Query-answering in Reiter's Default Logic
(Extended Abstract)

Thomas Linke
Institut für Informatik,
Universität Potsdam
Postfach 60 15 53, D–14415 Potsdam, Germany
linke@cs.uni-potsdam.de

Torsten Schaub
Institut für Informatik,
Universität Potsdam
Postfach 60 15 53, D–14415 Potsdam, Germany
torsten@cs.uni-potsdam.de

We introduce new concepts for default reasoning in the context of query-answering in regular default logic. For this purpose, we develop a proof-oriented approach for deciding whether a default theory has an extension containing a given query. The inherent problem in Reiter's default logic is that it necessitates the inspection of all default rules for answering no matter what query. Also, default theories are known to lack extensions occasionally. We address these two problems by sloting in a compilation phase before the actual query-answering phase. The examination of the entire set of default rules is then done only once in the compilation phase; this allows us to inspect only the ultimately necessary default rules during the actual query answering phase. In fact, the latter inspection must not only account for the derivability of the query, but moreover it must guarantee the existence of an encompassing extension. We address this traditionally important problem by furnishing novel criteria guaranteeing the existence of extensions that are arguably simpler and go well beyond existing approaches.

1 Introduction

In many AI applications default reasoning plays an important role since many subtasks involve reasoning from incomplete information. This is why there is a great need for corresponding implementations that allow us to integrate default reasoning capabilities into complex AI systems. For addressing this problem, we have chosen Reiter's *default logic* [11] as the point of departure: Default logic augments classical logic by *default rules* that differ from standard inference rules in sanctioning inferences that rely upon given as well as absent information. Knowledge is represented in default logic by *default theories* (D, W) consisting of a set of formulas W and a set of default rules D. A default rule $\frac{\alpha:\beta}{\gamma}$ has two types of antecedents: A *prerequisite* α which is established if α is derivable and a *justification* β which is established if β is consistent in a certain way. If both conditions hold, the *consequent* γ is concluded by default. A set of such conclusions (sanctioned by default rules and classical logic) is called an *extension* of an initial set of facts: Given a set of formulas W and a set of default rules D, any such extension E is a deductively closed set of formulas containing W such that, for any $\frac{\alpha:\beta}{\gamma} \in D$, if $\alpha \in E$ and $\neg\beta \notin E$ then $\gamma \in E$. (A formal introduction to default logic is given in Section 2.)

In what follows, we are interested in the basic approach to query-answering in default logic that allows for determining whether a formula is in *some* extension of a given default theory. The inherent problem with this in regular default logic is that it necessitates the inspection of all default rules, no matter which query is posed. This is related to the fact that default theories may not possess extensions at all. Hence, query-answering has so far only been addressed indirectly: On the one hand, we find approaches that are primarily interested in the construction of extensions; queries are then either answerable by simple membership tests or by directing the construction towards extensions containing the query [7, 14, 9, 3]. On the other hand, we find approaches based on variants of default logic that guarantee the existence of extensions and that allow for local proof procedures [12]. Unfortunately, these variants do not offer the same expressiveness as the original approach [1]. In the general case, there is thus no way of avoiding exhaustive computations while preserving the expressiveness of Reiter's default logic. Our key to this problem is given by the discovery that, for query-answering, the inspection of the entire default theory must only be done once, no matter which and how many queries are posed subsequently. This leads us to the following idea: We slot in a compilation phase before the actual query-answering process so that afterwards queries are answerable in a rather local fashion, insofar that the resulting procedures must examine only those default rules necessary for proving the query.

Consider an example where birds fly, birds have wings, penguins are birds, and penguins don't fly along with a formalization through general default theory

$$\left(\left\{\frac{b:\neg ab_b}{f},\frac{b:w}{w},\frac{p:b}{b},\frac{p:\neg ab_p}{\neg f}\right\},\{\neg f\to ab_b, f\to ab_p, p\}\right) \quad (1)$$

(For short we denote the default rules in (1) by $\delta_1, \delta_2, \delta_3, \delta_4$, resp.) An analysis of these rules should provide us with the information that the application of the first rule depends on the blockage of the last one (and vice versa), while the second and third rule can be applied no matter which of the other rules apply. We may thus derive $\neg f$ by ignoring the second and the third rule, while assuring that the first one is blocked. Notably, our initial analysis must be truly global and also extend to putatively unrelated parts of the theory. To see this, simply add the rule $\frac{:x}{\neg x}$, destroying all previous extensions consistent with x. Now, the application of each rule in (1) depends additionally on the blockage of $\frac{:x}{\neg x}$. Thus, given p, we can only apply $\frac{p:\neg ab_p}{\neg f}$ to derive $\neg f$, if $\frac{b:\neg ab_b}{f}$ and $\frac{:x}{\neg x}$ are blocked.

We note that the application of rules depends on the blockage of other rules. The outcome of our analysis must thus provide information on which rules may block other rules (or even themselves). But although we may allot somehow unlimited time to a compilation phase, we cannot provide unlimited space for its result. Our approach complies with this principle and offers as a result a so-called *block graph* whose size is quadratic in the number of defaults, although its computation may need exponential time in the worst case.[1] The block graph represents the essential information about blockage between default rules. In the query-answering phase this information is then used to focus the computation of *default proofs* on ultimately necessary defaults only. For instance, given the block graph of Theory (1), we may derive f from $(D, W \cup \{b\}$ by means of $\frac{b:\neg ab_b}{f}$. While this involves testifying the blockage of $\frac{p:\neg ab_p}{\neg f}$, the two other rules may be completely ignored. This is what makes our approach different from traditional, extension-oriented ones: Once we have compiled the blocking information, subsequent query-answering must only consider default rules *by need*.

In addition to the rules needed for deriving a query, like $\frac{b:\neg ab_b}{f}$ for f, however, this involves considering furthermore those rules threatening the rules in the proof, like $\frac{p:\neg ab_p}{\neg f}$, or even those menacing its encompassing extension, like $\frac{:x}{\neg x}$. Both type of rules are identifiable by means of the block graph. The first ones are necessarily among the predecessors (in the block graph) of the rules used for deriving the query. For guaranteeing an encompassing extension, we propose a range of novel criteria that can be directly read off the block graph, too. In fact, these criteria are arguably simpler and go well beyond existing approaches addressing the traditionally important problem of existence of extensions. This is formally proven in the full version of this paper.

