**Proceedings Paper:**
Beyersdorff, O, Meier, A, Thomas, M et al. (1 more author) (2009) The Complexity of Reasoning for Fragments of Default Logic. In: Theory and Applications of Satisfiability Testing 2009. 12th International Conference, SAT 2009, 30 June - 03 July 2009, Swansea. Lecture Notes in Computer Science, 5584 (Lectur). Springer Verlag , 51 - 64 . ISBN 978-3-642-02776-5

https://doi.org/10.1007/978-3-642-02777-2_7

# THE COMPLEXITY OF REASONING FOR FRAGMENTS OF DEFAULT LOGIC

OLAF BEYERSDORFF AND ARNE MEIER AND MICHAEL THOMAS AND HERIBERT VOLLMER

Institut für Theoretische Informatik, Gottfried Wilhelm Leibniz Universität
Appelstr. 4, 30167 Hannover, Germany
*E-mail address*: {beyersdorff,meier,thomas,vollmer}@thi.uni-hannover.de

Abstract. Default logic was introduced by Reiter in 1980. In 1992, Gottlob classified the complexity of the extension existence problem for propositional default logic as $\Sigma_2^p$-complete, and the complexity of the credulous and skeptical reasoning problem as $\Sigma_2^p$-complete, resp. $\Pi_2^p$-complete. Additionally, he investigated restrictions on the default rules, i.e., semi-normal default rules. Selman made 1992 a similar approach with disjunction-free and unary default rules. In this paper we systematically restrict the set of allowed propositional connectives. We give a complete complexity classification for all sets of Boolean functions in the meaning of Post's lattice for all three common decision problems for propositional default logic. We show that the complexity is a trichotomy ($\Sigma_2^p$-, NP-complete, trivial) for the extension existence problem, whereas for the credulous and sceptical reasoning problem we get a finer classification down to NL-complete cases.

## 1. Introduction

Reiter's default logic is one of the best known and most successful formalisms to model common-sense reasoning. Default logic extends the usual logical (first-order or propositional) derivations by patterns for default assumptions. These are of the form "in the absence of contrary information, assume ...". Reiter argued that his logic is an adequate formalization of the human reasoning under the *closed world assumption*. In fact, today default logic is widely used in artificial intelligence and computational logic.

What makes default logic computationally presumably harder than propositional or first-order logic is the fact that the semantics (i. e., the set of consequences) of a given set of premises is defined in terms of a fixed-point equation. The different fixed points (known as *extensions* or *expansions*) correspond to different possible sets of knowledge of an agent, based on the given premises.

In a seminal paper from 1992, Georg Gottlob formally defined three important decision problems for default logic:

(1) Given a set of premises, decide whether it has an extension at all.
(2) Given a set of premises and a formula, decide whether the formula occurs in at least one extension (so called *brave* or *credulous reasoning*).

*Key words and phrases:* Computational complexity, default logic, nonmonotonic reasoning, Post's lattice.
Supported in part by DFG grant VO 630/6-1.

(3) Given a set of premises and a formula, decide whether the formula occurs in all extensions (*cautious* or *sceptical reasoning*).

While in the case of first-order default logic, all these computational tasks are undecidable, Gottlob proved that for *propositional default logic*, the first and second are complete for the class $\Sigma_2^{\mathrm{p}}$, the second level of the polynomial hierarchy (Meyer-Stockmeyer hierarchy), while the third is complete for the class $\Pi_2^{\mathrm{p}}$ (the class of complements of $\Sigma_2^{\mathrm{p}}$ sets).

In the past, various semantical and syntactical restrictions have been proposed in order to identify computationally easier or even tractable fragments (see, e. g., [Sti90, Sti92, KS91]). This is the starting point of the present paper. We propose a systematic study of fragments of default logic defined by restricting the set of allowed propositional connectives. For instance, if we look at the fragment where we forbid negation and allow only conjunction and disjunction, the *monotone fragment* of default logic, we show that while the first problem is trivial (there always is an extension, in fact a unique one), the second and third problem become coNP-complete. In this paper we look at all possible sets $B$ of propositional connectives and study the three decision problems defined by Gottlob when all involved formulas contain only connectives from $B$. The computational complexity of the problems then, of course, becomes a function of $B$. We will see that *Post's lattice* of all closed classes of Boolean functions is the right way to study all such sets $B$. Depending on the location of $B$ in this lattice, we completely classify the complexity of all three reasoning tasks, see Figs. 1 and 2. We will show that, depending on the set $B$ of occurring connectives, the problem to determine the existence of an extension is either $\Sigma_2^{\mathrm{p}}$-complete, NP-complete, or trivial, while the other two problems are complete in one of the classes $\Sigma_2^{\mathrm{p}}$ (or $\Pi_2^{\mathrm{p}}$), NP, coNP, P or NL (under first-order reductions).

The motivation behind our approach lies in the hope that identifying fragments of default logic with simpler reasoning procedures may help us to understand the sources of hardness for the full problem and to locate the boundary between hard and easy fragments. Our improved algorithms for easier fragments could lead to better tools than we have today.

This paper is organized as follows. After some preliminary remarks in Sect. 2, we introduce Boolean clones in Sect. 3. At this place we also provide a full classification of the complexity of logical implications for fragments of propositional logic, as this classification will serve as a central tool for subsequent sections. In Sect. 4, we start to investigate propositional default logic. Section 5 then presents our main results on the complexity of the decision problems for default logic. Finally, in Sect. 6 we conclude with a summary and a discussion. Due to space restrictions, some proofs are deferred to the appendix.

## 2. Preliminaries

In this paper we make use of standard notions of complexity theory. The arising complexity degrees compass the classes NL, P, NP, coNP, $\Sigma_2^{\mathrm{p}}$ and $\Pi_2^{\mathrm{p}}$ (cf. [Pap94] for background information). For the hardness results we use *constant-depth* reductions, defined as follows: A language $A$ is *constant-depth reducible* to a language $B$ ($A \leq_{\mathrm{cd}} B$) if there exists a logtime-uniform $\mathrm{AC}^0$-circuit family $\{C_n\}_{n\geq 0}$ with unbounded fan-in $\{\wedge, \vee, \neg\}$-gates and oracle gates for $B$ such that for all $x$, $C_{|x|}(x) = 1$ iff $x \in A$ (cf. [Vol99]).

We also assume familiarity with propositional logic. The set of all propositional formulae is denoted by $\mathcal{L}$. For $A \subseteq \mathcal{L}$ and $\varphi \in \mathcal{L}$, we write $A \models \varphi$ iff all assignments satisfying all formulas in $A$ also satisfy $\varphi$. By $\mathrm{Th}(A)$ we denote the set of all consequences of $A$, i. e. $\mathrm{Th}(A) = \{\varphi \mid A \models \varphi\}$. For a literal $l$ and a variable $x$, we define the meta-language

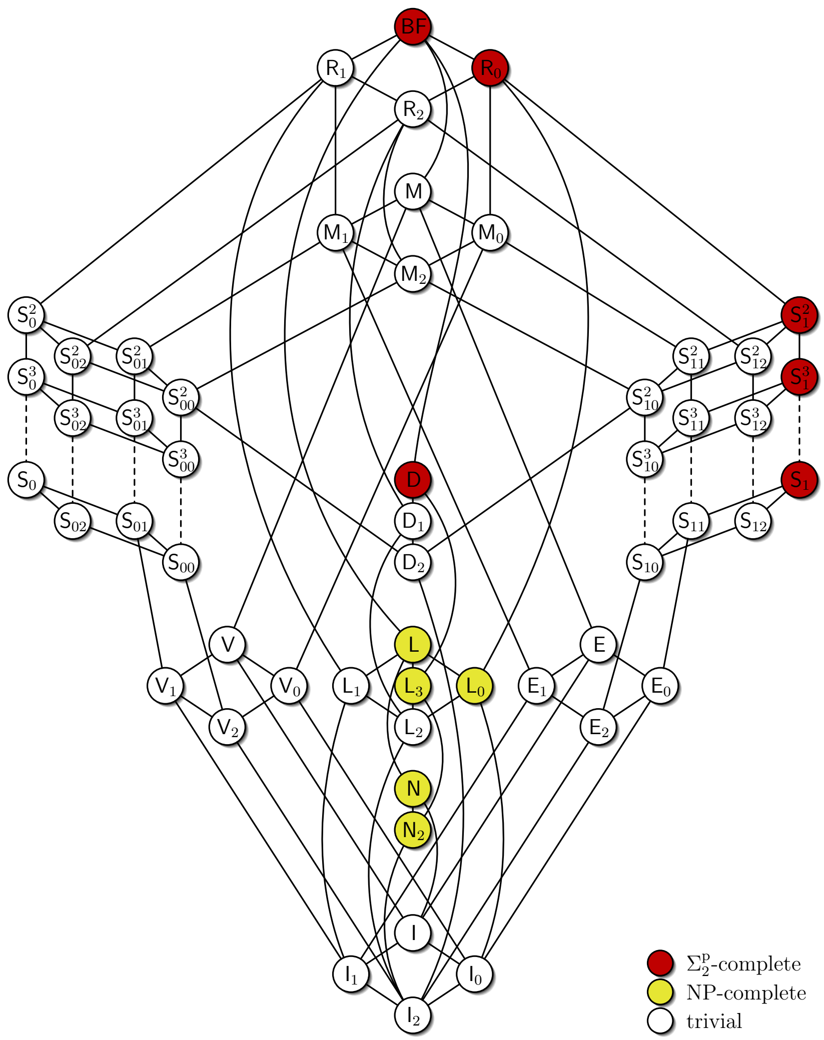


Figure 1: Post's lattice. Colors indicate the complexity of $\mathrm{EXT}(B)$, the Extension Existence Problem for $B$-formulae.

