defaults $: p_i/p_i$ and $: \neg p_i/\neg p_i$ for $1 \leq i \leq n$. Each such extension determines a truth value assignment to the propositional variables $p_1, p_2, \ldots, p_n$. Since F is valid, ϕ is consistent with each of these assignments, and hence the default $:\phi/\phi$ fires in each of the 2^n extensions. Thus ϕ belongs to each extension. Moreover, ϕ has the same valid default proof $\{: \phi/\phi\}$ in each of the extensions. Hence $\phi \in Safe(T)$, and thus, by Proposition 4.2, $\phi \in WFS_4(T)$.

Only If. We show that if F is not valid, then $\phi \notin WFS_4(T)$. Assume F is not valid. Then there exists a truth value assignment σ to $p_1, p_2, \ldots, p_n$ such that each extension σ' of σ to the variables $q_1, q_2, \ldots, q_m$ violates ϕ. Even though F is not valid, T has again exactly 2^n Reiter extensions in exact correspondence with the 2^n possible truth value assignments to the p_i variables. However, one extension E of T corresponds to the assignment σ and ϕ is thus inconsistent with E and hence $\phi \notin E$. Therefore, $\phi \notin Skep(T)$, whence $\phi \notin Safe(T)$, and, by Proposition 3.1, $\phi \notin WFS_4(T)$.

Proposition 5.1 shows that skeptical reasoning in our new well-founded default logics is as least as hard as the same reasoning task in in Reiter's classical default logic. But it comes worse. We can prove that skeptical reasoning for WFS_4 is even harder than skeptical reasoning in Reiter's default logic. In particular, we show that skeptical reasoning in WFS_4 is Σ_3^P complete, thus at the third level of the Polynomial Hierarchy. It follows that, unless the Polynomial Hierarchy collapses, WFS_4 reasoning cannot be polynomially translated into classical default reasoning.

The complexity class Σ_3^P consists of all problems solvable in polynomial time by a non-deterministic Turing machine having access to an oracle in Σ_2^P.

Proposition 5.2. Skeptical reasoning for coherent default theories under WFS_4 is Σ_3^P complete.

Proof:
By Propositions 3.1 and 4.2, for a given coherent default theory T and formula ϕ, checking whether $\phi \in WFS_4(T)$ amounts to check whether $\phi \in Safe(T)$. Therefore it suffices to prove the Σ_3^P-completeness of the latter problem.

Hardness. The following problem $\mathcal{P}$ is well-known to be Σ_3^P complete [21]: Given a quantified propositional formula F of the form

$$\exists p_1, \ldots p_n\, \forall q_1 \ldots q_m,\ \exists r_1, \ldots, r_k\ \phi,$$

where ϕ is a propositional formula on propositional variables $p_1, \ldots, p_n$, $q_1, \ldots, q_m$, and $r_1, \ldots, r_k$, determine whether F is valid. F is valid iff there exists a truth value assignment ν to the propositional variables $p_1, \ldots, p_n$ such that for each extension ν' of ν covering also the propositional variables $q_1, \ldots, q_m$, ν' can be extended to a truth value assignment ν^* covering also the propositional variables $r_1, \ldots, r_k$, such that ν^* satisfies ϕ.

We transform $\mathcal{P}$ into a skeptical reasoning problem under WFS_4. Let $T = (D, W)$ be the default theory defined by $W = \emptyset$ and

$$\begin{aligned} D = \{\ & :p_1/a_1,\ :\neg p_1/a_1,\ \ldots,\ :p_n/a_n,\ :\neg p_n/a_n, \\ & :q_1/q_1,\ :\neg q_1/\neg q_1,\ \ldots,\ :q_m/q_m,\ :\neg q_m/\neg q_m, \\ & a_1 \wedge \cdots \wedge a_n :\phi \wedge b/\phi \wedge b\ \} \end{aligned}$$

where $a_1, \ldots, a_i$ and b are new propositional variables.