The rest of the paper is organized as follows. In Section 2 we provide a formal introduction to default logic. Section 3 introduces blocking (supporting) sets and the block graph of a default theory, which actually is the result of the compilation phase. Section 5 develops a new representation of extensions, which relies on blocking sets and is suitable for query-answering. It also allows for defining local default proofs for different kinds of default theories in Section 6. Section 7 grasps the presented approach in its entirety and discusses its modularity.

2 Background

We start by completing our initial introduction to Reiter's default logic: A *default rule* $\frac{\alpha:\beta}{\gamma}$ is called *normal* if β is equivalent to γ; it is called *semi-normal* if β implies γ. We sometimes denote the *prerequisite* α of a default rule δ by $Pre(\delta)$, its *justification* β by $Just(\delta)$ and its *consequent* γ by $Cons(\delta)$.[2] A set of default rules D and a set of formulas W form a *default theory*[3] $\Delta = (D, W)$, that may induce one, multiple or even no *extensions* [11]:

Definition 2.1 *Let (D, W) be a default theory. For any set of formulas S, let $\Gamma(S)$ be the smallest set of formulas S' such that*

E1 $W \subseteq S'$,

E2 $Th(S') = S'$,

E3 *For any* $\frac{\alpha:\beta}{\gamma} \in D$, *if* $\alpha \in S'$ *and* $\neg\beta \notin S$ *then* $\gamma \in S'$.

A set of formulas E is an extension of (D, W) iff $\Gamma(E) = E$.

Any such extension represents a possible set of beliefs about the world. For example, Default theory (1) has two extensions: $Th(W \cup \{b, w, \neg f\})$ and $Th(W \cup \{b, w, f\})$, while theory $(D \cup \{\frac{:x}{\neg x}\}, W)$ (where D and W are taken as in (1)) has no extension.

For a set of formulas S and a set of defaults D, we define the *set of generating default rules* as $GD(D, S) = \{\delta \in D \mid S \vdash Pre(\delta) \text{ and } S \not\vdash \neg Just(\delta)\}$. The two last extensions are generated by $\{\delta_2, \delta_3, \delta_4\}$ and $\{\delta_1, \delta_2, \delta_3\}$, respectively. We call a set of default rules D *grounded* in

[1] The reader is reminded that query-answering in default logic has even two distinct sources of exponential complexity; it is Σ_2^P-complete [6].

[2] This notation generalizes to sets of default rules in the obvious way.

[3] If clear from the context, we sometimes refer with Δ to D and W (and vice versa) without mention.

a set of formulas W iff there exists an enumeration $\langle \delta_i \rangle_{i \in I}$ of D such that for all $i \in I$, $W \cup \mathit{Cons}(\{\delta_0, \ldots, \delta_{i-1}\}) \vdash \mathit{Pre}(\delta_i)$. Note that $E = \mathit{Th}(W \cup \mathit{Cons}(\mathit{GD}(D, E)))$ forms an extension of (D, W) if $\mathit{GD}(D, E)$ is grounded in W.

For simplicity, we assume for the rest of the paper that default theories (D, W) comprise finite sets only. Additionally, we assume that for each default rule δ in D, we have that $W \cup \mathit{Just}(\delta)$ is consistent. This can be done without loss of generality, because we can clearly eliminate all rules δ' from D for which $W \cup \mathit{Just}(\delta')$ is inconsistent, without altering the set of extensions.

3 Compressing Blocking Relations by Block Graphs

Our approach is founded on the concept of *blocking sets*: Given a default theory (D, W) and a default rule $\delta \in D$, intuitively, a *blocking set* for δ is a minimal set of default rules $B \subseteq D$ such that the joint application of its rules denies the application of δ. Such a blocking set provides a *candidate* for disabling the putatively applicable default rule δ. For this purpose, it is actually sufficient to refute a rule's justification, ignoring its prerequisite. This is because an existing derivation of a prerequisite can only be counterbalanced by refuting the justification of one of its default rules. This motivates Condition **BS1** in the formal definition of blocking sets, given below.

In order to become effective, a blocking set must belong to the generating default rules of an extension. Thus, it must be grounded and the respective justifications must be consistent with the extension. Groundedness is indispensable and also easily verifiable, leading to Condition **BS2**. Global consistency however is context-dependent since it makes reference to (possible) extensions encompassing the blocking set. In fact, in many cases, we are rather interested in showing that a critical blocking set *does not* contribute to a certain extension, either because it threatens a default proof at hand or even because it menaces the extension as such. Therefore, this (context-dependent) consistency condition must be finally addressed in the context of the respective query-answering process (see Section 6), so that we confine ourselves to a rather local notion of consistency reflected by Condition **BS3**.

This leads us to the following definition of (putative) blocking sets:

Definition 3.1 *Let $\Delta = (D, W)$ be a default theory. For $\delta \in D$, we define the set $\mathcal{B}_\Delta(\delta)$ of all blocking sets for δ in Δ as follows.*

If $B \subseteq D$, then $B \in \mathcal{B}_\Delta(\delta)$ iff B is a minimal set such that

BS1 $W \cup \mathit{Cons}(B) \vdash \neg \mathit{Just}(\delta)$,

BS2 *B is grounded in W,*

BS3 $W \cup \mathit{Cons}(B \setminus \{\delta\}) \not\vdash \neg \mathit{Just}(\delta)$ *for all $\delta \in B$.*

Note that the set of consequences, $\mathit{Cons}(B)$, is not required to be consistent. This is needed, for instance, to detect groups of default rules whose joint application blocks any other default, like $\left\{\frac{:a}{c}, \frac{:b}{\neg c}\right\}$.

The problem of deciding whether a set B satisfies **BS1** and **BS2** is co-NP-complete [13]. **BS3** is a standard NP-complete problem. Notably, the subset-minimality of B can be checked by appeal to **BS1** and **BS2** only (by monotonicity of $\vdash$); minimality is thus confirmable by confuting **BS1** and **BS2** for $B \setminus \{\delta\}$ for all $\delta \in B$. The set of all blocking sets for a(ll) default rule(s) can thus be computed with an algorithm iterating over 2^D with appeal to an oracle to a problem in NP $\cup$ co-NP.