expression $\sim l$ as $\sim l := x$ if $l = \neg x$ and $\sim l := \neg x$ if $l = x$. For a formula $\varphi$, let $\varphi_{[\alpha/\beta]}$ denote $\varphi$ with all occurrences of $\alpha$ replaced by $\beta$, and let $A_{[\alpha/\beta]} := \{\varphi_{[\alpha/\beta]} \mid \varphi \in A\}$ for $A \subseteq \mathcal{L}$.

## 3. Boolean Clones and the Complexity of the Implication Problem

A propositional formula using only connectives from a finite set $B$ of Boolean functions is called a $B$-formula. The set of all $B$-formulae is denoted by $\mathcal{L}(B)$. In order to cope with the infinitely many finite sets $B$ of Boolean functions, we require some algebraic tools to classify the complexity of the infinitely many arising reasoning problems. A *clone* is a set $B$ of Boolean functions that is closed under superposition, i. e., $B$ contains all projections and is closed under arbitrary composition. For an arbitrary set $B$ of Boolean functions, we denote by $[B]$ the smallest clone containing $B$ and call $B$ a *base* for $[B]$. In [Pos41] Post classified the lattice of all clones and found a finite base for each clone, see Fig. 1.

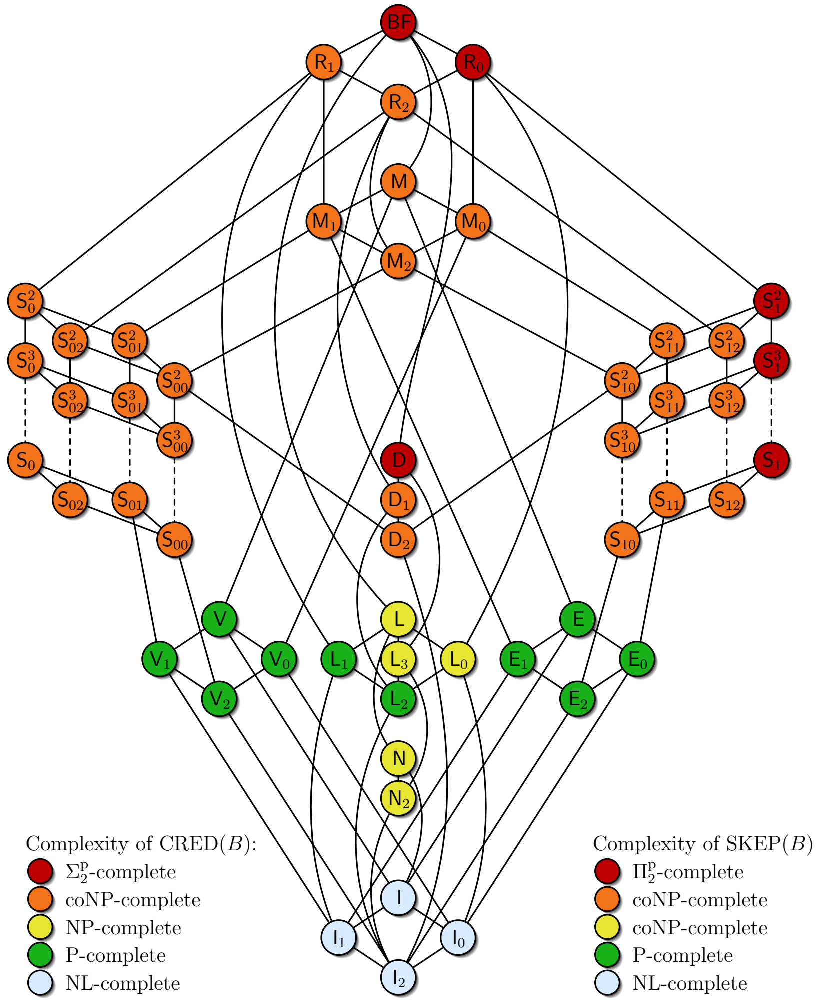


Figure 2: Post's lattice. Colors indicate the complexity of CRED($B$) and SKEP($B$), the Credulous and Skeptical Reasoning Problem for $B$-formulae.

In order to introduce the clones relevant to this paper, we define the following notions for $n$-ary Boolean functions $f$:

- $f$ is *c-reproducing* if $f(c, \ldots, c) = c$, $c \in \{0, 1\}$.
- $f$ is *monotone* if $a_1 \leq b_1, a_2 \leq b_2, \ldots, a_n \leq b_n$ implies $f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n)$.
- $f$ is *c-separating* if there exists an $i \in \{1, \ldots, n\}$ such that $f(a_1, \ldots, a_n) = c$ implies $a_i = c$, $c \in \{0, 1\}$.
- $f$ is *self-dual* if $f \equiv \mathrm{dual}(f)$, where $\mathrm{dual}(f)(x_1, \ldots, x_n) = \neg f(\neg x_1, \ldots, \neg x_n)$.
- $f$ is *linear* if $f \equiv x_1 \oplus \cdots \oplus x_n \oplus c$ for a constant $c \in \{0, 1\}$ and variables $x_1, \ldots, x_n$.

The clones relevant to this paper are listed in Table 1. The definition of all Boolean clones can be found, e. g., in [BCRV03].

For a finite set $B$ of Boolean functions, we define the *Implication Problem* for $B$-formulae IMP($B$) as the following computational task: given a set $A$ of $B$-formulae and a $B$-formula $\varphi$, decide whether $A \models \varphi$ holds. The following theorem provides a classification of the complexity of the implication problem. The full proof is contained in the appendix.

| Name | Definition | Base |
|---|---|---|
| $\mathsf{BF}$ | All Boolean functions | $\{\wedge, \neg\}$ |
| $\mathsf{R}_0$ | $\{f : f$ is 0-reproducing$\}$ | $\{\wedge, \not\rightarrow\}$ |
| $\mathsf{R}_1$ | $\{f : f$ is 1-reproducing$\}$ | $\{\vee, \rightarrow\}$ |
| $\mathsf{M}$ | $\{f : f$ is monotone$\}$ | $\{\vee, \wedge, 0, 1\}$ |
| $\mathsf{S}_0$ | $\{f : f$ is 0-separating$\}$ | $\{\rightarrow\}$ |
| $\mathsf{S}_1$ | $\{f : f$ is 1-separating$\}$ | $\{\not\rightarrow\}$ |
| $\mathsf{S}_{00}$ | $\mathsf{S}_0 \cap \mathsf{R}_0 \cap \mathsf{R}_1 \cap \mathsf{M}$ | $\{x \vee (y \wedge z)\}$ |
| $\mathsf{S}_{10}$ | $\mathsf{S}_1 \cap \mathsf{R}_0 \cap \mathsf{R}_1 \cap \mathsf{M}$ | $\{x \wedge (y \vee z)\}$ |
| $\mathsf{D}$ | $\{f : f$ is self-dual$\}$ | $\{(x \wedge \overline{y}) \vee (x \wedge \overline{z}) \vee (\overline{y} \wedge \overline{z})\}$ |
| $\mathsf{D}_2$ | $\mathsf{D} \cap \mathsf{M}$ | $\{(x \wedge y) \vee (y \wedge z) \vee (x \wedge z)\}$ |
| $\mathsf{L}$ | $\{f : f$ is linear$\}$ | $\{\oplus, 1\}$ |
| $\mathsf{L}_0$ | $\mathsf{L} \cap \mathsf{R}_0$ | $\{\oplus\}$ |
| $\mathsf{L}_1$ | $\mathsf{L} \cap \mathsf{R}_1$ | $\{\equiv\}$ |
| $\mathsf{L}_2$ | $\mathsf{L} \cap \mathsf{R}_0 \cap \mathsf{R}_1$ | $\{x \oplus y \oplus z\}$ |
| $\mathsf{L}_3$ | $\mathsf{L} \cap \mathsf{D}$ | $\{x \oplus y \oplus z, \neg\}$ |
| $\mathsf{V}$ | $\{f : f \equiv c_0 \vee \bigvee_{i=1}^{n} c_i x_i$ where the $c_i$s are constant$\}$ | $\{\vee, 0, 1\}$ |
| $\mathsf{V}_2$ | $[\{\vee\}]$ | $\{\vee\}$ |
| $\mathsf{E}$ | $\{f : f \equiv c_0 \wedge \bigwedge_{i=1}^{n} c_i x_i$ where the $c_i$s are constant$\}$ | $\{\wedge, 0, 1\}$ |
| $\mathsf{E}_2$ | $[\{\wedge\}]$ | $\{\wedge\}$ |
| $\mathsf{N}$ | $\{f : f$ depends on at most one variable$\}$ | $\{\neg, 0, 1\}$ |
| $\mathsf{N}_2$ | $[\{\neg\}]$ | $\{\neg\}$ |
| $\mathsf{I}$ | $\{f : f$ is a projection or a constant$\}$ | $\{\mathrm{id}, 0, 1\}$ |
| $\mathsf{I}_2$ | $[\{id\}]$ | $\{\mathrm{id}\}$ |