Observe that T has exactly 2^m Reiter extensions corresponding to the different choices between the conflicting defaults $:q_i/q_i$ and $\neg q_i/\neg q_i$, for $1 \leq i \leq m$. For each extension of T, at least one default deriving a_j fires for each $1 \leq j \leq n$. Thus each of the 2^m extensions of T contains the set $\{a_1, \ldots, a_n\}$ and thus the premise of the default $a_1 \wedge \cdots \wedge a_n : \phi \wedge b/\phi \wedge b$. Whether this default fires in a given extension E depends thus solely on ϕ. In case ϕ is consistent with the chosen q_i or $\neg q_i$, the default fires, otherwise it doesn't.

We claim that $\phi \wedge b \in Safe(T)$ iff F is valid. Let us sketch the proof of this claim.

If. Assume F is valid. Let ν be a truth value assignment to $p_1, \ldots, p_n$ as described above. Let N be the set of the n defaults among $\{:p_1/a_1,\ :\neg p_1/a_1,\ \ldots,\ :p_n/a_n,\ :\neg p_n/a_n\}$ compatible with ν, i.e., if $\nu(p_i) = \mathit{true}$ then N contains $:p_i/a_i$, otherwise N contains $:\neg p_i/a_i$, for $1 \leq i \leq n$. Let $P = N \cup \{a_1 \wedge \cdots \wedge a_n : \phi \wedge b/\phi \wedge b\}$. From the validity of F it follows that P is a default proof of $\phi \wedge b$ valid in all 2^m extensions of T. Therefore $\phi \wedge b \in Safe(T)$.

Only if. We show that $\phi \wedge b \in Safe(T)$ implies that F is valid. Assume that $\phi \wedge b \in Safe(T)$. This means that $\phi \wedge b$ is contained in all 2^m extensions of T and that there exists a T-default proof P for $\phi \wedge b$ valid in *each* of the 2^m extensions. Since b can only be derived via the default $a_1 \wedge \cdots \wedge a_n : \phi \wedge b/\phi \wedge b$, P must contain this default and therefore also a proof of each a_i for $1 \leq i \leq n$. From this and the validity of P it follows that for each $1 \leq i \leq m$, P must contain exactly one of the two defaults $: p_i/a_i$ and $: \neg p_i/a_i$. Hence P determines a truth value assignment ν to the variables $p_1, \ldots, p_n$. Since P is valid in all 2^m extensions, and each extension corresponds to a different truth value assignment to the variables $q_1, \ldots, q_m$, it holds that each extension ν' of ν that covers the q_i variables is consistent with $\phi \wedge b$, and thus with ϕ. In summary, there exists a truth variable assignment ν to the $p_1, \ldots, p_n$ such that for each extension ν' of ν to the variables $q_1, \ldots, q_m$, ν' can be extended to an assignment ν^* covering also the variables $r_1, \ldots, r_k$ such that ϕ is satisfied by ν^*. Thus F is valid.

This concludes the hardness part of the proof.

Membership. Let $T = (D, W)$ be a coherent default theory. In order to check whether $\phi \in Safe(T)$ one may proceed as follows:

- Guess an ordered set $P \subseteq D$ of defaults;
- For the guessed set P check that

  1. P is a T-default proof for ϕ, and
  2. P is valid in all Reiter extensions of T.

Checking task 1 can be done in polynomial time with queries to an oracle in NP, hence, this task is in Σ_2^P.

It thus remains to show that checking task 2 is solvable in polynomial time with queries to a Σ_2^P oracle. We prove this by showing that the complementary problem *comp*2, i.e., checking whether there exists at least one Reiter extension E of T violating P, is in Σ_2^P. In fact, in order to find such an extension E, one can proceed as follows:

- guess a set E of defaults
- For the guessed set E

  a) check that E is a generating set of defaults (this check is well-known to be feasible with polynomially many calls to an oracle A in NP), and

  b) check (in polynomial time) that at least one default from P does not occur in E, and hence P is not valid w.r.t. the extension generated by E.