We show in the full paper that in the worst case, a theory with n rules may comprise $O(2^n)$ blocking sets. However, the number of blocking sets is not related to the number of extensions of a given theory. To see this, observe that Theory $\left(\left\{\frac{:a_i}{\neg c_i}, \frac{:c_i}{\neg a_i} \mid i = 1..n\right\}, \emptyset\right)$ has 2^n extensions but only $2n$ blocking sets. That is, although we encounter an exponential number of extensions, we have only a linear number of blocking sets.

For further illustration, consider Theory (1) along with its blocking sets given in (2)–(5):

$$\mathcal{B}_\Delta(\delta_1) = \{\{\delta_4\}\} \tag{2}$$
$$\mathcal{B}_\Delta(\delta_2) = \emptyset \tag{3}$$
$$\mathcal{B}_\Delta(\delta_3) = \emptyset \tag{4}$$
$$\mathcal{B}_\Delta(\delta_4) = \{\{\delta_1, \delta_3\}\} \tag{5}$$

For example, $\{\delta_4\}$ is the only blocking set for δ_1, because it comprises a possible refutation of ab_b, the justification of δ_1. In general, a single default rule may have multiple blocking sets. For example, adding $\frac{:u}{v \wedge \neg v}$ to Theory (1) augments each set $\mathcal{B}_\Delta(\delta_i)$ by $\left\{\frac{:u}{v \wedge \neg v}\right\}$. The addition of $\frac{:x}{\neg x}$ to (1) leaves blocking sets (2)–(5) unaffected and yields $\mathcal{B}_\Delta(\frac{:x}{\neg x}) = \{\{\frac{:x}{\neg x}\}\}$ reflecting self blockage.

Observe that **BS3** allows us to discard blocking sets that block their own constituent rules. For instance, $\{\delta_1, \delta_3, \delta_4\}$ is a putative blocking set of δ_2; it is ruled out by **BS3** since it contains both δ_1 and one of its blocking sets, $\{\delta_4\}$.

Now, given the concept of blocking sets, we are ready to define the outcome of our compilation phase: The *block graph* of a default theory.

Definition 3.2 *Let $\Delta = (D, W)$ be a default theory. The block graph $\Gamma_\Delta = (V_\Delta, A_\Delta)$ of Δ is a directed graph with* *vertices* $V_\Delta = D$ *and arcs*

$$A_\Delta = \{(\delta, \delta') \mid \text{there is some } B \in \mathcal{B}_\Delta(\delta') \text{ with } \delta \in B\}.$$

So, there is an arc (δ', δ) between default rules δ' and δ in the block graph iff δ' belongs to some blocking set for δ. For Default theory (1), we obtain the block graph given in Figure 1; it has arcs (δ_4, δ_1), (δ_1, δ_4) and (δ_3, δ_4).

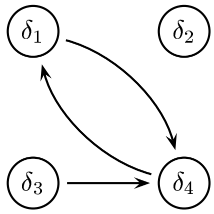

Figure 1: Block graph of Default theory (1).

We observe that the size of the block graph is quadratic in the number of default rules, although there may be an exponential number of blocking sets. The block graph thus comprises the essential information from the blocking sets; this is accomplished by abstracting from the membership of default rules in specific blocking sets. It is actually unnecessary to keep any (possibly exponential number of) blocking sets beyond the compilation phase, because they can be effectively reconstructed from Γ_Δ during the query-answering phase.

Notably, we do not even have to keep or recompute blocking sets during the compilation phase itself, since the block graph is computable incrementally:

Lemma 3.1 *Let* $\Delta = (D, W)$ *and* $\Delta' = (D', W)$ *be default theories with* $D' \subseteq D$.

If $\delta \in D'$, *then* $B \in \mathcal{B}_{\Delta'}(\delta)$ *implies* $B \in \mathcal{B}_\Delta(\delta)$.

The usage of the block graph is further detailed in the following sections.

A key issue during query-answering is to cover the safe application of default rules needed for proving a query. For this, we must guarantee that all (possible) blocking sets of such rules are blocked themselves. This leads us to the concept of *supporting sets*, which are intuitively simply blocking sets for blocking sets. First, we extend the notion of blocking sets to sets of rules: For a default theory $\Delta = (D, W)$ and sets $B, B' \subseteq D$, we call B' a *blocking set* for B, written $B' \succ_\Delta B$, if there is some default rule $\delta \in B$ such that $B' \in \mathcal{B}_\Delta(\delta)$.

Definition 3.3 *Let* $\Delta = (D, W)$ *be a default theory. For* $\delta \in D$, *we define the set* $\mathcal{S}_\Delta(\delta)$ *of all supporting sets for* δ *as*

$$\mathcal{S}_\Delta(\delta) = \{B'_1 \cup \ldots \cup B'_n \mid B'_i \subseteq D \text{ such that } B'_i \succ_\Delta B_i \text{ and } \mathcal{B}_\Delta(\delta) = \{B_1, \ldots, B_n\}\}$$

provided $\mathcal{B}_\Delta(\delta) \neq \emptyset$. *Otherwise, we define it as* $\mathcal{S}_\Delta(\delta) = \{\emptyset\}$.

Observe that $\mathcal{S}_\Delta(\delta) = \emptyset$ whenever $\mathcal{B}_\Delta(\delta') = \emptyset$ for all δ' in some $B_i \in \mathcal{B}_\Delta(\delta)$, because then for B_i there is no set of default rules B'_i such that $B'_i \succ_\Delta B_i$, that is, $B'_1 \cup \ldots \cup B'_n$ is undefined.