Table 1: A list of Boolean clones with definitions and bases.

**Theorem 3.1.** *Let $B$ be a finite set of Boolean functions. Then* $\mathrm{IMP}(B)$ *is* coNP*-complete if* $\mathsf{S}_{00} \subseteq [B]$*,* $\mathsf{S}_{10} \subseteq [B]$ *or* $\mathsf{D}_2 \subseteq [B]$*, and in* P *for all other cases.*

In the above theorem, we can further classify the tractable part into $\oplus$L-complete and $\mathrm{AC}^0$-complete cases, but this refined analysis is not needed for the results of this paper.

## 4. Default Logic

Fix some finite set $B$ of Boolean functions and let $\alpha, \beta, \gamma$ be propositional $B$-formulae. A *$B$-default (rule)* is an expression $d = \frac{\alpha:\beta}{\gamma}$; $\alpha$ is called *prerequisite*, $\beta$ is called *justification* and $\gamma$ is called *consequent* of $d$. A *$B$-default theory* is a pair $\langle W, D\rangle$, where $W$ is a set of propositional $B$-formulae and $D$ is a set of $B$-default rules. Henceforth will we will omit the prefix "$B$-" if $B = \mathsf{BF}$ or the meaning is clear from the context.

For a given default theory $\langle W, D\rangle$ and a set of formulae $E$, let $\Gamma(E)$ be the smallest set of formulae such that

(1) $W \subseteq \Gamma(E)$,
(2) $\Gamma(E)$ is closed under deduction, i.e. $\Gamma(E) = \mathrm{Th}(\Gamma(E))$, and
(3) for all defaults $\frac{\alpha:\beta}{\gamma} \in D$ with $\alpha \in \Gamma(E)$ and $\neg\beta \notin E$, it holds that $\gamma \in \Gamma(E)$.

A *(stable) extension* of $\langle W, D\rangle$ is a fixpoint of $\Gamma$, i.e. a set $E$ such that $E = \Gamma(E)$.

The following theorem by Reiter provides an alternative characterization of extensions:

**Theorem 4.1** ([Rei80])**.** *Let $\langle W, D\rangle$ be a default theory and $E$ be a set of formulae.*

(1) *Let* $E_0 = W$ *and* $E_{i+1} = \mathrm{Th}(E_i) \cup \left\{\gamma \;\middle|\; \frac{\alpha:\beta}{\gamma} \in D, \alpha \in E_i \text{ and } \neg\beta \notin E\right\}$*. Then* $E$ *is a stable extension of* $\langle W, D\rangle$ *iff* $E = \bigcup_{i\in\mathbb{N}} E_i$*.*

(2) *Let* $G = \left\{\frac{\alpha:\beta}{\gamma} \in D \;\middle|\; \alpha \in E \text{ and } \neg\beta \notin E\right\}$*. If* $E$ *is a stable extension of* $\langle W, D\rangle$*, then*

$$E = \mathrm{Th}(W \cup \{\gamma \mid \tfrac{\alpha:\beta}{\gamma} \in G\}).$$

*In this case,* $G$ *is also called the set of* generating defaults *for* $E$*.*

Note that stable extensions need not be consistent. However, the following proposition shows that this only occurs if the set $W$ is inconsistent already.

**Proposition 4.2.** *Let* $\langle W, D\rangle$ *be a default theory. Then* $\mathcal{L}$ *is a stable extension of* $\langle W, D\rangle$ *iff* $W$ *is inconsistent.*

*Proof.* Let $\langle W, D\rangle$ be a default theory. We compute the set $\Gamma(\mathcal{L})$ from the above definition of $\Gamma$. The third condition does not apply, because for $E = \mathcal{L}$, $\neg\beta \notin E$ is never fulfilled. Therefore, $\Gamma(\mathcal{L}) = \mathrm{Th}(W)$ by conditions 1 and 2. Now, $\mathcal{L}$ is a fixpoint of $\Gamma$ iff $\mathrm{Th}(W) = \mathcal{L}$, i.e., if $W$ is inconsistent. ∎

As a consequence we obtain:

**Corollary 4.3.** *Let* $\langle W, D\rangle$ *be a default theory.*

- *If* $W$ *is consistent, then every stable extension of* $\langle W, D\rangle$ *is consistent.*
- *If* $W$ *is inconsistent, then* $\langle W, D\rangle$ *has a stable extension.*

The main reasoning tasks in nonmonotonic logics give rise to the following three decision problems:

(1) the *Extension Existence Problem* $\mathrm{EXT}(B)$
   *Instance:* a $B$-default theory $\langle W, D\rangle$
   *Question:* Has $\langle W, D\rangle$ a stable extension?

(2) the *Credulous Reasoning Problem* $\mathrm{CRED}(B)$
   *Instance:* a $B$-formula $\varphi$ and a $B$-default theory $\langle W, D\rangle$
   *Question:* Is there a stable extension of $\langle W, D\rangle$ that includes $\varphi$?

(3) the *Skeptical Reasoning Problem* $\mathrm{SKEP}(B)$
   *Instance:* a $B$-formula $\varphi$ and a $B$-default theory $\langle W, D\rangle$
   *Question:* Does every stable extension of $\langle W, D\rangle$ include $\varphi$?

The next theorem follows from [Got92] and states the complexity of the above decision problems for the general case $[B] = \mathsf{BF}$.

**Theorem 4.4.** *Let* $B$ *be a finite set of Boolean functions such that* $[B] = \mathsf{BF}$*. Then* $\mathrm{EXT}(B)$ *and* $\mathrm{CRED}(B)$ *are* $\Sigma_2^{\mathrm{p}}$*-complete, whereas* $\mathrm{SKEP}(B)$ *is* $\Pi_2^{\mathrm{p}}$*-complete.*

*Proof.* The upper bounds given in [Got92] do not depend on the Boolean connectives allowed and thus hold for any finite set $B$ of Boolean functions. For $\Sigma_2^{\mathrm{p}}$- and $\Pi_2^{\mathrm{p}}$-hardness, it suffices to note that if $[B] = \mathsf{BF}$, then there exist $B$-formulae $f(x, y)$, $g(x, y)$ and $h(x)$ such that $f(x, y) \equiv x \wedge y$, $g(x, y) \equiv x \vee y$, $h(x) \equiv \neg x$ and both $x$ and $y$ occur at most once in $f$, $g$ and $h$ [Lew79]. Hence, the hardness results proved by Gottlob [Got92] generalize to arbitrary bases $B$ with $[B] = \mathsf{BF}$. ∎

## 5. The Complexity of Default Reasoning

In this section we will classify the complexity of the three problems $\mathrm{EXT}(B)$, $\mathrm{CRED}(B)$, and $\mathrm{SKEP}(B)$ for all choices of Boolean connectives $B$. We start with some preparations which will substantially reduce the number of cases we have to consider.

**Lemma 5.1.** *Let* P *be any of the problems* EXT, CRED, *or* SKEP. *Then for each finite set $B$ of Boolean functions,* $\mathrm{P}(B) \equiv_{\mathrm{cd}} \mathrm{P}(B \cup \{1\})$.