Thus checking *comp*2 is solvable in nondeterministic polynomial time with calls to an NP oracle set A, and hence this task lies in Σ_2^P.

In summary, checking whether $\phi \in Safe(T)$ can be done in nondeterministic polynomial time with access to two oracle sets in Σ_2^P, namely one for checking task 1 and one for checking task 2. Since every NP oracle Turing machine querying two Σ_2^P oracle sets can be easily transformed into an equivalent NP Turing machine accessing a single Σ_2^P oracle set, checking whether $\phi \in Safe(T)$ is feasible in nondeterministic polynomial time with access to an oracle in Σ_2^P and is thus in Σ_3^P.

The determination of the precise complexity of skeptical reasoning in WFS_2, WFS_3 both in the general case and in case of normal default theories and of WFS_4 in case of normal defaults will be carried out elsewhere.

6. Application to logic programming

Interestingly, our investigation of different well-founded semantics for default logic also leads to new semantics for normal and extended logic programs. Let us first consider normal programs. They consist of rules of the form

$$c \leftarrow a_1, \ldots a_m, not\ b_1, \ldots, not\ b_n$$

where c, a_i, b_j are atoms and *not* represents negation as failure. By the standard translation this rule can be viewed as the default

$$a_1 \wedge \ldots \wedge a_m : \neg b_1, \ldots, \neg b_n / c$$

that is, normal logic programs can be viewed as special default theories where $W = \emptyset$ and D is obtained by translating each program rule in the manner just described. The standard semantics for normal logic programs is the stable model semantics of Gelfond and Lifschitz [7]. It is well-known that the stable models of a program and the extensions of its default logic translation coincide: the stable models of a program are just the atoms true in the corresponding extensions.

Well-founded semantics for normal logic programs can be viewed as an approximation to stable model semantics. As was mentioned earlier, Baral and Subrahmanian have demonstrated that the original three-valued formulation of well-founded semantics for normal logic programs can be restated in terms of Γ^2. From the results of the last section it follows immediately that our new semantics WFS_i, $i \in \{2, 3, 4\}$, for default logic provide - via the standard translation - semantics for normal logic programs which are at least as strong as WFS. We show that even in the case of normal logic programs the new semantics are stronger, i.e. produce more conclusions in certain cases.

For a program P we will use $WFS_i(P)$ as abbreviation for $WFS_i(trans(P))$, where $trans(P)$ is the standard translation of P to default logic. To demonstrate our claim we use an example similar to the one in the last section showing the differences between WFS_2, WFS_3 and WFS_4. Consider the following normal logic program P:

1) $a \leftarrow not\ d$	3) $c \leftarrow not\ b$	5) $b \leftarrow not\ b, d$
2) $d \leftarrow not\ a$	4) $f \leftarrow not\ d$	

This program has a single stable model, namely $\{a, c, f\}$. The original well-founded semantics leaves all atoms undefined. We obtain $WFS(P) = WFS_2(P) = Th(\emptyset)$. However, $WFS_3(P) = Th(\{c\})$. The reason is that $\Gamma_3(\emptyset)$ does not contain b since there is no justifiable default proof for this atom: rule 5), the only rule with head b, is "self-defeating" and cannot appear in a justifiable proof.

We obtain even more conclusions in WFS_4. The only atoms with extendable default proofs are a, c and f. Therefore $WFS_4(P) = \Gamma(Th(\{a, c, f\})) = Th(\{a, c, f\})$.

This shows that even for normal logic programs some instances of our framework yield more conclusions and thus provide a better approximation to stable model semantics than the original well-founded semantics.

Extended logic programs consist of rules which are similar to those of normal logic programs, except that arbitrary literals may appear instead of atoms. Such programs thus have two types of negation, the weak negation *not* and the classical negation $\neg$. Extended logic programs turn out to be particularly useful for knowledge representation [1].