The purpose of supporting sets is to rule out blocking sets as subsets of the generating default rules: Once a supporting set for δ has been applied (ie. it belongs to the generating default rules), δ itself can be applied safely. The supporting sets in Theory (1) are given in (6)–(9):

$$\mathcal{S}_\Delta(\delta_1) = \{\{\delta_3, \delta_1\}\} \tag{6}$$
$$\mathcal{S}_\Delta(\delta_2) = \{\emptyset\} \tag{7}$$
$$\mathcal{S}_\Delta(\delta_3) = \{\emptyset\} \tag{8}$$
$$\mathcal{S}_\Delta(\delta_4) = \{\{\delta_4\}\} \tag{9}$$

For example, consider the supporting set for δ_1: We have to find *one* blocking set for each blocking set in $\mathcal{B}_\Delta(\delta_1) = \{\{\delta_4\}\}$. In this easy case, we have to find some blocking set for δ_4, yielding $\{\delta_3, \delta_1\}$. Here, $\{\delta_3, \delta_1\}$ is the only supporting set for δ_1. Similarly, for δ_4, we have to find a blocking set B' for $\{\delta_3, \delta_1\}$ (see (5)). That is, we must have $B' \in \mathcal{B}_\Delta(\delta_3)$ or $B' \in \mathcal{B}_\Delta(\delta_1)$. Because $\mathcal{B}_\Delta(\delta_3)$ is empty, we get $\{\delta_4\} \in \mathcal{B}_\Delta(\delta_1)$ as the only supporting set for δ_4. The occurrence of δ_4 in its supporting set is due to the fact that there is a direct conflict between δ_4 and its blocking set. This says that δ_4 is safely applicable on its own. In general this need not be the case. For example, in default theory $(\{\frac{:b}{a}, \frac{:c}{\neg b}, \frac{:d}{\neg c}, \}, \emptyset)$ the last rule forms the single supporting set for the first one.

We discuss further properties of blocking and supporting sets in the full paper.

4 Existence of Extensions

For query-answering, it is clearly important to know whether a default theory has an extension, since reasonable conclusions must reside in such an extension. In fact, determining whether a default theory has an extension or not is on itself a major computational problem, pertinent to default logic. In previous works, broad subclasses of default theories always possessing extensions have been identified, among them we find *normal* [11], *ordered* [5], *ps-even* [10],[4], and *strongly stratified* [2] default theories. In this section, we provide a class of default theories guaranteeing the existence of extensions that are more general

[4] We use ps-even for referring to the notion of *evenness* due to Papadimitriou and Sideri [10].

than the previous ones (see Section 7). The corresponding criteria are thus not only a salient part of our approach to query-answering, but they moreover represent an important contribution on their own.[5]

The basic idea is to further exploit the structure of blocking graphs for providing sufficient conditions on whether the underlying default theory has an extension.

To begin with, we call a default theory (D, W) *non-conflicting*, if it has no blocking sets, that is, if its block graph has no arcs; otherwise we call it *conflicting*. Non-conflicting default theories have unique extensions and trivially allow for query-answering without consistency checks[6].

Lemma 4.1 *Every non-conflicting default theory has a single extension.*

For instance, default theory $(\{\frac{:a}{a}, \frac{:b}{b}\}, \emptyset)$ is non-conflicting, yielding a block graph with no arcs. The same holds for Theory (1), when eliminating either $\frac{b : \neg ab_b}{f}$ or $\frac{p : \neg ab_p}{\neg f}$.

More interestingly, we call a default theory *well-ordered*, if its block graph is acyclic. The next result shows that well-ordered default theories have single extensions.

Theorem 4.2 *Every well-ordered default theory has a single extension.*

For instance, default theory $(\{\frac{:a}{a}, \frac{: b\wedge\neg a}{b}\}, \emptyset)$ is well-ordered; its block graph contains a single arc, indicating that the first rule may block the second one (but *not* vice versa).

We call a default theory *even*, if its block graph contains cycles with even length only.[7] Our first major result of this section states that even default theories always have extensions:

Theorem 4.3 *Every even default theory has an extension.*

For instance, default theory $(\{\frac{: a\wedge\neg b}{a}, \frac{: b\wedge\neg a}{b}\}, \emptyset)$ is even; its block graph contains two arcs, indicating that the first rule may block the second one, *and* vice versa.

Evenness is also enjoyed by our initial default theory in (1), as can be easily verified by regarding its block graph in Figure 1. Unlike this, default theory $(\{\frac{:x}{\neg x}\}, \emptyset)$ is not even, since its block graph contains an odd cycle, namely one of length one.

In fact, the elimination of putative extensions is somehow always due to odd cycles. Not all of them, however, do lead to the destruction of extensions, as can be read off the block graph:[8] A default theory is called *weakly odd* if its block graph contains only odd cycles having two distinct successive nodes u and v such that (u, v) and (v, u) belong to the block graph. Clearly, even default theories are weakly odd.

Our second major result of this section states that weakly odd default theories always have extensions:

Theorem 4.4 *Every weakly odd default theory has an extension.*

For example, the theory consisting of $\frac{: a\wedge\neg b}{a}$, $\frac{: b\wedge\neg c}{b}$, $\frac{: c\wedge\neg a}{c}$, and no facts has no extension; its block graph contains a cycle of length three, say $(\delta_c, \delta_b, \delta_a)$ (where the indexes refer to consequents). Adding formula $c \rightarrow b$ yields actually a weakly odd theory, whose only extension contains c and b. Although the block graph of this theory contains the same odd cycle, it does now contain additionally arc (δ_a, δ_c), which counterbalances the self-blocking-behavior of the odd cycle.

The salient advantage of weak oddness, evenness, and well-orderedness is that these properties are simple, that is, they can be tested in polynomial time, and they rely on a simple data structure, namely a block graph. And even more importantly, they apply to general default theories and are syntax-independent, unlike other approaches [5, 10] that apply to semi-normal default theories only[9] and give different results on equivalent yet syntacticly different theories (see Section 7 for more details). Even though we put forward weak oddness (evenness, or well-orderedness) for testing existence of extensions, it is already worth mentioning that this is no characteristic feature of our approach to query-answering, since it is actually modular in allowing for alternative checks, provided that they are more appropriate.

5 Formal foundations of query-answering

This section lays the formal foundations of our approach to query-answering. We have already seen in the introductory section that we need additional efforts for backing up a default proof. This involves two tasks, namely, protecting the

[5] For continuity, we delay a detailed comparison with the aforementioned approaches to Section 7.

[6] Observe that although non-conflicting default theories ignore justifications, they are still non-monotonic because extensions may be invalidated after augmenting a non-conflicting theory.

[7] The *length* of a directed cycle in a graph is the total number of arcs occurring in the cycle.

