Complexity of Non-Monotonic Logics

Michael Thomas and Heribert Vollmer*

1 Introduction

Over the past few decades, non-monotonic reasoning has developed to be one of the most important topics in computational logic and artificial intelligence. The non-monotonicity here refers to the fact that, while in usual (monotonic) reasoning adding more axioms leads to potentially more possible conclusions, in non-monotonic reasoning adding new facts to a knowledge base may prevent previously valid conclusions. Different ways to introduce non-monotonic aspects to classical logic have been considered:

(1) The derivation process may be extended using non-monotonic inference rules.

(2) The logical language may be extended with non-monotonic belief operators.

(3) The definition of semantics may be changed.

In this survey we consider a logical formalism from each of the above possibilities, namely

- Reiter's *default logic* (as a candidate for possibility (1) above), which introduces default inference rules of the form $\frac{\alpha:\beta}{\gamma}$, where $\frac{\alpha:\beta}{\gamma}$ intuitively expresses that γ can be derived from α as long as β is consistent with our knowledge;
- Moore's *autoepistemic logic* (a candidate for (2)), that extends classical logic with a modal operator L to express the beliefs of an ideal rational agent, in the sense that $L\varphi$ expresses that φ is provable;

*Leibniz Universität Hannover, Institut für Theoretische Informatik, {thomas,vollmer}@thi.uni-hannover.de. Work partially supported by DFG grant VO 630/6-2.

– McCarthy's *circumscription* (as candidate for (3)), which restricts the semantics to the minimal models of a formula or set of formulae.

Additionally we survey *abduction*, where one is not interested in inferences from a given knowledge base but in computing possible explanations for an observation with respect to a given knowledge base.

Complexity results for different reasoning tasks for propositional variants of these logics have been studied already in the nineties. It was shown that in each case, the complexity is higher than for usual propositional logic (typically complete for some level of the polynomial-time hierarchy). In recent years, however, a renewed interest in complexity issues can be observed. One current focal approach is to consider parameterized problems and identify reasonable parameters that allow for FPT algorithms. In another approach, the emphasis lies on identifying fragments, *i.e.*, restriction of the logical language, that allow more efficient algorithms for the most important reasoning tasks.

In this survey we focus on this second aspect. We describe complexity results for fragments of logical languages obtained by either restricting the allowed set of operators (*e.g.*, forbidding negations one might consider only monotone formulae) or by considering only formulae in conjunctive normal form but with generalized clause types (which are also called Boolean constraint satisfaction problems).

The algorithmic problems we consider are suitable variants of satisfiability and implication in each of the logics, but also certain *counting problems*, where one is not only interested in the existence of certain objects (*e.g.*, models of a formula) but asks for their number.

2 Post's Lattice

In 1941, Post showed that the sets of Boolean functions closed under projections and arbitrary composition, called *clones*, form a lattice containing only countably infinite such closed sets, and he identified a finite base for each of them [55]. The closure operation, denoted by $[\cdot]$, is not arbitrarily chosen but rather captures an intuitive understanding of expressiveness: Given a set B of Boolean functions, $[B]$ denotes the set of Boolean functions expressible using functions from B, or equivalently: computable with Boolean circuits with gates performing functions from B. Moreover, it is well behaved with respect to computational complexity; *e.g.*, if $\Pi(B)$ is a decision problem defined over Boolean circuits with gates corresponding to Boolean functions from B, then $\Pi(B) \leq_m^{\log} \Pi(B')$ for all finite sets B' of Boolean functions such that all functions from B can be expressed in B', *i.e.*, $B \subseteq [B']$. Similar statements hold

for decision problems defined over Boolean formulae (see [65]). *Post's lattice* thus holds the key to study and classify the computational complexity of problems parameterized by finite sets of available Boolean functions. In this section, we will define the required terms and notation to introduce Post's lattice.

Let $\mathcal{L}$ be the set of propositional formulae, *i.e.*, the set of formulae defined via

$$\varphi ::= a \mid f(\varphi, \ldots, \varphi),$$

where a is a proposition and f is an n-ary Boolean function (we do not distinguish between connectives and their associated functions). For a finite set B set of Boolean functions, a *B-formula* is a Boolean formula using functions from B only. The set of all B-formulae is denoted by $\mathcal{L}(B)$.

A *clone* is a set of Boolean functions that is closed under superposition, *i.e.*, B contains all projections (that is, the functions $\mathrm{I}_m^n(x_1, \ldots, x_n) = x_m$ for $n \in \mathbb{N}$ and $1 \leq m \leq n$) and is closed under arbitrary composition [54]. For a set B of Boolean functions, we denote by $[B]$ the smallest clone containing B and call B a *base* for $[B]$. A B-formula g is called *B-representation* of f if $f \equiv g$.