Well-founded semantics for extended logic programs based on Γ^2 have been proposed by Przymusinski [17]. An alternative, somewhat stronger approach was developed by Pereira and Alferes [15], the semantics WFSX. This semantics implements the intuition that a weakly negated literal should be implied by the corresponding classically negated literal. The authors call this the coherence principle. To satisfy the principle they use the seminormal version of a program P, denoted $S(P)$, which is obtained from P by replacing each rule

$$c \leftarrow a_1, \ldots, a_n, not\ b_1, \ldots, not\ b_m$$

by the rule

$$c \leftarrow a_1, \ldots, a_n, not\ b_1, \ldots, not\ b_m, not\ -c$$

where $-c$ is the complement of c, i.e. $\neg c$ if c is an atom and a if $c = \neg a$. Based on this notion Pereira and Alferes consider the following monotone operator:

$$\Omega_P(X) = \Gamma_P \Gamma^\star_{S(P)}(X)$$

where $\Gamma^\star$ is like Γ but does not require logical closedness. As to be expected the conclusions of a program P under $WFSX$, denoted $WFSX(P)$, are defined as the lfp of Ω_P.

The use of the seminormal version of the program when $\Gamma^\star$ is applied guarantees that a literal l is not considered a potential conclusion whenever the complementary literal is already known to be true. Our notion of an S-consistent default proof underlying the definition of Γ_2 guarantees this property. The proof of the following proposition is thus staightforward:

Proposition 6.1. Let P be an extended logic program. We have $WFSX(P) \subseteq WFS_2(P)$.

In fact, it turns out that WFS_2 is actually stronger than WFSX as illustrated in the following example:

1) $c \leftarrow not\ a$	3) $b \leftarrow not\ \neg b$
2) $a \leftarrow b, \neg b$	4) $\neg b \leftarrow not\ b$

WFSX considers a as a potential conclusion and for this reason does not conclude c. WFS_2, on the other hand, derives c since a has no $\emptyset$-consistent proof and is thus not considered a potential conclusion. It appears that Ω-semantics eliminates some of the S-inconsistent proofs but not all of them. Note that c is true in all answer sets of the program [8].

7. Adding Priorities

Priorities play an important role in many applications of nonmonotonic reasoning. For instance, one often wants to give more specific defaults preference over more general defaults. In configuration tasks design goals are often represented as defeasible constraints and it is convenient to rank these constraints using preferences among them. Also in model based diagnosis preferences can play an important role: preferences among normalcy assumptions can considerably reduce the number of produced diagnoses, eliminating the less plausible ones. We will therefore show in this section how preferences can be taken into account in our framework.

A prioritized default theory is a triple $(D, W, <)$ where (D, W) is a default theory and $<$ a strict partial order on D. Intuitively, $r < r'$ says that r is preferred over r'. For each semantics WFS_i we will now define a corresponding prioritized semantics WFS_i^{pr} that reduces to WFS_i whenever $<$ is empty and produces more conclusions otherwise. Again, the prioritized semantics is defined as the lfp of a certain monotonic operator Ψ_i. This time the operator is defined directly and not composed of two anti-monotone operators.

Definition 7.1. Let T be a prioritized default theory, S a set of formulas, and $r = a : b_1, \ldots, b_n/c \in D$. We say r is (S, i)-applicable iff every Γ_i-default proof for $\neg b_j$ ($j \in \{1, \ldots, n\}$) from D^S contains a default $r' = d : e_1, \ldots, e_m/f$ such that

1. $r < r'$, and
2. there is a default proof for $\neg e_k$ from $D^{\Gamma_i(S)} \cup \{r\}$ for some $k \in \{1, \ldots, m\}$.

Note that by Γ_i-default proof we mean the type of proof admitted in the definition of Γ_i, that is, arbitrary default proof for $i = 1$, S-consistent default proof for $i = 2$ etc.