[8] The full paper gives moreover a class of odd cycles indicating the non-extension of an extension.

[9] The approaches in [5, 10] do not apply to so-called *weak* semi-normal theories (allowing for default rules without justifications) that have been shown to be equivalent to general default theories in [8].

constituent default rules of a default proof and assuring an encompassing extension.

The first task is accomplishable by blocking and supporting sets:

Definition 5.1 *Let $\Delta = (D, W)$ be a default theory. A set of default rules $D' \subseteq D$ is protected in D iff for each $\delta \in D'$ we have that (i) $S \subseteq D'$ for some $S \in \mathcal{S}_\Delta(\delta)$ and (ii) $B \subseteq D'$ for no $B \in \mathcal{B}_\Delta(\delta)$.*

In words, a set of defaults is protected if it contains some supporting set for each constituent default and if it contains no blocking set for any of its defaults.

The second task is actually more crucial. This is because there are default theories without extensions and, in view of local query-answering, the test for an encompassing extension should be accomplishable without actually computing such an extension. For this purpose, we can build on the criteria developed in the last section, which allow us to guarantee the existence of an extension of a default theory by looking at its block graph.

By combining the two previous tasks, we obtain an alternative characterization of extensions. For this, define for $\Delta = (D, W)$,

$$\Delta \ominus D' \quad \text{as} \quad \big(D \setminus (D' \cup \overline{D'}), W \cup \mathit{Cons}(D')\big) \quad (10)$$

where $D' \subseteq D$ and $\overline{D'} = \{\delta \in D \mid W \cup \mathit{Cons}(D') \vdash \neg \mathit{Just}(\delta)\}$[10]. Then, we have the following result.

Theorem 5.1 *Let $\Delta = (D, W)$ be a default theory and let E be a set of formulas. Then, E is an extension of Δ iff $E = Th(W \cup \mathit{Cons}(D') \cup E')$ for some $D' \subseteq D$ such that (i) D' is grounded in W, (ii) D' is protected in D and (iii) $\Delta \ominus D'$ has extension E'.*

For general default theories $\Delta \ominus D'$, Condition *(iii)* is implementable by weak odd-, even- or well-orderedness. In this case, Theorem 5.1 furnishes a characterization of extensions in terms of blocking and supporting sets (due to Condition *(ii)* and *(iii)*). In case $\Delta \ominus D'$ is semi-normal, the methods in [5, 10] could work just as fine (except for their syntax-dependency and the more complicated data structures needed). The test is even trivial in case $\Delta \ominus D'$ is normal or non-conflicting. This demonstrates the modularity of our approach as regards verifying Condition *(iii)*, that is, existence of extensions of $\Delta \ominus D'$.

For query-answering, it is important to notice that the set D' in Theorem 5.1 needs not to be maximal. In fact, D' can be any grounded subset of the generating default rules $GD(D, E)$ of extension E. This attributes to D' the character of an extension-dependent *default proof*: Given the set of generating default rules $GD(D, E)$ for some extension E, a default proof of some formula φ is simply a grounded set $D' \subseteq GD(D, E)$ such that $W \cup \mathit{Cons}(D') \vdash \varphi$. In such a predetermined setting, we do neither have to care about the consistent application of the rules in D' (this is assured by $D' \subseteq GD(D, E)$) nor (trivially) about the existence of an encompassing extension. Both issues, addressed by *(ii)* and *(iii)* in Theorem 5.1, are however of crucial importance, whenever there is no such extension at hand.

For example, in Default theory (1), the set $D' = \{\delta_3, \delta_1\}$ may serve as a default proof for *f*; it satisfies conditions *(i)* and *(ii)* because it is grounded and protected wrt (1). (The latter is easily verifiable in (2)–(5) and (6)–(9).) For showing that *f* belongs to an existing extension, it is now sufficient to demonstrate that $\Delta \ominus D'$ has an extension, notably *without* computing it. We get a non-conflicting theory

$$\begin{aligned} \Delta \ominus D' &= (D \setminus (\{\delta_3, \delta_1\} \cup \{\delta_4\}), W \cup \{f, b\}) \\ &= (\{\delta_2\}, \{p, ab_p, f, b\}) \end{aligned}$$

which has obviously an extension, due to an empty block graph. Hence, we have shown that *f* is a default conclusion of (1) without computing the corresponding extension. The next corollary makes these ideas precise for query-answering:

Corollary 5.2 *Let $\Delta = (D, W)$ be a default theory and let φ be a formula. Then, $\varphi \in E$ for some extension E of Δ iff $W \cup \mathit{Cons}(D') \vdash \varphi$ for some $D' \subseteq D$ such that (i) D' is grounded in W, (ii) D' is protected in D and (iii) $\Delta \ominus D'$ has an extension.*

Query-answering thus boils down to finding an appropriate set of default rules that is sufficient for deriving the query and for protecting itself against possible threats, provided that its "application" preserves an encompassing extension.

6 Default Proofs for Query-Answering

After the compilation phase, we are able to decide by examining the block graph whether a default theory is non-conflicting, well-ordered, etc. or neither of them. In analogy to these classes of default theories (and for conceptual clarity) we give an incremental definition of a default proof. We start with default proofs for the simplest case of non-conflicting default theories.

Definition 6.1 (Pure default proof [11]) *Let $\Delta = (D, W)$ be a default theory and φ a formula. A set of default rules $P_0 \subseteq D$ is a pure default proof for φ from Δ iff*

[10] The purpose of $\overline{D'}$ is to eliminate defaults with inconsistent justifications in $\Delta \ominus D'$.

[11] Observe that although pure default proofs ignore justifications, they are still non-monotonic because they may be invalidated after augmenting the underlying non-conflicting theory.

P1 $W \cup Cons(P_0) \vdash \varphi$,

P2 P_0 *is grounded in* W.

The problem of deciding whether a set P_0 is a pure default proof is co-NP-complete [13].

Actually, one may wonder whether we must stipulate an additional consistency condition in Definition 6.1, viz.

P3 $W \cup Cons(P_0) \not\vdash \neg Just(\delta)$ for all $\delta \in P_0$,

because this is central for applying default rules. But here, we concentrate on the very simple class of non-conflicting default theories, which ensures that **P3** holds anyway. Otherwise there would be at least one blocking set and the default theory would be conflicting. As a nice consequence, pure default proofs correspond to extension-dependent default proofs but without necessitating such an extension.