*Proof.* The reductions $\mathrm{P}(B) \leq_{\mathrm{cd}} \mathrm{P}(B \cup \{1\})$ are obvious. For the converse reductions, we will essentially substitute the constant 1 by a new variable $t$ that is forced to be true (this trick goes already back to Lewis [Lew79]). For EXT, the reduction is given by $\langle W, D\rangle \mapsto \langle W', D'\rangle$, where $W' = W_{[1/t]} \cup \{t\}$, $D' = D_{[1/t]}$, and $t$ is a new variable not occurring in $\langle W, D\rangle$. If $\langle W', D'\rangle$ possesses a stable extension $E'$, then $t \in E'$. Hence, $E'_{[t/1]}$ is a stable extension of $\langle W, D\rangle$. On the other hand, if $E$ is a stable extension of $\langle W, D\rangle$, then $\mathrm{Th}(E_{[1/t]} \cup \{t\}) = E_{[1/t]}$ is a stable extension of $\langle W', D'\rangle$. Therefore, each extension $E$ of $\langle W, D\rangle$ corresponds to the extension $E_{[1/t]}$ of $\langle W', D'\rangle$, and vice versa.

For the problems CRED and SKEP, it suffices to note that the above reduction $\langle W, D\rangle \mapsto \langle W', D'\rangle$ has the additional property that for each formula $\varphi$ and each extension $E$ of $\langle W, D\rangle$, $\varphi \in E$ iff $\varphi_{[1/t]} \in E_{[1/t]}$. ∎

The next lemma shows that, quite often, $B$-default theories have unique extensions.

**Lemma 5.2.** *Let $B$ be a finite set of Boolean functions such that $[B] \subseteq \mathsf{R}_1$ or $[B] \subseteq \mathsf{M}$. Let $\langle W, D\rangle$ be a $B$-default theory with finite $D$. Then $\langle W, D\rangle$ has a unique stable extension.*

*Proof.* For $[B] \subseteq \mathsf{R}_1$, every premise, justification and consequent is 1-reproducing. As all consequences of 1-reproducing functions are again 1-reproducing and the negation of a 1-reproducing function is not 1-reproducing, the justifications in $D$ become irrelevant. Hence the characterization of stable extensions from the first item in Theorem 4.1 simplifies to the following iterative construction: $E_0 = W$ and $E_{i+1} = \mathrm{Th}(E_i) \cup \left\{\gamma \;\middle|\; \frac{\alpha:\beta}{\gamma} \in D, \alpha \in E_i\right\}$. As $D$ is finite, this construction terminates after finitely many steps, i.e., $E_k = E_{k+1}$ for some $k \geq 0$. Then $E = \bigcup_{i \leq k} E_i$ is the unique stable extension of $\langle W, D\rangle$.

For $[B] \subseteq \mathsf{M}$, every formula is either 1-reproducing or equivalent to 0. As rules with justification equivalent to 0 are never applicable, each $B$-default theory $\langle W, D\rangle$ with finite $D$ has a unique stable extension by the same argument as above. ∎

As an immediate corollary, the credulous and the sceptical reasoning problem are equivalent for the above choices of the underlying connectives.

**Corollary 5.3.** *Let $B$ be a finite set of Boolean functions such that $[B] \subseteq \mathsf{R}_1$ or $[B] \subseteq \mathsf{M}$. Then* $\mathrm{CRED}(B) \equiv_{\mathrm{cd}} \mathrm{SKEP}(B)$.

### 5.1. The Extension Existence Problem

Now we are ready to classify the complexity of EXT. The next theorem shows that this is a trichotomy: the $\Sigma_2^{\mathrm{p}}$-completeness of the general case [Got92] is inherited by all clones above $\mathsf{S}_1$ and $\mathsf{D}$, for a number of clones the complexity of EXT reduces to NP-completeness, and, due to Lemma 5.2, for the majority of cases the problem becomes trivial.

**Theorem 5.4.** *Let $B$ be a finite set of Boolean functions. Then* $\mathrm{EXT}(B)$ *is*

(1) $\Sigma_2^{\mathrm{p}}$-*complete if* $\mathsf{S}_1 \subseteq [B] \subseteq \mathsf{BF}$ *or* $\mathsf{D} \subseteq [B] \subseteq \mathsf{BF}$,
(2) NP-*complete if* $[B] \in \{\mathsf{N}, \mathsf{N}_2, \mathsf{L}, \mathsf{L}_0, \mathsf{L}_3\}$, *and*
(3) *trivial in all other cases (i. e., if* $[B] \subseteq \mathsf{R}_1$ *or* $[B] \subseteq \mathsf{M}$*).*

*Proof.* For $\mathsf{S}_1 \subseteq [B] \subseteq \mathsf{BF}$ or $[B] = \mathsf{D}$, observe that in both cases $\mathsf{BF} = [B \cup \{1\}]$. Claim 1 then follows from Theorem 4.4 and Lemma 5.1.

For the second claim, it suffices to prove membership in NP for $\mathrm{EXT}(B)$ for every finite $B \subseteq \mathsf{L}$ and NP-hardness for $\mathrm{EXT}(B)$ for every finite $B$ with $\mathsf{N} \subseteq [B]$. The remaining cases $[B] \in \{\mathsf{N}_2, \mathsf{L}_0, \mathsf{L}_3\}$ all follow from Lemma 5.1, because $[\mathsf{N}_2 \cup \{1\}] = \mathsf{N}$, $[\mathsf{L}_0 \cup \{1\}] = \mathsf{L}$, and $[\mathsf{L}_3 \cup \{1\}] = \mathsf{L}$.

We start by showing $\mathrm{EXT}(\mathsf{L}) \in \mathrm{NP}$. Given a default theory $\langle W, D\rangle$, we first guess a set $G \subseteq D$ which will serve as the set of generating defaults for a stable extension. Let $G' = W \cup \{\gamma \mid \frac{\alpha:\beta}{\gamma} \in G\}$. We use Theorem 4.1 to verify whether $\mathrm{Th}(G')$ is indeed a stable extension of $\langle W, D\rangle$. For this we inductively compute generators $G_i$ for the sets $E_i$ from Theorem 4.1, until eventually $E_i = E_{i+1}$ (note, that because $D$ is finite, this always occurs). We start by setting $G_0 = W$. Given $G_i$, we check for each rule $\frac{\alpha:\beta}{\gamma} \in D$, whether $G_i \models \alpha$ and $G' \not\models \neg\beta$ (as all formulas belong to $\mathcal{L}(B)$, this is possible by Theorem 3.1). If so, then $\gamma$ is put into $G_{i+1}$. If this process terminates, i.e., if $G_i = G_{i+1}$, then we check whether $G' = G_i$. By Theorem 4.1, this test is positive iff $G$ generates a stable extension of $\langle W, D\rangle$.

To show NP-hardness of $\mathrm{EXT}(B)$ for $\mathsf{N} \subseteq [B]$, we will $\leq_{\mathrm{cd}}$-reduce 3SAT to $\mathrm{EXT}(B)$. Let $\varphi = \bigwedge_{i=1}^{n}(l_{i1} \vee l_{i2} \vee l_{i3})$ and $l_{ij}$ be literals over propositions $\{x_1, \ldots, x_m\}$ for $1 \leq i \leq n$, $1 \leq j \leq 3$. We transform $\varphi$ to the $B$-default theory $\langle W, D_\varphi\rangle$, where $W := \emptyset$ and

$$D_\varphi := \left\{ \frac{1 : x_i}{x_i} \;\middle|\; 1 \leq i \leq m \right\} \cup \left\{ \frac{1 : \neg x_i}{\neg x_i} \;\middle|\; 1 \leq i \leq m \right\} \cup \left\{ \frac{\sim l_{i\pi(1)} : \sim l_{i\pi(2)}}{l_{i\pi(3)}} \;\middle|\; 1 \leq i \leq n, \pi \text{ is a permutation of } \{1,2,3\} \right\} .$$

To prove the correctness of the reduction, first assume $\varphi$ to be satisfiable. For each satisfying assignment $\sigma : \{x_1, \ldots, x_m\} \to \{0, 1\}$ for $\varphi$, we claim that

$$E := \mathrm{Th}(\{x_i \mid \sigma(x_i) = 1\} \cup \{\neg x_i \mid \sigma(x_i) = 0\})$$

is a stable extension of $\langle W, D_\varphi\rangle$. We will verify this claim with the help of the first part of Theorem 4.1. Starting with $E_0 = \emptyset$, we already get $E_1 = E$ by the default rules $\frac{1:x_i}{x_i}$ and $\frac{1:\neg x_i}{\neg x_i}$ in $D_\varphi$. As $\sigma$ is a satisfying assignment for $\varphi$, each consequent of a default rule in $D_\varphi$ is already in $E$. Hence $E_2 = E_1$ and therefore $E = \bigcup_{i \in \mathbb{N}} E_i$ is a stable extension of $\langle W, D_\varphi\rangle$.