Definition 7.2. Let T be a prioritized default theory, S a set of formulas. We define

$$\Psi_i(S) = \{p \mid p \text{ has a default proof from } (S, i)\text{-applicable rules in } D\}$$

We first observe that every default in $D^{\Gamma_i(S)}$ is trivially (S, i)-applicable since for these defaults no Γ_i-default proof from D^S for the negation of a justification exists. Also, whenever $<$ is empty we get back our original semantics WFS_i since in that case the (S, i)-applicable defaults coincide with $D^{\Gamma_i(S)}$. The reason is that no other default can satisfy the first condition of Def. 7.1. In the general case, however, more than the defaults in $D^{\Gamma_i(S)}$ will be (S, i)-applicable and we obtain more conclusions. Here is an example:

1) p	3) $p : \neg f/\neg f$
2) b	4) $b : f/f$

WFS_i for $i \in \{1, \ldots, 4\}$ equals $Th(W)$. Assume we have $3 < 4$ and consider $WFS^{pr} = WFS_1^{pr}$. We first have to determine the $(S, 1)$-applicable default rules for $S = \emptyset$. The only default proof for f from D contains rule 4) which is known to be less preferred than 3). Moreover, rule 3) yields a default proof for $\neg f$, a formula defeating 4). Therefore 3) is $(\emptyset, 1)$-applicable and $\Psi_1(\emptyset) = Th(\{p, b, \neg f\})$. This is also the lfp of Ψ_1. Note that the same lfp is obtained for $\Psi_2, \ldots, \Psi_4$.

Condition 2 in Def. 7.1 is necessary to guarantee correctness of the prioritized approach wrt. credulous conclusions. The following example illustrates this:

1) $: a \wedge \neg b/a$
2) $: b/b$

Assume we have $1 < 2$. If condition 2 were omitted, rule 1) would be considered $(\emptyset, i)$-applicable. The lfp would thus contain a which is not a credulous conclusion. Condition 2 guarantees that rule 1) is not $(\emptyset, i)$-applicable and the right lfp is obtained, namely $Th(\{b\})$.

Instead of using a predefined ordering $<$ on defaults it is also possible to represent preference information in the logical language. This makes it possible to reason about preferences dynamically. This has been worked out for the case of extended logic programs with two types of negation in a related paper [4].

8. Discussion

We presented in this paper a range of semantics for default logic. All semantics are correct wrt. safe conclusions. They are based on a monotone operator. The conclusions are defined in terms of the lfp of this operator. The semantics are thus similar in spirit to Baral/Subrahmanian's well-founded and Przymusinska/Przymusinski's stationary semantics yet they produce more conclusions. In particular, they do not break down to monotonic reasoning whenever the set of potential conclusions possessing a default proof from $D^{Th(T)}$ is inconsistent.

Via the standard translation from logic programs to default logic our approach also defines new semantics for normal and extended logic programs. As we saw these new semantics are stronger than well-founded semantics and WFSX for logic programs. Finally, we showed how prioritized versions of all semantics in the range (including original well-founded semantics) can be defined.

Unfortunately, our complexity analysis revealed that - contrary to the situation in logic programming - our new semantics do not lead to a gain in complexity: all important reasoning tasks are hard for the second level of the polynomial hierarchy, i.e., as least as hard as the corresponding reasoning tasks in Reiter's default logic. In the case of WFS_4, reasoning is complete for the class Σ_3^P, and thus at the third level of the Polynomial Hierarchy. As to be expected, the more accurate computation of potential conclusions needs more time resources. This certainly is somewhat disappointing from the perspective of approximations. Nevertheless, we think well-founded default logics are of their own interest because of the type of conclusions they sanction. In situations where security matters safe conclusions may be exactly what one should look for. Consider the task of engineering a bridge. Assume two engineers come to the conclusion that the bridge will carry the intended load, but their conclusion is based on entirely different and incompatible reasons. Then it is probably inadequate to simply rely on the two opinions, even if they agree. In other words, in such a situation it may be reasonable to consider conclusions of well-founded default logic rather than those obtained in Reiter's system.

Acknowledgements

The authors wish to thank H. Tompits and M. Truszczynski for helpful comments.