For example, let Δ'' be the default theory obtained from the one in (1) by leaving out $\frac{b\,:\,\neg ab_b}{f}$. Then, Δ'' is non-conflicting and $P_0 = \{\frac{p\,:\,\neg ab_p}{\neg f}\}$ is a pure default proof for $\neg f$. This proof is found without consistency checking nor any measures guaranteeing the existence of an encompassing extension.

We have the following result for non-conflicting default theories:

Theorem 6.1 *Let Δ be a non-conflicting default theory and φ a formula. Then $\varphi \in E$ for an extension E of Δ iff there is a pure default proof for φ from Δ.*

In case we have a conflicting yet well-ordered (or even or weakly odd) default theory, we have to take supporting sets into account, because then it is necessary to protect the constituent default rules of a default proof: Whenever we add a default rule to a partially constructed default proof, we have to make sure that there is some supporting set for this default (if necessary). Clearly, this must be done for the default rules in the supporting sets, too. We thus recursively add supporting sets to default proofs: Let $\Delta = (D, W)$ be a default theory and $\delta \in D$ a default rule. A set of default rules $C \subseteq D$ is a *complete support of δ in Δ* iff C is a minimal set such that for each $\delta' \in C \cup \{\delta\}$ there is some supporting set $S' \in \mathcal{S}_\Delta(\delta')$ with $S' \subseteq C$.

This leads us to the concept of supported default proofs:

Definition 6.2 (Supported default proof) *Let $\Delta = (D, W)$ be a default theory and φ a formula. A set of default rules $P \subseteq D$ is a supported default proof for φ from Δ iff $P = P_0 \cup C$ with $C = \bigcup_{\delta \in P_0} C_\delta$ such that*

SP1 *P_0 is a pure default proof for φ from Δ,*

SP2 *C_δ is a complete support of δ for each $\delta \in P_0$,*

SP3 *P contains no blocking set for each $\delta \in P$.*

In the full paper, we show that verifying whether a pure default proof is a supported default proof can be done within Σ_2^P of the polynomial hierarchy.

Condition **SP2** ensures that there is a supporting set for each default in P, if necessary. Condition **SP3** prevents P from blocking some of its own members; thus **SP2** and **SP3** together imply that P is a protected set of default rules. Therefore, **P3** is also valid for supported default proofs, because otherwise there would be a blocking set in P, which is a contradiction to **SP3**. As an important consequence, we do not need any global consistency checks when querying weakly odd (even or well-ordered) default theories.

Similar checks are rather done locally during the determination of supporting sets. For obtaining these sets during the query-answering phase, we can draw on the block graph for restricting the search space. To find out the blocking sets for a default δ, we only have to take the predecessors of δ in the block graph into account, which form usually a rather limited subset of all default rules. The consistent application of δ is then guaranteed by determining one supporting set for δ among a fixed subset of all default rules, namely the pre-predecessors of δ. We may also think of different heuristics for a more elaborated (re)construction of blocking sets, like caching selected blocking sets retained during the compilation phases.

For illustration, let us return to even Default theory (1). We have already seen that $P_0 = \{\delta_4\} = \{\frac{p\,:\,\neg ab_p}{\neg f}\}$ is a pure default proof for $\neg f$, that is, **SP1** holds. Because Theory (1) is conflicting, we have to warrant a complete support for each default in P_0. First of all, it is important to note that our approach leaves plenty of room for accomplishing this task, depending on whether and if so how many blocking sets were maintained from the compilation phase: In case we have kept $\mathcal{S}_\Delta(\delta_4)$, we can directly choose one of its members, yielding $C_{\delta_4} = \{\delta_4\}$. If not, we first verify whether it is actually necessary to generate a support for δ_4 by checking whether δ_4 has predecessors in the block graph. Since this is true in our example (δ_4 has predecessors δ_1 and δ_3), we must (re)generate one member of $\mathcal{S}_\Delta(\delta_4)$. This can be done by looking at the predecessors of δ_1 and δ_3 in the block graph, which immediately yields $\{\delta_4\}$ as the only supporting set; no matter whether or not $\mathcal{B}_\Delta(\delta_1)$ and $\mathcal{B}_\Delta(\delta_3)$ are given explicitly. In any case, we thus get $C = C_{\delta_4} = \{\delta_4\}$, so that we have

$$P = P_0 \cup C = \left\{ \frac{p\,:\,\neg ab_p}{\neg f} \right\}.$$

This establishes Condition **SP2** for P. For verifying **SP3** it is sufficient to observe that no members of P are connected in the block graph. Hence P is a supported default proof for $\neg f$ from Theory (1). Given the block graph, this proof is found without any consistency checks and no measures guaranteeing the existence of an encompassing extension.

Similar arguments show that $\{\frac{p:b}{b}, \frac{b:\neg ab_b}{f}\}$ and $\{\frac{p:b}{b}, \frac{b:w}{w}\}$ are supported default proofs for f and w from Theory (1), respectively. It is important to note that for establishing this, we only had to warrant a support for $\frac{b:\neg ab_b}{f}$, since it is the only one among all involved default rules, having a predecessor in the block graph. As with $\frac{p:\neg ab_p}{\neg f}$ above, however, $\frac{b:\neg ab_b}{f}$ supports itself, so that no other defaults had to be added to the pure default proof obtained in the first case. In the second one, the default proof is even found without any search for supporting sets (nor any consistency check). This illustrates nicely how query-answering draws upon the information furnished by the block graph in order to concentrate efforts on the ultimately necessary defaults only.

As a result, we get that supported default proofs furnish a sound and complete concept for querying weakly odd (or even, etc.) default theories:

Theorem 6.2 *Let $\Delta = (D, W)$ be a weakly odd default theory and φ a formula. Then $\varphi \in E$ for an extension E of Δ iff there is a supported default proof for φ from Δ.*

The stipulation of a weakly odd default theory can be replaced by any other adequate condition guaranteeing the existence of extensions.