Post showed that the set of all clones ordered by inclusion together with meet $A \wedge B = [A \cap B]$ and join $A \vee B = [A \cup B]$ forms the lattice depicted in Figure 1. To give the list of all the clones, we need the following properties. Say that a set $A \subseteq \{0,1\}^n$ is *c-separating*, $c \in \{0,1\}$, if there exists an $i \in \{1, \ldots, n\}$ such that $(a_1, \ldots, a_n) \in A$ implies $a_i = c$. Let f be an n-ary Boolean function and define the dual of f to be the Boolean function $\mathrm{dual}(f)(x_1, \ldots, x_n) := \neg f(\neg x_1, \ldots, \neg x_n)$. We say that:

- f is *c-reproducing* if $f(c, \ldots, c) = c$, $c \in \{0,1\}$;
- f is *c-separating* if $f^{-1}(c)$ is c-separating, $c \in \{0,1\}$;
- f is *c-separating of degree m* if all $A \subseteq f^{-1}(c)$ with $|A| = m$ are c-separating;
- f is *monotone* if $a_1 \leq b_1, \ldots, a_n \leq b_n$ implies $f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n)$;
- f is *self-dual* if $f \equiv \mathrm{dual}(f)$; here, $\mathrm{dual}(f)(x_1, \ldots, x_n) = \neg f(\neg x_1, \ldots, \neg x_n)$;
- f is *affine* if $f(x_1, \ldots, x_n) \equiv x_1 \oplus \cdots \oplus x_n \oplus c$ with $c \in \{0,1\}$;
- f is *essentially unary* if f depends on at most one variable.

The above properties canonically extend to sets of Boolean functions. The list of all clones is given in Table 1, where id denotes the identity I_1^1 and T_n^{n+1} is the $(n+1)$-ary threshold function with threshold n defined as $\mathrm{T}_n^{n+1}(x_0, \ldots, x_n) :=$

Clone	Definition	Base
BF	All Boolean functions	$\{x \wedge y, \neg x\}$
R_0	$\{f \in \mathsf{BF} \mid f$ is 0-reproducing$\}$	$\{x \wedge y, x \oplus y\}$
R_1	$\{f \in \mathsf{BF} \mid f$ is 1-reproducing$\}$	$\{x \vee y, x \leftrightarrow y\}$
R_2	$\mathsf{R}_0 \cap \mathsf{R}_1$	$\{x \vee y, x \wedge (y \leftrightarrow z)\}$
M	$\{f \in \mathsf{BF} \mid f$ is monotone$\}$	$\{x \wedge y, x \vee y, 0, 1\}$
M_0	$\mathsf{M} \cap \mathsf{R}_0$	$\{x \wedge y, x \vee y, 0\}$
M_1	$\mathsf{M} \cap \mathsf{R}_1$	$\{x \wedge y, x \vee y, 1\}$
M_2	$\mathsf{M} \cap \mathsf{R}_2$	$\{x \wedge y, x \vee y\}$
S_0	$\{f \in \mathsf{BF} \mid f$ is 0-separating$\}$	$\{x \rightarrow y\}$
S_0^n	$\{f \in \mathsf{BF} \mid f$ is 0-separating of degree $n\}$	$\{x \rightarrow y, \mathrm{dual}(\mathrm{T}_n^{n+1})\}$
S_1	$\{f \in \mathsf{BF} \mid f$ is 1-separating$\}$	$\{x \nrightarrow y\}$
S_1^n	$\{f \in \mathsf{BF} \mid f$ is 1-separating of degree $n\}$	$\{x \nrightarrow y, \mathrm{T}_n^{n+1}\}$