Conversely, assume that $E$ is a stable extension of $\langle W, D_\varphi\rangle$. Because of the default rules $\frac{1:x_i}{x_i}$ and $\frac{1:\neg x_i}{\neg x_i}$, we either get $x_i \in E$ or $\neg x_i \in E$ for all $i = 1, \ldots, m$. The rules of the type $\frac{\sim l_{i1}:\sim l_{i2}}{l_{i3}}$ ensure that $E$ contains at least one literal from each clause $l_{i1} \vee l_{i2} \vee l_{i3}$ in $\varphi$. As $E$ is deductively closed, $E$ contains $\varphi$. By Corollary 4.3, the extension $E$ is consistent, and therefore $\varphi$ is satisfiable.

Finally, the third item of the theorem directly follows from Lemma 5.2. ■

### 5.2. The Credulous and the Sceptical Reasoning Problem

Now we will analyse the credulous and the sceptical reasoning problem. For these problems, there are two sources for the complexity. On the one hand, we need to determine a candidate for a stable extension. On the other hand, we have to verify that this candidate is indeed a finite characterization of some stable extension — a task that requires to test for formula implication. Whence the $\Sigma_2^{\mathrm{p}}$-completeness of $\mathrm{CRED}(B)$ and the $\Pi_2^{\mathrm{p}}$-completeness of $\mathrm{SKEP}(B)$ if $[B] = \mathsf{BF}$. Depending on the Boolean connectives allowed, one or both tasks can be performed in polynomial time. We obtain coNP-completeness for clones that guarantee the existence of a stable extension but whose implication problem remains coNP-complete. Conversely, if the implication problem becomes easy, but determining an extension candidates is hard, then $\mathrm{CRED}(B)$ is NP-complete, while $\mathrm{SKEP}(B)$ has to test for all extensions and is coNP-complete. This is the case for the clones $[B] \in \{\mathsf{N}, \mathsf{N}_2, \mathsf{L}, \mathsf{L}_0, \mathsf{L}_3\}$. Finally, for clones $B$ that allow for solving both tasks in polynomial time, $\mathrm{CRED}(B)$ and $\mathrm{SKEP}(B)$ are in P. The complete classification of $\mathrm{CRED}(B)$ is given in the following theorem.

**Theorem 5.5.** *Let $B$ be a finite set of Boolean functions. Then* $\mathrm{CRED}(B)$ *is*

(1) $\Sigma_2^{\mathrm{p}}$-*complete if* $\mathsf{S}_1 \subseteq [B] \subseteq \mathsf{BF}$ *or* $\mathsf{D} \subseteq [B] \subseteq \mathsf{BF}$,
(2) coNP-*complete if* $X \subseteq [B] \subseteq Y$, *where* $X \in \{\mathsf{S}_{00}, \mathsf{S}_{10}, \mathsf{D}_2\}$ *and* $Y \in \{\mathsf{R}_1, \mathsf{M}\}$,
(3) NP-*complete if* $[B] \in \{\mathsf{N}, \mathsf{N}_2, \mathsf{L}, \mathsf{L}_0, \mathsf{L}_3\}$,
(4) P-*complete if* $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$, $\mathsf{E}_2 \subseteq [B] \subseteq \mathsf{E}$ *or* $[B] \in \{\mathsf{L}_1, \mathsf{L}_2\}$, *and*
(5) NL-*complete if* $\mathsf{I}_2 \subseteq [B] \subseteq \mathsf{I}$.

The proof of Theorem 5.5 follows from the upper and lower bounds given in the Propositions 5.6 and 5.7 below.

**Proposition 5.6.** *Let $B$ be a finite set of Boolean functions. Then* $\mathrm{CRED}(B)$ *is contained*

(1) *in* $\Sigma_2^{\mathrm{p}}$ *if* $\mathsf{S}_1 \subseteq [B] \subseteq \mathsf{BF}$ *or* $\mathsf{D} \subseteq [B] \subseteq \mathsf{BF}$,
(2) *in* coNP *if* $[B] \subseteq \mathsf{R}_1$ *or* $[B] \subseteq \mathsf{M}$,
(3) *in* NP *if* $[B] \subseteq \mathsf{L}$,
(4) *in* P *if* $[B] \subseteq \mathsf{V}$, $[B] \subseteq \mathsf{E}$ *or* $[B] \subseteq \mathsf{L}_1$, *and*
(5) *in* NL *if* $[B] \subseteq \mathsf{I}$.

*Proof.* Part 1 follows from Theorem 4.4 and Lemma 5.1.

For $[B] \subseteq \mathsf{R}_1$, let $\langle W, D\rangle$ be an $\mathsf{R}_1$-default theory and $\varphi \in \mathcal{L}(\mathsf{R}_1)$. As for every default rule $\frac{\alpha:\beta}{\gamma} \in D$ we can never derive $\neg\beta$ (as $\neg\beta$ is not 1-reproducing), the justifications $\beta$ are irrelevant for computing a stable extension. Thence, using the characterization in the first part of Theorem 4.1, we can iteratively compute the applicable defaults and eventually check whether $\varphi$ is implied by $W$ and those generating defaults. Algorithm 1 implements these steps on a deterministic Turing machine using a coNP-oracle to test for implication of $B$-formulae. Clearly, Algorithm 1 terminates after a polynomial number of steps. Moreover, Algorithm 1 is a monotone $\leq_{\mathrm{T}}^{\mathrm{p}}$-reduction from $\mathrm{CRED}(B)$ to $\mathrm{IMP}(B)$, in the sense that for any deterministic oracle Turing machine $M$ that executes Algorithm 1, $A \subseteq B$ implies $L(M, A) \subseteq L(M, B)$, where $L(M, X)$ is the language recognized by $M$ with oracle $X$. As coNP is closed under monotone $\leq_{\mathrm{T}}^{\mathrm{p}}$-reductions [Sel82], $\mathrm{CRED}(B) \in$ coNP.

**Algorithm 1** Determine existence of a stable extension of $\langle W, D\rangle$ containing $\varphi$.

**Require:** $\langle W, D\rangle$, $\varphi$
$G_{\mathrm{new}} \leftarrow W$
**repeat**
  $G_{\mathrm{old}} \leftarrow G_{\mathrm{new}}$
  **for all** $\frac{\alpha:\beta}{\gamma} \in D$ **do**
    **if** $G_{\mathrm{old}} \models \alpha$ **then**
      $G_{\mathrm{new}} \leftarrow G_{\mathrm{new}} \cup \{\gamma\}$
    **end if**
  **end for**
**until** $G_{\mathrm{new}} = G_{\mathrm{old}}$
**if** $G_{\mathrm{new}} \models \varphi$ **then**
  **return true**
**else**
  **return false**
**end if**

For $[B] \subseteq \mathsf{M}$, Algorithm 1 can be easily adopted, because we are restricted to 1-reproducing functions and the constant 0. Thus, before executing Algorithm 1, we just delete all rules $\frac{\alpha:\beta}{\gamma}$ with $\beta \equiv 0$ from $D$, as these rules are never applicable.

For $[B] \subseteq \mathsf{L}$, we proceed similarly as in the proof of item 2 in Theorem 5.4. First, we guess a set $G$ of generating defaults and subsequently verify that both $\mathrm{Th}(W \cup \{\gamma \mid \frac{\alpha:\beta}{\gamma} \in G\})$ is a stable extension and that $W \cup \{\gamma \mid \frac{\alpha:\beta}{\gamma} \in G\} \models \varphi$. Using Theorem 3.1, both conditions may be verified in polynomial time.

For $[B] \subseteq \mathsf{V}$, $[B] \subseteq \mathsf{E}$, and $[B] \subseteq \mathsf{L}_1$, we again use Algorithm 1. As for these types of $B$-formulae, we have an efficient test for implication (Theorem 3.1), $\mathrm{CRED}(B) \in \mathrm{P}$.

For $[B] \subseteq \mathsf{I}$, we show that $\mathrm{CRED}(B)$ is constant-depth reducible to the graph accessibility problem, GAP, a problem that is $\leq_{\mathrm{cd}}$-complete for NL. Let $\langle W, D\rangle$ be an $\mathsf{I}$-default theory with $D = \{\frac{\alpha_i:\beta_i}{\gamma_i} \mid 1 \leq i \leq k\}$ and let $\varphi$ be an $\mathsf{I}$-formula. We transform $(\langle W, D\rangle, \varphi)$ to the GAP-instance $(G, \bigwedge_{\psi \in W} \psi, \varphi)$, where $G = (V, E)$ is a directed graph with

$$V := \{\alpha_i \mid 1 \leq i \leq k\} \cup \{\gamma_i \mid 1 \leq i \leq k\} \cup \{\textstyle\bigwedge_{\psi \in W} \psi, \varphi\} \text{ and}$$
$$E := \{(\alpha_i, \gamma_i) \mid 1 \leq i \leq k\} \cup \{(u, v) \in V \mid u \models v\} \ .$$

Then $\varphi$ is included in the (unique) stable extension of $\langle W, D\rangle$ iff $G$ contains a path from $\bigwedge_{\psi \in W} \psi$ to $\varphi$. As implication testing for all $B \subseteq \mathsf{I}$ is possible in $\mathrm{AC}^0$, $\mathrm{CRED}(B) \leq_{\mathrm{cd}} \mathrm{GAP}$. ∎

We will now establish the lower bounds required to complete the proof of Theorem 5.5.