In the general case, dealing with arbitrary default theories, we must additionally guarantee that there is an encompassing extension, whose generating default rules comprises a given (supported) default proof. As always, this should be accomplished without actually computing such an extension. For this, we proceed as follows. After computing a supported default proof P from a theory Δ, we check whether $\Delta \ominus P$ has an extension. This is reflected by Condition $\mathbf{DP2}$ of the following definition. The resulting default proof is then called a general default proof:

Definition 6.3 (General default proof) *Let $\Delta = (D, W)$ be a default theory and φ a formula. A (general) default proof for φ from Δ is a set of default rules $P \subseteq D$ such that*

$\mathbf{DP1}$ *P is a supported default proof for φ from Δ and*

$\mathbf{DP2}$ *$\Delta \ominus P$ is weakly odd (even, well-ordered, or non-conflicting).*

Observe that $\mathbf{DP2}$ can be verified by means of the block graph in polynomial time. As above, the stipulation of even- or well-ordered- or even non-conflicting-ness can be replaced by any other condition guaranteeing the existence of extensions. This establishes further evidence of the modularity of our approach.

For illustration, let us render Default theory (1) a non-even, that is an *odd*, theory by adding self-circular default rule $\delta_5 = \frac{:f}{\neg f}$. We refer to the resulting default theory as Δ'. The corresponding block graph is given in Figure 2. This

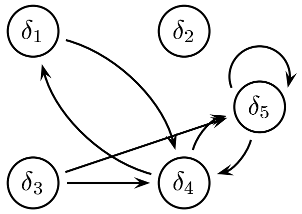

Figure 2: Block graph of Δ'.

makes us add (δ_3, δ_5), (δ_5, δ_5), (δ_4, δ_5) and (δ_5, δ_4) to the block graph in Figure 2; the rest of the graph remains unchanged. Δ' has a single extension: $Th(W \cup \{b, w, \neg f\})$. Now, let us reconsider in turn the previous (supported) default proofs, in the light of this change.

We start with $P = \{\delta_4\} = \{\frac{p:\neg ab_p}{\neg f}\}$. Since the set of predecessors of δ_4 now contains additionally default δ_5, we have to verify former supporting set C_{δ_4}. In fact, $\{\delta_4\}$ forms a blocking set for δ_5 so that C_{δ_4} remains unchanged. Hence, P is still a supported default proof; thus establishing $\mathbf{DP1}$. To verify $\mathbf{DP2}$ for P, we have to construct the block graph $\Gamma_{\Delta' \ominus P}$ of $\Delta' \ominus P$ for verifying its even- or well-orderedness. By the monotonicity property expressed in Lemma 3.1, $\Gamma_{\Delta' \ominus P}$ is obtained from $\Gamma_{\Delta'}$ by simply deleting vertices and arcs. First, we delete in $\Gamma_{\Delta'}$ all defaults in $P = \{\delta_4\}$ along with all adjacent arcs. The same is done with δ_1 and δ_5 because $W \cup Cons(\{\delta_4\}) \vdash \neg Just(\delta_i)$ for $i = 1, 5$ (cf. (10)). As a result, we obtain for $\Gamma_{\Delta' \ominus P}$ a graph with vertices δ_2, δ_3 and no arcs, which implies that $\Delta' \ominus P$ is even non-conflicting. This shows that $\Delta' \ominus P$ has an extension and hence that P is a general default proof for $\neg f$ from Δ'.

Next, consider $P' = \{\delta_3, \delta_1\} = \{\frac{p:b}{b}, \frac{b:\neg ab_b}{f}\}$, the supported proof of f from (1). Unlike above, however, we encounter now an additional predecessor of δ_1, namely, δ_5. Investigating the predecessors of δ_5 yields to possible supporting sets: $\{\delta_3, \delta_1\}$ and $\{\delta_5\}$. The latter one is ruled out immediately, since its addition to P' would violate $\mathbf{SP2}$, so that we are left with $\{\delta_3, \delta_1\}$, and thus P' itself. For addressing $\mathbf{DP2}$, we have to inspect $\Gamma_{\Delta' \ominus P'}$, given in Figure 3. We see that there remains an odd cycle, all of whose nodes have no predecessors outside the cycle. Thus, none of the defaults is blockable, and we obtain no extension. As we show in the full paper, $\Gamma_{\Delta' \ominus P'}$ comprises a so-called

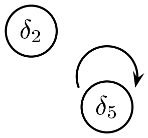

Figure 3: Block graph of $\Delta' \ominus P'$.

active odd cycle, indicating that $\Delta' \ominus P'$ has no extension. As a consequence, P' violates $\mathbf{DP2}$; it is therefore no valid default proof from Δ', as expected in view of Δ''s single extension containing $\neg f$.

Finally, let us consider $P'' = \{\delta_3, \delta_2\} = \{\frac{p\,:\,b}{b}, \frac{b\,:\,w}{w}\}$, the supported proof of w from (1). Although the absence of predecessors for both default rules in $\Gamma_{\Delta'}$ does not urge us to determine (direct) supporting sets for them, we do actually need an indirect support for guaranteeing an encompassing extension. To see this, consider the block graph of $\Delta' \ominus P''$, given in Figure 4. As in Figure 3, we encounter

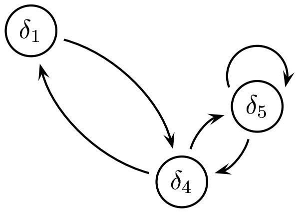

Figure 4: Block graph of $\Delta' \ominus P''$.

an odd cycle in $\Gamma_{\Delta' \ominus P''}$. In contrast to the block graph of $\Delta' \ominus P'$, however, there is now an arc entering the odd cycle, namely (δ_4, δ_5), indicating that δ_5 can be blocked by δ_4. In fact, by adding δ_4 as an indirect support to P'', we can eliminate the odd cycle in the block graph and thus guarantee an encompassing extension. In more detail, we must now verify that $P'' \cup \{\delta_4\}$ satisfies $\mathbf{DP1}$ and $\mathbf{DP2}$. The fact that $P'' \cup \{\delta_4\}$ is a supported default proof is established as shown above. For $\mathbf{DP2}$, we must inspect the block graph of $\Delta' \ominus (P'' \cup \{\delta_4\})$, which turns out to be the empty graph. In all, $P'' \cup \{\delta_4\}$ is thus a valid default proof for w from Δ'.