S_{02}^n	$\mathsf{S}_0^n \cap \mathsf{R}_2$	$\{x \vee (y \wedge \neg z), \mathrm{dual}(\mathrm{T}_n^{n+1})\}$
S_{02}	$\mathsf{S}_0 \cap \mathsf{R}_2$	$\{x \vee (y \wedge \neg z)\}$
S_{01}^n	$\mathsf{S}_0^n \cap \mathsf{M}$	$\{\mathrm{dual}(\mathrm{T}_n^{n+1}), 1\}$
S_{01}	$\mathsf{S}_0 \cap \mathsf{M}$	$\{x \vee (y \wedge z), 1\}$
S_{00}^n	$\mathsf{S}_0^n \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \vee (y \wedge z), \mathrm{dual}(\mathrm{T}_n^{n+1})\}$
S_{00}	$\mathsf{S}_0 \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \vee (y \wedge z)\}$
S_{12}^n	$\mathsf{S}_1^n \cap \mathsf{R}_2$	$\{x \wedge (y \vee \neg z), \mathrm{T}_n^{n+1}\}$
S_{12}	$\mathsf{S}_1 \cap \mathsf{R}_2$	$\{x \wedge (y \vee \neg z)\}$
S_{11}^n	$\mathsf{S}_1^n \cap \mathsf{M}$	$\{\mathrm{T}_n^{n+1}, 0\}$
S_{11}	$\mathsf{S}_1 \cap \mathsf{M}$	$\{x \wedge (y \vee z), 0\}$
S_{10}^n	$\mathsf{S}_1^n \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \wedge (y \vee z), \mathrm{T}_n^{n+1}\}$
S_{10}	$\mathsf{S}_1 \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \wedge (y \vee z)\}$
D	$\{f \in \mathsf{BF} \mid f$ is self-dual$\}$	$\{(x \wedge y) \vee (x \wedge \neg z) \vee (\neg y \wedge \neg z)\}$
D_1	$\mathsf{D} \cap \mathsf{R}_2$	$\{(x \wedge y) \vee (x \wedge \neg z) \vee (y \wedge \neg z)\}$
D_2	$\mathsf{D} \cap \mathsf{M}$	$\{(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)\}$
L	$\{f \in \mathsf{BF} \mid f$ is affine$\}$	$\{x \oplus y, 1\}$
L_0	$\mathsf{L} \cap \mathsf{R}_0$	$\{x \oplus y\}$
L_1	$\mathsf{L} \cap \mathsf{R}_1$	$\{x \leftrightarrow y\}$
L_2	$\mathsf{L} \cap \mathsf{R}_2$	$\{x \oplus y \oplus z\}$
L_3	$\mathsf{L} \cap \mathsf{D}$	$\{x \oplus y \oplus z \oplus 1\}$
E	$\{f \in \mathsf{BF} \mid f$ is constant or a conjunction$\}$	$\{x \wedge y, 0, 1\}$
E_0	$\mathsf{E} \cap \mathsf{R}_0$	$\{x \wedge y, 0\}$
E_1	$\mathsf{E} \cap \mathsf{R}_1$	$\{x \wedge y, 1\}$
E_2	$\mathsf{E} \cap \mathsf{R}_2$	$\{x \wedge y\}$
V	$\{f \in \mathsf{BF} \mid f$ is constant or a disjunction$\}$	$\{x \vee y, 0, 1\}$
V_0	$\mathsf{V} \cap \mathsf{R}_0$	$\{x \vee y, 0\}$
V_1	$\mathsf{V} \cap \mathsf{R}_1$	$\{x \vee y, 1\}$
V_2	$\mathsf{V} \cap \mathsf{R}_2$	$\{x \vee y\}$
N	$\{f \in \mathsf{BF} \mid f$ is essentially unary$\}$	$\{\neg x, 0, 1\}$
N_2	$\mathsf{N} \cap \mathsf{D}$	$\{\neg x\}$
I	$\{f \in \mathsf{BF} \mid f$ is constant or a projection$\}$	$\{\mathrm{id}, 0, 1\}$
I_0	$\mathsf{I} \cap \mathsf{R}_0$	$\{\mathrm{id}, 0\}$
I_1	$\mathsf{I} \cap \mathsf{R}_1$	$\{\mathrm{id}, 1\}$
I_2	$\mathsf{I} \cap \mathsf{R}_2$	$\{\mathrm{id}\}$