**Proposition 5.7.** *Let $B$ be a finite set of Boolean functions. Then* $\mathrm{CRED}(B)$ *is*

(1) $\Sigma_2^{\mathrm{p}}$*-hard if* $\mathsf{S}_1 \subseteq [B]$ *or* $\mathsf{D} \subseteq [B]$*,*
(2) coNP*-hard if* $\mathsf{S}_{00} \subseteq [B]$*,* $\mathsf{S}_{10} \subseteq [B]$ *or* $\mathsf{D}_2 \subseteq [B]$*,*
(3) NP*-hard if* $\mathsf{N}_2 \subseteq [B]$ *or* $\mathsf{L}_0 \subseteq [B]$*,*
(4) P*-hard if* $\mathsf{V}_2 \subseteq [B]$*,* $\mathsf{E}_2 \subseteq [B]$ *or* $\mathsf{L}_2 \subseteq [B]$*, and*
(5) NL*-hard for all other clones.*

*Proof.* Part 1 follows from Theorem 4.4 and Lemma 5.1.

For $\mathsf{S}_{00} \subseteq [B]$, $\mathsf{S}_{10} \subseteq [B]$, and $\mathsf{D}_2 \subseteq [B]$, coNP-hardness is established by a $\leq_{\mathrm{cd}}$-reduction from $\mathrm{IMP}(B)$. Let $A \subseteq \mathcal{L}(B)$ and $\varphi \in \mathcal{L}(B)$. Then the default theory $\langle A, \emptyset\rangle$ has the unique stable extension $\mathrm{Th}(A)$, and hence $A \models \varphi$ iff $(\langle A, \emptyset\rangle, \varphi) \in \mathrm{CRED}(B)$. Therefore, $\mathrm{IMP}(B) \leq_{\mathrm{cd}} \mathrm{CRED}(B)$, and the claim follows with Theorem 3.1.

For the third item, it suffices to prove NP-hardness for $\mathsf{N}_2 \subseteq [B]$. For $\mathsf{L}_0 \subseteq [B]$, the claim then follows by Lemma 5.1. For $\mathsf{N}_2 \subseteq [B]$, we obtain NP-hardness of $\mathrm{CRED}(B)$ by adjusting the reduction given in the proof of item 2 of Theorem 5.4. Consider the mapping $\varphi \mapsto (\langle\{\psi\}, D_\varphi\rangle, \psi)$, where $D_\varphi$ is the set of default rules constructed from $\varphi$ in Theorem 5.4, and $\psi$ is a satisfiable $B$-formula such that $\varphi$ and $\psi$ do not use common variables. By Theorem 5.4, $\varphi \in \mathrm{3SAT}$ iff $\langle\{\psi\}, D_\varphi\rangle$ has a stable extension. As any extension of $\langle\{\psi\}, D_\varphi\rangle$ contains $\psi$, we obtain $\mathrm{3SAT} \leq_{\mathrm{cd}} \mathrm{CRED}(B)$ via the above reduction.

To prove P-hardness for $\mathsf{E}_2 \subseteq [B]$, $\mathsf{V}_2 \subseteq [B]$, and $[B] \in \{\mathsf{L}_1, \mathsf{L}_2\}$, we provide a reduction from the accessibility problem for directed hypergraphs, HGAP. HGAP is P-complete under $\leq_{\mathrm{cd}}$-reductions [SI90]. In directed hypergraphs $H = (V, E)$, hyperedges $e \in E$ consist of a set of source nodes $\mathrm{src}(e) \subseteq V$ and a destination $\mathrm{dest}(e) \in V$. Instances of HGAP contain a directed hypergraph $H = (V, E)$, a set $S \subseteq V$ of source nodes, and a target node $t \in V$.

We transform such an instance $(H, S, t)$ to the CRED($\{\wedge\}$)-instance $(\langle W, D\rangle, \varphi)$, where

$$W := \{p_s \mid s \in S\}, \qquad D := \left\{ \frac{\bigwedge_{v\in\mathrm{src}(e)} p_v \ : \ \bigwedge_{v\in\mathrm{src}(e)} p_v}{p_{\mathrm{dest}(e)}} \;\middle|\; e \in E \right\}, \qquad \varphi := p_t$$

with pairwise distinct propositions $p_v$ for $v \in V$. For $\mathsf{V}_2 \subseteq [B]$, we set

$$W := \{\bigvee_{s\notin S} p_s\}, \qquad D := \left\{ \frac{\bigvee_{v\in V\setminus\mathrm{src}(e)} p_v \ : \ \bigvee_{v\in V\setminus\mathrm{src}(e)} p_v}{\bigvee_{v\in V\setminus(\mathrm{src}(e)\cup\{\mathrm{dest}(e)\})} p_v} \;\middle|\; e \in E \right\}, \qquad \varphi := \bigvee_{v\in V\setminus\{t\}} p_v.$$

For $[B] \in \{\mathsf{L}_1, \mathsf{L}_2\}$, we again modify the above reduction and map $(H, S, t)$ to the CRED($B$)-instance

$$W := \{p_s \mid s \in S\}, \qquad D := \left\{ \frac{\equiv_{v\in\mathrm{src}(e)} p_v \ : \ \equiv_{v\in\mathrm{src}(e)} p_v}{p_{\mathrm{dest}(e)}} \;\middle|\; e \in E \right\}, \qquad \varphi := p_t.$$

The correctness of these reductions is easily verified.

Finally, it remains to show NL-hardness for $\mathsf{I}_2 \subseteq [B]$. We give a $\leq_{\mathrm{cd}}$-reduction from GAP to CRED($\{\mathrm{id}\}$). For a directed graph $G = (V, E)$ and two nodes $s, t \in V$, we transform the GAP-instance $(G, s, t)$ to the CRED($\mathsf{I}_2$)-instance

$$W := \{p_s\}, \qquad D := \left\{ \frac{p_u : p_u}{p_v} \;\middle|\; (u, v) \in V \right\}, \qquad \varphi := p_t.$$

Clearly, $(G, s, t) \in$ GAP iff $\varphi$ is contained in all stable extensions of $\langle W, D\rangle$. ∎

Finally, we will classify the complexity of the sceptical reasoning problem. The analysis is similar to the classification of the credulous reasoning problem (Theorem 5.5).

**Theorem 5.8.** *Let $B$ be a finite set of Boolean functions. Then* SKEP($B$) *is*

(1) $\Pi_2^{\mathrm{p}}$*-complete if* $\mathsf{S}_1 \subseteq [B] \subseteq \mathsf{BF}$ *or* $\mathsf{D} \subseteq [B] \subseteq \mathsf{BF}$*,*
(2) coNP*-complete if* $X \subseteq [B] \subseteq Y$*, where* $X \in \{\mathsf{S}_{00}, \mathsf{S}_{10}, \mathsf{N}_2, \mathsf{L}_0\}$ *and* $Y \in \{\mathsf{R}_1, \mathsf{M}, \mathsf{L}\}$*,*
(3) P*-complete if* $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$*,* $\mathsf{E}_2 \subseteq [B] \subseteq \mathsf{E}$ *or* $[B] \in \{\mathsf{L}_1, \mathsf{L}_2\}$*, and*
(4) NL*-complete if* $\mathsf{I}_2 \subseteq [B] \subseteq \mathsf{I}$*.*

*Proof.* The first part again follows from Theorem 4.4 and Lemma 5.1.

For $[B] \in \{\mathsf{N}, \mathsf{N}_2, \mathsf{L}, \mathsf{L}_0, \mathsf{L}_3\}$, we guess similarly as in Theorem 5.4 a set $G$ of defaults and then verify in the same way whether $W$ and $G$ generate a stable extension $E$. If not, then we accept. Otherwise, we check if $E \models \varphi$ and answer according to this test. This yields a coNP-algorithm for SKEP($B$). Hardness for coNP is achieved by modifying the reduction from Theorem 5.4 (cf. also the proof of Proposition 5.7): map $\varphi$ to $(\langle \emptyset, D_\varphi\rangle, \psi)$, where $D_\varphi$ is defined as in the proof of Theorem 5.4, and $\psi$ is a $B$-formula such that $\varphi$ and $\psi$ do not share variables. Then $\varphi \notin$ 3SAT iff $\langle \emptyset, D_\varphi\rangle$ does not have a stable extension. The latter is true iff $\psi$ is in all extensions of $\langle \emptyset, D_\varphi\rangle$. Hence $\overline{\mathrm{3SAT}} \leq_{\mathrm{cd}}$ SKEP($B$), establishing the claim.