References

[1] Baral, C. and Gelfond, M.: "Logic Programming and Knowledge Representation", *Journal of Logic Programming* **19,20**, 1994, 73–148.

[2] Baral, C. and Subrahmanian, V.S.: "Duality between Alternative Semantics of Logic Programs and Nonmonotonic Formalisms", *Proc. 1st Intl. Workshop on Logic Programming and Nonmonotonic Reasoning*, Washington, D.C., MIT Press, 1991, 69–86.

[3] Brewka, G.: Cumulative Default Logic: "In Defense of Nonmonotonic Inference Rules", *Artificial Intelligence* **50**, 1991, 183-205.

[4] Brewka, G.: "Well-Founded Semantics for Extended Logic Programs with Dynamic Preference Information", *Journal of Artificial Intelligence Research* **4**, 1996, 19–36.

[5] Doyle, J.: "A Truth Maintenance System", *Artificial Intelligence* **12(3)**, 1979, 231–272.

[6] van Gelder, A., Ross, K. and Schlipf, J.: "The Well-Founded Semantics for General Logic Programs", *Journal of ACM*, 1990, 221–230.

[7] Gelfond, M. and Lifschitz, V.: "The Stable Models Semantics for Logic Programming", *Proc. 5th Intl. Conference on Logic Programming*, Seattle, 1988.

[8] Gelfond, M. and Lifschitz, V.: "Logic Programs with Classical Negation", *Proc. 7th Intl. Conference on Logic Programming*, 1990.

[9] Gottlob, G.: "Complexity Results for Nonmonotonic Logics", *Journal of Logic and Computation*, **2**, No. 3, 1992, 397–425.

[10] Gottlob, G.: "The Complexity of Default Reasoning under the Stationary Fixed Point Semantics", *Information and Computation*, **121**, No. 1, 1995, 81–92.

[11] Johnson, D.S.: "A Catalog of Complexity Classes", In vol. A of *Handbook of Theoretical Computer Science*, Elsevier Science Publishers, North Holland, 1990.

[12] Marek, W. and Truszczynski, M.: *Nonmonotonic Logic: Context-Dependent Reasoning*, Springer Verlag, 1993.

[13] Moore, R.C.: "Semantical Considerations on Nonmonotonic Logic", *Artificial Intelligence*, **59**(1-2), 1985, 75–94.

[14] Papadimitriou, CH. and Sideri, M.: "On Finding Extensions of Default Theories", *Proceedings of the 4th International Conference on Database Theory (ICDT'92)*, Berlin, Germany, 1992, Springer LNCS No.646, 1992, 276–281.

[15] Pereira, L.M. and Alferes, J.J.: "Well-Founded Semantics for Logic Programs with Explicit Negation", *Proc. 10th European Conference on Artificial Intelligence*, Vienna, 1992.

[16] Przymusinski, T.: "The Well-Founded Semantics Coincides with the Three-Valued Stable Semantics", *Fundamenta Informaticae* **13**(4), 1990,445–464.

[17] Przymusinski, T.: "Stable Semantics for Disjunctive Programs", *New Generation Computing*, **9**, 1991, 401–424.

[18] Przymusinska, H. and Przymusinski, T.: "Stationary Default Extensions", *Fundamenta Informaticae* **21**(1,2), 1994, 76–87.

[19] Reiter, R.: "A Logic for Default Reasoning", *Artificial Intelligence* **13**, 1980, 81–132.

[20] Stillman, J.: "The Complexity of Propositional Default Logic", *Proceedings AAAI-92*, 1992.

[21] Stockmeyer, L. and Meyer, A.: "Word Problems Requiring Exponential Time", In *Conference Record of the Fifth Annual Symp. on Theory of Computing*, ACM, New York, 1973, 1–9.

[22] Tarski, A.: "A lattice-theoretical fixpoint theorem and its applications", *Pacific Journal of Mathematics* **5**, 1955, 285–309.