Table 1: List of all clones with definition and bases

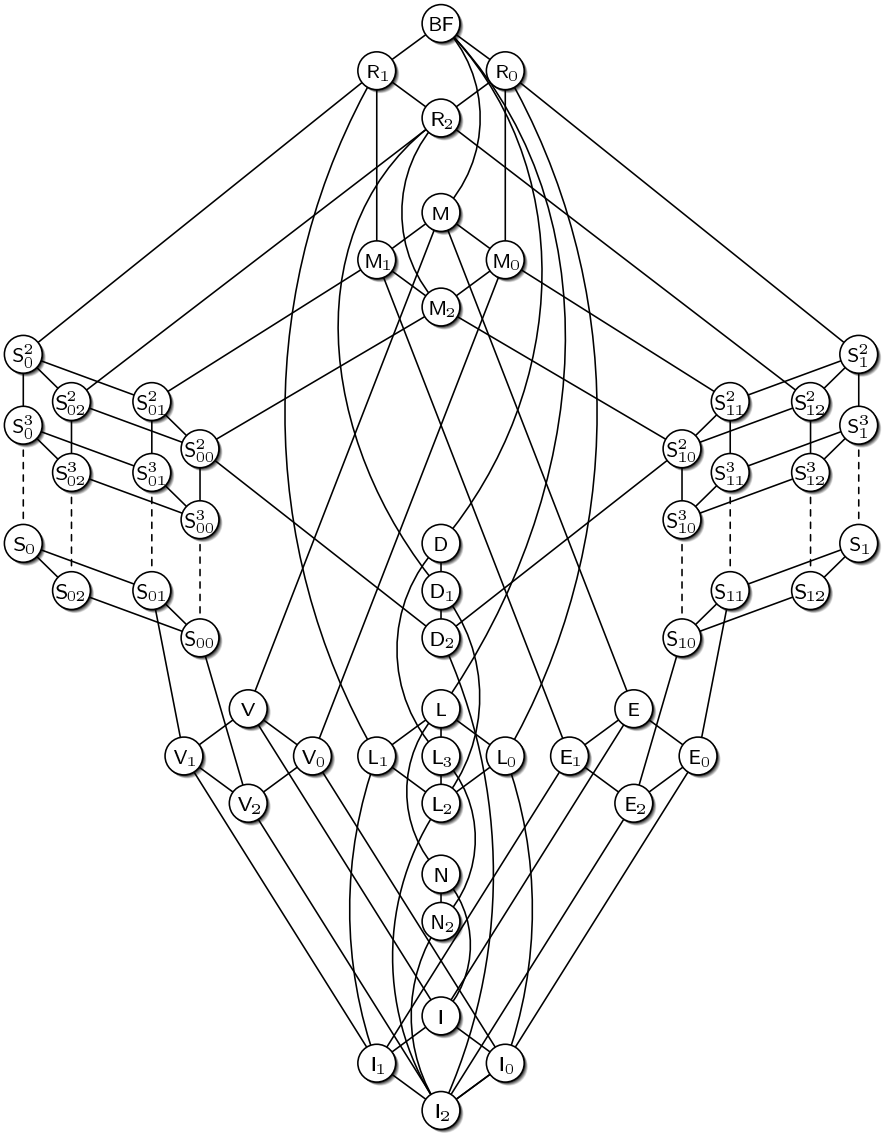

Figure 1: Post's lattice

$\bigvee_{i=0}^{n}(x_0 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_n)$, *i.e.*, $\mathrm{T}_n^{n+1}(x_0, \ldots, x_n)$ is 1 if at least n of its inputs are 1.

The following easy observation, which will be useful in the subsequent sections, gives a a first example of the relationship between function-restricted sets of Boolean formulae.

Lemma 2.1 *Let B be a finite set of Boolean functions and let L be a set of $(B \cup \{1\})$-formulae. Then L can be transformed in logspace into a set L' of B-formulae such that the number of satisfying assignments of L and L' coincide.*

Proof sketch. The idea is to add a fresh proposition t to L and to replace all occurrences of the constant 1 with t. As a result we obtain that, for problems Π defined over sets of Boolean formulae, $\Pi(B \cup \{1\}) \equiv_m^{\log} \Pi(B)$. □

3 Default Logic

Default logic is among the best known and most successful formalisms for non-monotonic reasoning. It was proposed by Raymond Reiter in 1980 [56] and extends classical logic with *default rules*, *i.e.*, defeasible inference rules with an additional justification. These capture the process of deriving conclusions based on inferences of the form "in the absence of contrary information, assume ...". As with few exceptions most of our knowledge about the world is almost true rather than an absolute truth, Reiter argued that his logic is an adequate formalization of the human reasoning under the *closed world assumption*, which allows one to assume the negation of all facts not derivable from the given knowledge base.

Formally, a default rule is an expression of the form $\frac{\alpha:\beta}{\gamma}$, where α, β, γ are formulae; α is called the *premise*, β is called *justification*, and γ is called *conclusion*. Further, a *default theory* is a pair (W, D), where W is a set of formulae and D is a set of default rules. The intended interpretation of default rules is that γ holds if α can be derived and β is consistent with our knowledge and beliefs about the world. It is this consistency condition that introduces the non-monotonicity:

Example 3.1 Consider the default theory (W, D) with $W := \{x\}, D := \left\{\frac{x:y}{z}\right\}$. Then we should be able to derive z from (W, D), as x is a fact from W and y is consistent with the consequences of $W \cup \{z\}$. However, if W is extended with $\{\neg y\}$ then W is no longer consistent with the justification of $\frac{x:y}{z}$ and we have no right to conclude z. The addition of $\neg y$ to W thus invalidates the consequence z.

As another consequence, the derivable knowledge depends on the set of applied defaults:

Example 3.2 Consider the default theory $(\emptyset, D)$ with $D := \left\{ \frac{1:x}{\neg y}, \frac{1:y}{\neg x} \right\}$. If we apply the left default, then $\neg y$ is derivable while the right default is blocked, as its justifications is inconsistent with the conclusion $\neg y$. On the other hand, if we apply the right default first, then $\neg x$ is derived and the left default rule gets blocked.

Thus, to appropriately represent the knowledge derivable from a default theory, we introduce the notion of stable extensions.

Definition 3.3 ([56]) Let (W, D) be a default theory and E be a set of formulae. 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 (W, D) if and only if $E = \bigcup_{i \in \mathbb{N}} E_i$.

Stable extensions can alternatively be characterized as the least fixed points of an operator $\Gamma_{W,D}$: For a given default theory (W, D) and a set E of formulae, let $\Gamma_{W,D}(E)$ be the smallest set of formulae such that

1. $W \subseteq \Gamma(E)$,
2. $\Gamma(E)$ is deductively closed (*i.e.*, $\Gamma_{W,D}(E) = \mathrm{Th}(\Gamma_{W,D}(E))$), and
3. for all $\frac{\alpha:\beta}{\gamma} \in D$ with $\alpha \in \Gamma_{W,D}(E)$ and $\neg\beta \notin E$, it holds that $\gamma \in \Gamma_{W,D}(E)$
   (in this case, we also say that the default $\frac{\alpha:\beta}{\gamma}$ is *applicable*).