For all remaining clones $B$, observe that $[B] \subseteq \mathsf{R}_1$ or $[B] \subseteq \mathsf{M}$. Hence, Corollary 5.3 and Theorem 5.5 imply the claim. ∎

## 6. Conclusion

In this paper we provided a complete classification of the complexity of the main reasoning problems for default propositional logic, one of the most common frameworks for nonmonotonic reasoning. The complexity of the extension existence problem shows an interesting similarity to the complexity of the satisfiability problem [Lew79], because in both cases the hardest instances lie above the clone $\mathsf{S}_1$ (with the exception that instances from $\mathsf{D}$ are still hard for EXT, but easy for SAT). The complexity of the membership problems, i.e., credulous and skeptical reasoning, rests on two sources: first, whether there exist unique extensions (cf. Lemma 5.2), and second, how hard it is to test for formula implication. For this reason, we also classified the complexity of the implication problem $\mathrm{IMP}(B)$.

In the light of our present contribution, it is interesting to remark that by results of Konolige [Kon88], propositional default logic and Moore's autoepistemic logic are essentially equivalent. Even more, the translation is efficiently computable. Unfortunately, this translation requires a complete set of Boolean connectives, whence our results do not immediately transfer to autoepistemic logic. It is nevertheless interesting to ask whether the exchange of default rules with the introspective operator $L$ yields further efficiently decidable fragments.

## Acknowledgements

We thank Ilka Schnoor for sending us a manuscript with the proof of the NP-hardness of $\mathrm{EXT}(B)$ for all $B$ such that $\mathsf{N}_2 \subseteq [B]$. We also acknowledge helpful discussions on various topics of this paper with Peter Lohmann.

## References

[BCRV03] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. *SIGACT News*, 34(4):38–52, 2003.

[Got92] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic Computation*, 2(3):397–425, 1992.

[Kon88] K. Konolige. On the relation between default and autoepistemic logic. *Artificial Intelligence*, 35(3):343–382, 1988. Erratum: *Artificial Intelligence*, 41(1):115.

[KS91] H. A. Kautz and B. Selman. Hard problems for simple default logics. *Artificial Intelligence*, 49:243–279, 1991.

[Lew79] H. Lewis. Satisfiability problems for propositional calculi. *Mathematical Systems Theory*, 13:45–53, 1979.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

[Pos41] E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.

[Rei80] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[Rei03] S. Reith. On the complexity of some equivalence problems for propositional calculi. In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science*, pages 632–641, 2003.

[Sel82] A. L. Selman. Reductions on NP and p-selective sets. *Theoretical Computer Science*, 19:287–304, 1982.

[SI90] R. Sridhar and S. Iyengar. Efficient parallel algorithms for functional dependency manipulations. In *Proceedings of the 2nd International Symposium on Databases in Parallel and Distributed Systems*, pages 126–137, 1990.

[Sti90] J. Stillman. It's not my default: The complexity of membership problems in restricted propositional default logics. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 571–578, 1990.

[Sti92] J. Stillman. The complexity of propositional default logics. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 794–800, 1992.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity – A Uniform Approach*. Texts in Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 1999.

## Appendix A. The Complexity of the Implication Problem

Let $B$ be a finite set of Boolean functions. The *Implication Problem* for $B$-formulae is defined as

| | |
|---|---|
| *Problem*: | $\mathrm{IMP}(B)$ |
| *Instance*: | A set $A$ of $B$-formulae and a $B$-formula $\varphi$. |
| *Question*: | Does $A \models \varphi$ hold? |

We have already stated the classification of the complexity of $\mathrm{IMP}(B)$ in Theorem 3.1 (cf. also Fig. 3):

**Theorem 3.1.** *Let $B$ be a finite set of Boolean functions. Then* $\mathrm{IMP}(B)$ *is* coNP*-complete if* $\mathsf{S}_{00} \subseteq [B]$, $\mathsf{S}_{10} \subseteq [B]$ *or* $\mathsf{D}_2 \subseteq [B]$, *and in* P *for all other cases.*

We split the proof of Theorem 3.1 into several lemmas.

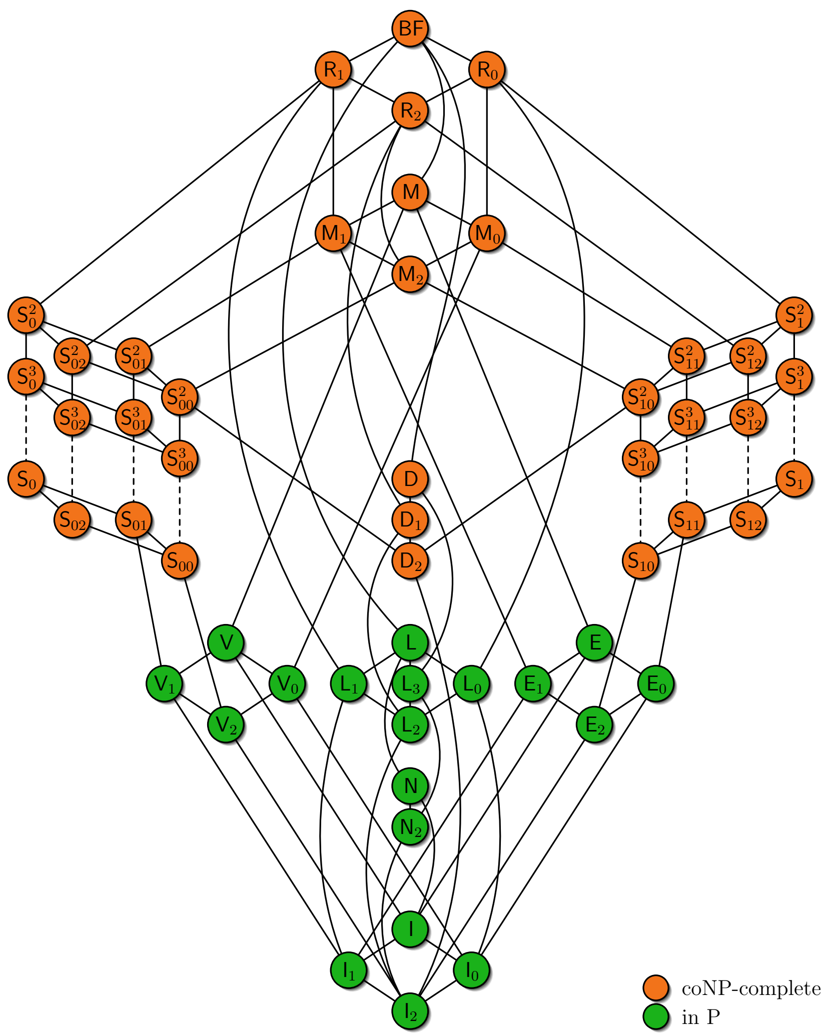


Figure 3: Post's lattice. Colors indicate the complexity of $\mathrm{IMP}(B)$, the Implication Problem for $B$-formulae.

**Lemma A.1.** *Let $B$ be a finite set of Boolean functions. The implication problem for propositional $B$-formulae,* $\mathrm{IMP}(B)$*, is* coNP*-complete if* $\mathsf{S}_{00} \subseteq [B]$ *or* $\mathsf{S}_{10} \subseteq [B]$.

*Proof.* Membership in coNP is apparent, because given $A$ and $\varphi$, we just have to check that for all assignments $\sigma$ to the variables of $A$ and $\varphi$, either $\sigma \not\models A$ or $\sigma \models \varphi$.

The hardness proof is inspired by [Rei03]. Observe that $\mathrm{IMP}(B) \equiv_{\mathrm{cd}} \mathrm{IMP}(B \cup \{1\})$ if $\wedge \in [B]$, and that $\mathrm{IMP}(B) \equiv_{\mathrm{cd}} \mathrm{IMP}(B \cup \{0\})$ if $\vee \in [B]$ (because $\varphi \models \psi \iff \varphi_{[1/t]} \wedge t \models \psi_{[1/t]}$ and $\varphi \models \psi \iff \varphi_{[0/f]} \models \psi_{[0/f]} \vee f$ where $t, f$ are new variables). It hence suffices to show that $\mathrm{IMP}(B)$ is coNP-hard for $\mathsf{M}_0 = [\mathsf{S}_{00} \cup \{0\}]$ and $\mathsf{M}_1 = [\mathsf{S}_{10} \cup \{1\}]$. We will show that $\mathrm{IMP}(B)$ is coNP-hard for each base $B$ with $\mathsf{M}_2 \subseteq [B]$. To prove this claim, we will provide a reduction from $\mathrm{TAUT}_{\mathrm{DNF}}$ to $\mathrm{IMP}(B)$, where $\mathrm{TAUT}_{\mathrm{DNF}}$ is the coNP-complete problem to decide, whether a given propositional formula in disjunctive normal form is a tautology.