Condition $\mathbf{DP2}$ is thus verified by first eliminating the defaults involved in the default proof from the block graph and then checking whether the resulting block graph is even, acyclic, or arcless, all of which is doable in polynomial time. In order to apply these criteria, however, we might have to add additional rules eliminating odd cycles.

We have the following result, establishing soundness and completeness in the general case.

Theorem 6.3 *Let $\Delta = (D, W)$ be a (general) default theory and φ a formula. Then $\varphi \in E$ for an extension E of Δ iff there is a (general) default proof for φ from Δ.*

We have seen that the formation of default proofs benefits considerably from the usage of block graphs. As a major consequence, we may restrict our attention to ultimately necessary default rules; thus avoiding the construction of entire extensions. In fact, we have seen that irrelevant default rules, like $\frac{b\,:\,w}{w}$ for answering queries f or $\neg f$, are always ignored, since they are not related to the constituent rules of the proof in the block graph. As a result, all previous default proofs consist of true subsets of the generating defaults. Of course in the worst case, there are default theories and queries, for which we have to consider all generating defaults. But this is then arguably a matter of fact and unavoidable.

7 Discussion

We have introduced new concepts on default logic leading to alternative characterizations of default proofs and their encompassing extensions. The central role among these concepts is played by that of a block graph. First, it allows us to determine the existence of extensions in polynomial time by means of simple graph operations. Second, it tells us exactly which default rules must be considered for applying a default rule. The effort put into the computation of the block graph thus pays off whenever it allows us to validate default proofs by appeal to rather small subsets of the overall set of default rules.

We have presented our concepts in the context of query-answering. The resulting characterization of default proofs for queries is (to the best of our knowledge) the first one that does not appeal to the computation of entire extensions. For this, we have divided query-answering in Reiter's default logic in an off-line and an on-line process: We start with a compilation phase which results in the block graph Γ_Δ of a default theory Δ. Γ_Δ contains the essential information about the blocking relations between the default rules in Δ and is quadratic in the number of defaults. Unlike [7, 14], our approach does thus not suffer from exponential space complexity. The subsequent query-answering phase aims at finding a default proof P such that $\Delta \ominus P$ possesses an extension. This gives us a default proof which contains only the ultimately necessary defaults. To be more precise, a default rule belongs to P only (i) if it contributes to the derivation of the query or if it is needed (iia) for supporting a constituent rule of the proof or (iib) for supporting an encompassing extension. While the former is fixed by the standard inferential relation, the two latter are determined by the block graph. For delineating such a proof, we can draw on Γ_Δ for (re)computing the blocking and supporting sets of its constituent rules. Blocking sets are found among the direct predecessors of a rule, while the search for its supporting sets can be restricted to its pre-predecessors. This is clearly the more efficient the sparser the block graph. A default rule neither belonging to the actual proof nor being related to it via a path in the block graph must thus never be taken into account during query-answering (provided that an encompassing extension exists). This is what makes our approach different from extension-oriented ones [7, 14, 9, 3].

An important underlying problem is that of guaranteeing extensions of Δ or its derivate $\Delta \ominus P$. We addressed this problem by providing a range of criteria, each of which can be read off the block graph in polynomial time. However, our investment into the construction of a block graph does not only provide us with fast checks for encompassing extensions, which is also a great benefit in view of repetitive queries, but it furnishes moreover criteria that go beyond existing ones: We show in the full paper (i) that every default theory satisfying the criterion proposed by Papadimitriou and Sideri in [10] (which is itself a generalization of the one in [4]) is an even theory *but* not vice versa and (ii) that every default theory satisfying the criterion proposed by Cholewiński in [2] is weakly odd *but* not vice versa. Moreover, our criteria are fully syntax-independant which is not the case with any of the other approaches. Also, we can treat general default theories, while [4, 10] apply to semi-normal default theories only[12] and [2] imposes strong conditions on the occorrence of propositional variables. For fairness, we draw the readers attention to the fact that all initial analysis taken by the aforementioned approaches are polynomial, unlike the computation of the block graph.

Finally, it is important to note that our query-answering approach is modular in testing the existence of extensions. Depending on the different kinds of default theories (general, semi-normal or normal), we are able to choose different criteria for this test.

References

[1] G. Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press, Cambridge, 1991.

[2] P. Cholewiński. Reasoning with stratified default theories. In *Proceedings of the Third International Conference on Logic Programming and Nonmonotonic Reasoning*, 1995.

[3] P. Cholewiński, V. Marek, and M. Truszczyński. Default reasoning system DeReS. In *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, 1996.

[4] D. Etherington. *Reasoning with Incomplete Information: Investigations of Non-Monotonic Reasoning*. PhD thesis, Department of Computer Science, University of British Columbia, Vancouver, BC, 1986. Revised Version appeared as: Research Notes in AI, Pitman.

[5] D. Etherington. Formalizing nonmonotonic reasoning. *Artificial Intelligence*, 31:41–85, 1987.

[6] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, June 1992.

[7] U. Junker and K. Konolige. Computing the extensions of autoepistemic and default logic with a TMS. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 1990.

[8] W. Marek and M. Truszczyński. *Nonmonotonic logic: context-dependent reasoning*. Artifical Intelligence. Springer Verlag, 1993.

[9] I. Niemelä. Towards efficient default reasoning. In C. Mellish, editor, *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 312–318. Morgan Kaufmann Publishers, 1995.

[10] C. Papadimitriou and M. Sideri. Default theories that always have extensions. *Artificial Intelligence*, 69:347–357, 1994.

[11] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[12] T. Schaub. A new methodology for query-answering in default logics via structure-oriented theorem proving. *Journal of Automated Reasoning*, 15(1):95–165, 1995.

[13] G. Schwarz and M. Truszczyński. Subnormal modal logics for knowledge representation. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 438–443. Morgan Kaufmann Publishers, 1993.

[14] C. Schwind and V. Risch. Tableau-based characterization and theorem proving for default logic. *Journal of Automated Reasoning*, 13:223–242, 1994.

[12] Note that general default theories are reducible to *weak* semi-normal theories only [8], to which [4, 10] are inapplicable.