Proposition 3.4 ([56]) *Let (W, D) be a default theory and a E be a set of formulae. Then E is a stable extension of (W, D) iff E is a fixed point of $\Gamma_{W,D}$.*

We have already observed in Example 3.2 that a default theory may possess several stable extensions; indeed, a default theory with n default rules may have any number of different stable extensions between 0 and 2^n. Thus the following questions naturally arise:

(*Extension existence*) Does a given default theory admit a stable extension?

(*Credulous reasoning*) Is a given formula contained in at least one stable extension of a given default theory?

(*Skeptical reasoning*) And, finally, is a given formula contained in all stable extensions of a given default theory?

The computational complexity of the corresponding decision problems was first explored by Kautz and Selman [38], and Stillman [61], who both

presented results for syntactically restricted fragments of disjunction-free default logics. In 1992, Gottlob [24] and Stillman [62] then independently showed that the computational complexity of these questions is presumably higher than that of the corresponding satisfiability and implication problem in propositional logic:

Theorem 3.5 ([24, 62]) *The extension existence problem and the credulous reasoning problem for default logic are Σ_2^{p}-complete, whereas the skeptical reasoning problem for default logic is Π_2^{p}-complete.*

More recently, Liberatore and Schaerf [41] showed that model checking (*i.e.*, the task to decide whether a given assignment is a model of any extension of a given default theory) is Σ_2^{p}-complete, too. And Ben-Eliyahu-Zohary [4] extended the complexity landscape of default logics with results on disjunction-free fragments dual to those studied by Kautz and Selman, and Stillman. The study of these fragments was motivated by embeddings of other formalisms into default logic. However, little was known about the complexity of not-disjunction-free default logics. In [7], the authors devise a systematic study of the fragments of default logic obtained by restricting the set of available Boolean functions. The results provide insight into the source of the hardness of default reasoning and reveal the trade-off between expressivity and computational complexity of fragments of default logic.

To present the results, let B be a finite set of Boolean functions. Say that the default theory (W, D) is a *B-default theory* if $W \cup \left\{ \alpha, \beta, \gamma \;\middle|\; \frac{\alpha:\beta}{\gamma} \right\} \subseteq \mathcal{L}(B)$ and let B-default logic denote default logic restricted to B-default theories.

Theorem 3.6 ([7]) *Let B be a finite set of Boolean functions. Then the extension existence problem for B-default logic is*

1. *Σ_2^{p}-complete if $\mathsf{S}_1 \subseteq [B]$ or $\mathsf{D} \subseteq [B]$,*
2. *Δ_2^{p}-complete if $\mathsf{S}_{11} \subseteq [B] \subseteq \mathsf{M}$,*
3. *NP-complete if $[B] \in \{\mathsf{N}, \mathsf{N}_2, \mathsf{L}, \mathsf{L}_0, \mathsf{L}_3\}$,*
4. *P-complete if $[B] \in \{\mathsf{V}, \mathsf{V}_0, \mathsf{E}, \mathsf{E}_0\}$,*
5. *NL-complete if $[B] \in \{\mathsf{I}, \mathsf{I}_0\}$, and*
6. *trivial in all other cases (that is, if $[B] \subseteq \mathsf{R}_1$),*

via logspace many-one reductions.

A key observation to the proof of this theorem is the following lemma.

Lemma 3.7 *Let B be a finite set of Boolean functions. If $[B] \subseteq \mathsf{M}$ then any B-default theory has at most one stable extension; if $[B] \subseteq \mathsf{R}_1$ then any B-default theory has at exactly one stable extension.*

Proof. Suppose first that $[B] \subseteq \mathsf{M}$. Then each function $f \in B$ is either 1-reproducing or equivalent to 0. Default rules with a justification equivalent to 0 are not applicable unless W is inconsistent. As in this case $\mathcal{L}$ is the only stable extension of the default theory [56, Corollary 2.3], we w.l.o.g. suppose that all justifications are 1-reproducing, *i.e.*, $[B] \subseteq \mathsf{R}_1$. Now observe that the negation of a 1-reproducing function is not 1-reproducing while all consequents of 1-reproducing functions are. Indeed, if $[B] \subseteq \mathsf{R}_1$ then any stable extension of a default theory is consistent and satisfied by the assignment setting to 1 all propositions, which also satisfies the justification of each default rule. Thus any monotone default theory may possess at most one stable extension while any 1-reproducing default theory possesses exactly one stable extension. □

We will sketch the proof of Theorem 3.6.

Proof sketch. For $\mathsf{S}_1 \subseteq [B]$ or $\mathsf{D} \subseteq [B]$, the Σ_2^{p}-hardness follows from Theorem 3.5 and Lemma 2.1, as $[\mathsf{S}_1 \cup \{1\}] = [\mathsf{D} \cup \{1\}] = \mathsf{BF}$ and the upper bound easily generalizes from $\{\wedge, \vee, \neg\}$ to arbitrary sets of Boolean functions.