Let $\varphi$ be a propositional formula in disjunctive normal form over the propositions $X = \{x_1, \ldots, x_k\}$. Then $\varphi = \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} l_{ij}$, where $l_{ij}$ are literals over $X$. We take new variables $Y = \{y_1, \ldots, y_k\}$ and replace in $\varphi$ each negative literal $l_{ij} = \neg x_l$ by $y_l$. Let $\varphi'$ be the resulting formula. Now define $\psi_1 := \bigwedge_{i=1}^{k} (x_i \vee y_i)$ and $\psi_2 := \varphi'$. We claim that $\varphi \in \mathrm{TAUT}_{\mathrm{DNF}} \iff \psi_1 \models \psi_2$.

Let us first assume $\varphi \in \mathrm{TAUT}_{\mathrm{DNF}}$ and let $\sigma\colon X \cup Y \to \{0,1\}$ be an assignment such that $\sigma \models \psi_1$. As $\varphi$ is a tautology, $\sigma \models \varphi$. But also $\sigma \models \varphi'$, as we simply replaced the negated variables in $\varphi$ by positive ones and $\varphi'$ is monotone. As $\sigma$ was arbitrarily chosen, $\psi_1 \models \psi_2$.

For the opposite direction, let $\varphi \notin \mathrm{TAUT}_{\mathrm{DNF}}$. Then there exists an assignment $\sigma\colon X \to \{0,1\}$ such that $\sigma \not\models \varphi$. We extend $\sigma$ to an assignment $\sigma'\colon X \cup Y \to \{0,1\}$ by setting $\sigma'(y_i) = 1 - x_i$ for $i = 1, \ldots, k$. Then $\sigma'(x_i) = 0$ iff $\sigma'(y_i) = 1$, and consequently $\sigma'$ simulates $\sigma$ on $\varphi'$. As a result, $\sigma' \not\models \varphi' = \psi_2$. Yet, either $\sigma'(x_l) = 1$ or $\sigma'(y_l) = 1$ and thus $\sigma' \models \psi_1$, yielding $\psi_1 \not\models \psi_2$. ∎

**Lemma A.2.** *Let $B$ be a finite set of Boolean functions. Then* $\mathrm{IMP}(B)$ *is* coNP*-complete if* $\mathsf{D}_2 \subseteq [B]$.

*Proof.* Again we just have to argue for coNP-hardness of $\mathrm{IMP}(B)$. As $\mathsf{D}_2 \subseteq [B]$, we know that $g(x,y,z) := (x \wedge y) \vee (y \wedge z) \vee (x \wedge z) \in [B]$. Clearly, $g(x,y,0) = x \wedge y$ and $g(x,y,1) = x \vee y$. Denote by $\psi_i^B$, $i \in \{1,2\}$, the formula $\psi_i$ with all occurrences of $x \wedge y$ and $x \vee y$ replaced with a $B$-representation of $g(x,y,f)$ and $g(x,y,t)$, resp.

We give a reduction from the general coNP-hard implication problem $\mathrm{IMP}(B)$ for $[B] = \mathsf{BF}$ to $\mathrm{IMP}(B)$ for $[B] \subseteq \mathsf{D}_2$ by a modification of the reduction given in the proof of Lemma A.1. We map a pair $(\psi_1, \psi_2)$ of propositional formulae to $(\{\psi_1', t\}, \psi_2')$ where

$$\psi_1' := g(g(\psi_1^B, t, f), f, t) \text{ and } \psi_2' := g(g(\psi_2^B, t, f), f, t).$$

As the variables $x$ and $y$ may occur several times in $g$, $\psi_1^B$ and $\psi_2^B$ might be exponential in the length of $\varphi$ (recall that $\psi_2$ is $\varphi$ with all negative literals replaced by new variables). That this is not the case follows from the associativity of $\wedge$ and $\vee$: insert parentheses in such a way that we get a tree of logarithmic depth. We claim that $\psi_1 \models \psi_2 \iff \{\psi_1', t\} \models \psi_2'$.

Let $\sigma$ be an arbitrary assignment for the variables in $\varphi$. Then $\sigma$ may be extended to $\{f, t\}$ in the following ways:

$\sigma(t) := 0$: Then $\{\psi_1', t\} \equiv 0$ and $\{\psi_1', t\} \models \psi_2'$.

$\sigma(t) := 1$, $\sigma(f) := 1$: In this case, $g(g(\psi_1^B, t, f), f, t) \equiv 1 \equiv g(g(\psi_2^B, t, f), f, t)$ and thus $\{\psi_1', t\} \models \psi_2'$.

$\sigma(t) := 1$, $\sigma(f) := 0$: Then $\psi_1' = g(g(\psi_1^B, t, f), f, t) \equiv (\psi_1^B \wedge t) \vee f \equiv \psi_1$ and $\psi_2' = g(g(\psi_2^B, t, f), f, t) \equiv (\psi_2^B \wedge t) \vee f \equiv \psi_2$. Thus $\psi_1 \models \psi_2$ iff $\{\psi_1', t\} \models \psi_2'$.

Hence, as claimed, $\psi_1 \models \psi_2 \iff \{\psi_1', t\} \models \psi_2'$. ■

**Lemma A.3.** *Let $B$ be a finite set of Boolean functions such that $[B] \subseteq \mathsf{V}$, $[B] \subseteq \mathsf{E}$ or $[B] \subseteq \mathsf{L}$. Then* $\mathrm{IMP}(B)$ *is in* P.

*Proof.* Consider the case $[B] \subseteq \mathsf{V}$ first. Let $B$ be a finite set of Boolean functions such that $[B] \subseteq \mathsf{V}$. Let $A$ be a finite set of $B$-formulae and let $\varphi$ be a $B$-formula such that $A$ and $\varphi$ only use the variables $x_1, \ldots, x_n$. Let $\varphi \equiv c_0 \vee c_1x_1 \vee \cdots \vee c_nx_n$ with constants $c_i \in \{0,1\}$ for $0 \le i \le n$. Equally, every formula from $A$ is equivalent to an expression of the form $c_0' \vee c_1'x_1 \vee \cdots \vee c_n'x_n$ with $c_i' \in \{0,1\}$. Then, $A \models \varphi$ iff either $c_0 = 1$ or there exists a formula $\psi \equiv c_0' \vee c_1'x_1 \vee \cdots \vee c_n'x_n$ from $A$ such that $c_i' \le c_i$ for all $0 \le i \le n$.

The value of $c_0$ can be determined by evaluating $\varphi(0, \ldots, 0)$. Furthermore, for $1 \le i \le n$, $c_i = 0$ iff $c_0 = 0$ and

$$\varphi(\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0) = 0.$$

The values of the coefficients of formulae in $A$ can be computed analogously. Thus $\mathrm{IMP}(B)$ can be computed in constant depth using oracle gates for $B$-formula evaluation. As $B$-formula evaluation is in P, the claim follows.

In the case $[B] \subseteq \mathsf{E}$, the proof is analogous to the above proof for $[B] \subseteq \mathsf{V}$.

For the remaining case, let $B$ be a finite set of Boolean functions with $[B] \subseteq \mathsf{L}$. In order to show that $B$-implication is polynomial-time solvable, observe that $F \models \alpha$ iff $F \cup \{\alpha \oplus 1\}$ is inconsistent. Let $F'$ denote $F \cup \{\alpha \oplus 1\}$ rewritten such that for all $\varphi \in F'$,

$$\varphi = c_0 \oplus c_1x_1 \oplus \cdots \oplus c_nx_n,$$

where $c_0, \ldots, c_n \in \{0,1\}$. $F'$ is polynomial-time constructible, since $c_0$ can be determined by evaluating $\varphi(0, \ldots, 0)$, and for $i = 1, \ldots, n$, $c_i = 1$ iff

$$\varphi(0, \ldots, 0) \not\equiv \varphi(\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0).$$

$F'$ can now be transformed into a system of linear equations $S$ via

$$c_0 \oplus c_1x_1 \oplus \cdots \oplus c_nx_n \;\mapsto\; c_0 + c_1x_1 + \cdots + c_nx_n = 1 \pmod 2.$$

Clearly, the resulting system of linear equations has a solution iff $F'$ is consistent. The equations are furthermore defined over the field $\mathbb{Z}_2$ and can thence be solved in polynomial time using the Gaussian algorithm. Thus $B$-implication can by solved in polynomial time. ■