The case $[B] \subseteq \mathsf{R}_1$ follows directly from Lemma 3.7. It hence remains to consider those sets B such that $[B \cup \{1\}]$ contains the constant 0. These are all included in either the clone M or the clone L (or both).

For $\mathsf{S}_{11} \subseteq [B] \subseteq \mathsf{M}$, membership in Δ_2^{p} follows similarly from Lemmas 2.1 and 3.7: the only way for a monotone default theory not to possess a stable extension is to contain a default rule $\frac{\alpha:\beta}{\gamma}$ such that $\gamma \equiv 0$. It thus suffices to compute the set of applicable defaults using subsequent calls to a coNP-oracle for B-formula implication and to verify that their conclusions are satisfied by the assignment setting to 1 all propositions. It is straightforward to implement this as a Δ_2^{p}-algorithm. The Δ_2^{p}-hardness on the other hand is established using a reduction from the *sequentially nested satisfiability problem*, which was first identified to be Δ_2^{p}-complete in [25, Theorem 3.4] (see also [40]).

If one further restricts the set B such that $[B] \in \{\mathsf{V}, \mathsf{V}_0, \mathsf{E}, \mathsf{E}_0\}$ (*i.e.*, $[B \cup \{1\}]$ does contain the Boolean constants and either conjunctions or disjunctions), then formula implication and the the extension existence problem become tractable [6]. Indeed, the problem becomes P-complete, as can be shown by a reduction from a variant of the circuit value problem.

Further restricting the set B such that $[B] \in \{\mathsf{I}, \mathsf{I}_0\}$ (*i.e.*, $[B \cup \{1\}]$ does only contain the Boolean constants) leads to default theories with rules whose premise and conclusion are a single proposition. As a result, the existence of a stable extension reduces to the complement of the reachability problem in directed graphs. Similarly, reachability in such graphs can be transformed

into the question whether a stable extension does not have a stable extension. As this problem is NL-complete and NL is closed under complement, the extension existence problem for such default theories is NL-complete.

Finally, for $[B] \in \{\mathsf{L}, \mathsf{L}_0, \mathsf{L}_3, \mathsf{N}, \mathsf{N}_2\}$ (*i.e.*, $\neg \in [B \cup \{1\}]$ and $[B]$ is affine), we observe a different situation. There may exist exponentially many different stable extensions; yet, the verification of a candidate is tractable because implication and satisfiability of B-formulae are [6]. Hence, the extension existence problem becomes solvable in NP. The NP-hardness, on the other hand, is obtained by reducing from the satisfiability problem for 3CNF formulae: given a formula $\varphi \equiv \bigwedge_{i=1}^{n} c_i$ with $c_i \equiv \ell_{i1} \vee \ell_{i2} \vee \ell_{i3}$, we construct the default theory $(\emptyset, D)$ with

$$D := \left\{ \frac{1 : x_i}{x_i}, \frac{1 : \neg x_i}{\neg x_i} \;\middle|\; x_i \in \mathrm{Vars}(\varphi) \right\} \cup \left\{ \frac{\overline{\ell_{i1}} : \overline{\ell_{i2}}}{\ell_{i3}} \;\middle|\; 1 \leq i \leq n \right\},$$

where, for a literal ℓ, $\overline{\ell}$ denotes the literal of opposite polarity, and for a formula φ, $\mathrm{Vars}(\varphi)$ denotes the set of all variables in φ. It is easy to verify the correctness of this reduction. As the above default theory can easily be written as a B-default theory for all B such that $\neg \in [B]$, the proof is complete. □

Remark 3.8 As default rules require the justification β to be consistent with a stable extension E (*i.e.*, $\neg\beta \notin E$), another conceivable formalization of B-default logic would be to require α and γ to be B-formulae and β to be the negation of a B-formula. For this formalization, the extension existence problem for B-default logic is Σ_2^{p}-complete if $\mathsf{S}_{00} \subseteq [B]$ or $\mathsf{S}_{10} \subseteq [B]$ or $\mathsf{D}_2 \subseteq [B]$, and tractable otherwise (with this case splitting into $\oplus$L-complete cases and logspace-solvable cases).

Given the upper and lower bounds for the stable extension problem it is easy to settle the complexity of the credulous and skeptical reasoning problem. Define the *credulous (resp. skeptical) reasoning problem for B-default logic* as the problem to decide, given a B-default theory (W, D) and a B-formula φ, whether φ is contained in a stable extension (resp. all stable extensions) of (W, D).

Theorem 3.9 ([7]) *Let B be a finite set of Boolean functions. Then the credulous (resp. skeptical) reasoning problem for B-default logic is*

1. *Σ_2^{p}-complete (resp. Π_2^{p}-complete) if $\mathsf{S}_1 \subseteq [B]$ or $\mathsf{D} \subseteq [B]$,*
2. *Δ_2^{p}-complete if $\mathsf{S}_{11} \subseteq [B] \subseteq \mathsf{M}$,*
3. *coNP-complete if $\mathsf{S}_{00} \subseteq [B] \subseteq \mathsf{R}_1$ or $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{R}_1$ or $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{R}_1$,*

4. NP-*complete (resp.* coNP-*complete) if* $[B] \in \{\mathsf{N}, \mathsf{N}_2, \mathsf{L}, \mathsf{L}_0, \mathsf{L}_3\}$,
5. P-*complete if* $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$ *or* $\mathsf{E}_2 \subseteq [B] \subseteq \mathsf{E}$ *or* $[B] \in \{\mathsf{L}_1, \mathsf{L}_2\}$, *and*
6. NL-*complete in all other cases (that is, if* $[B] \subseteq \mathsf{I}$*),*

via logspace many-one reductions.

Another possibility to study fragments of default logic, aside restricting the available Boolean functions, is *Schaefer's framework* [57]. This framework is motivated by constraint satisfaction problem, where a set of conditions represented as logical relations has to be simultaneously satisfied. Hence, the set W and the formulae occurring in D are assumed to be a set of applications of relations from S to the variables in $\mathrm{Vars}(W) \cup \mathrm{Vars}(D)$, where $\mathrm{Vars}(D)$ is a shorthand for $\mathrm{Vars}\left(\left\{\alpha, \beta, \gamma \,\middle|\, \frac{\alpha:\beta}{\gamma} \in D\right\}\right)$. That is, W and the formulae occurring in D are of the form $\{R_1(x_{11}, \ldots, x_{1m_1}), \ldots, R_n(x_{n1}, \ldots, x_{nm_n})\}$, where the R_i's are relations of arity m_i from a fixed set S of available relations over the domain $\{0, 1\}$ and the variables $x_{11}, \ldots, x_{nm_n}$ are from $\mathrm{Vars}(W) \cup \mathrm{Vars}(D)$. Such a set of applications of relations is correspondingly said to be satisfied by an assignment σ if $(\sigma(x_{i1}), \ldots, \sigma(x_{im_i})) \in R_i$ for all $1 \leq i \leq n$. Call a relation R *Schaefer* if it is either

- *affine* (coincides with the set of models of an $\{\oplus\}$-formula),
- *bijunctive* (coincides with the set of models of a 2CNF formula),
- *Horn* (coincides with the set of models of a Horn formula), or
- *dual Horn* (coincides with the set of models of a dual Horn formula).

And say that a set of relations is Schaefer if there is one of the above four properties that is satisfied by all relations in S. Call a default theory (W, D) such that $W \cup \left\{\alpha, \beta, \gamma \,\middle|\, \frac{\alpha:\beta}{\gamma}\right\}$ is a set of applications of relations from S a *default theory over relations from* S.

We define the *extension existence problem for default logic over relations from* S as the problem to decide, given a default theory (W, D) over relations from S, whether (W, D) has a stable extension. Further, define the *credulous* (resp. *skeptical*) *reasoning problem for default logic over relations from* S as the problem to decide, given a default theory (W, D) over relations from S and a set φ of applications of relations from S, whether φ is contained in at least one (resp. any) stable extension of (W, D). In [10], Chapdelaine *et al.* study the complexity of these problems and establish the following trichotomies:

Theorem 3.10 ([10, 58]) *Let* S *be a set of relations. Then the extension existence problem for default logic over relations from* S *is*

1. Σ_2^{p}-*complete if* S *is not Schaefer,*
2. NP-*complete if* S *is Schaefer but neither* 0-*valid or* 1-*valid,*
3. *in* P *in all other cases.*

Theorem 3.11 ([10, 58]) *Let* S *be a set of relations. Then the credulous (resp. skeptical) reasoning problem for default logic over relations from* S *is*

1. Σ_2^{p}-*complete if* S *is not Schaefer,*
2. NP-*complete (resp.* coNP-*complete) if* S *is Schaefer but neither* 0-*valid nor* 1-*valid,*
3. coNP-*complete if* S 0-*valid or* 1-*valid but not Schaefer,*
4. *in* P *in all other cases.*

For detailed proofs of these results, see [58].

We like to remark that also results like these about Boolean constraint satisfaction problems are proved using Post's lattice. This is because the classes of Boolean relations above (affince, bijunctive, Horn, dual Horn) can all be defined using so called *polymorphism*, a kind of closure property, of Boolean relations. These sets of polymorphism are always clones, *i.e.*, they appear somewhere in the lattice. To state only one example, a relation is Horn iff the set of its polymorphisms is the class E_2. The structure of the lattice is then used in the proof to make a case distinction on all possible sets of polymorphisms of S and determine the complexity in each case. For more details, we refer the reader to [14].

Having settled the complexity of these decision problems, mind that these results do only speak about the existence of objects, *e.g.*, stable extensions. But what about the complexity of counting them? We will conclude this survey of the complexity of default logic with a treatment of the problem to count the number stable extensions.

Let us introduce the relevant notions and counting complexity classes first. For alphabets Σ and Π, let $A \subseteq \Sigma^\star \times \Pi^\star$ be a binary relation such that the set $A(x) := \{y \in \Pi^\star \mid (x, y) \in A\}$ is finite for all $x \in \Sigma^\star$. We write $\#A$ to denote the following *counting problem*: Given $x \in \Sigma^\star$, compute $|A(x)|$. The class of counting problems computable in polynomial time is denoted by FP. To characterize the complexity of counting problems that are not known to be in FP, we follow [31] and define an operator $\#\cdot\mathcal{C}$ on classes $\mathcal{C}$ of decision problems: $\#A \in \#\cdot\mathcal{C}$ if (a) there exists a polynomial p, such that for all x and all $y \in A(x)$, $|y| \leq p(|x|)$ and (b) the problem to decide, given x and y, whether $y \in A(x)$ is in $\mathcal{C}$. Clearly, $\#\cdot$P coincides with #P, the class of functions counting the number of accepting path of

nondeterministic polynomial-time Turing machines—the natural analogue of NP in the counting complexity context [68]. Applying $\#\cdot$ to the classes of the polynomial hierarchy, we now obtain a linearly ordered hierarchy of counting complexity classes [67, 31]: $\#\mathrm{P} \subseteq \#\cdot\mathrm{NP} \subseteq \#\cdot\mathrm{coNP} = \#\cdot\mathrm{P}^{\mathrm{NP}} \subseteq \#\cdot\Sigma_2^{\mathrm{p}} \subseteq \#\cdot\Pi_2^{\mathrm{p}} = \#\cdot\mathrm{P}^{\Sigma_2^{\mathrm{p}}} \subseteq \cdots$.

The counting complexity of default logic has, to the authors' best knowledge, first been considered in [66]. There, it was shown that counting the number of stable extensions is complete for the second level of the counting polynomial hierarchy, $\#\cdot\mathrm{coNP}$, whenever $[B \cup \{1\}] = \mathsf{BF}$; becomes Δ_2^{p}-complete for all sets B such that $[B \cup \{1\}] = \mathsf{M}$; complete for the first level of the counting hierarchy for all affine sets B such that $\neg$ can be implemented from $B \cup \{1\}$; and becomes efficiently computable in all other cases. The counting complexity thus decreases analogously to the complexity of the extension existence problem. However, observe that we blur over the distinction between decision problems and their characteristic functions: By Lemma 3.7 any monotone B-default theory has at most one stable extension. The problem to count the number stable extensions thus coincides with the characteristic function of the extension existence problem, which Δ_2^{p}-complete.

Theorem 3.12 ([66]) *Let B be a finite set of Boolean functions. Then the problem to count the number of stable extensions in B-default logic is*

1. *$\#\cdot\mathrm{coNP}$-complete if $\mathsf{S}_1 \subseteq [B]$ or $\mathsf{D} \subseteq [B]$,*
2. *Δ_2^{p}-complete if $\mathsf{S}_{11} \subseteq [B] \subseteq \mathsf{M}$,*
3. *$\#\mathrm{P}$-complete if $[B] \in \{\mathsf{N}, \mathsf{N}_2, \mathsf{L}, \mathsf{L}_0, \mathsf{L}_3\}$,*
4. *in FP in all other cases (that is, if $[B] \subseteq \mathsf{V}$ or $[B] \subseteq \mathsf{E}$ or $[B] \subseteq \mathsf{R}_1$)*

via parsimonious reductions.

Note that for the classification in Theorem 3.12 the conceptually simple and well-behaved parsimonious reductions are sufficient (a counting problem $\#A$ parsimoniously reduces to a counting problem $\#B$ if there is a polynomial-time computable function f such that for all inputs x, $|A(x)| = |B(f(x))|$ [68]), while for related classifications in the literature less restrictive and more complicated reductions had to be used (see, *e.g.*, [18, 28, 3]).

4 Autoepistemic Logic

Autoepistemic logic was introduced by Moore [45] to overcome some of the peculiarities of the non-monotonic logics devised by McDermott and

Doyle [44] and McDermott [43]. While Moore defined autoepistemic logic without referring to any particular modal system, it turned out that his logic coincides with the non-monotonic modal logic based on KD45 [59]. Therefore, autoepistemic logic can be considered a popular representative among the non-monotonic modal logics. The connection of these logics and particularly autoepistemic logic to default logic has been extensively studied. The first major approach in this direction was taken by Konolige [37], who showed that default logic can be embedded into autoepistemic logic using slightly different semantics for the latter. Subsequently, Marek and Truszczynski [47, 48] showed that, using strengthened notions in autoepistemic logic or weakened notions in default logic, the two logics coincide in terms of expressivity. Finally, Gottlob [26] showed that default logic can be embedded into standard autoepistemic logic, while the converse direction was shown to hold by Janhunen [32].

The intention of Moore was to create a logic modelling the beliefs of an ideally rational agent, *i.e.*, an agent that believes all things he can deduce and refutes belief in everything else. To this end, autoepistemic logic extends propositional logic with the modal operator L stating that its argument "is believed". The set of all autoepistemic formulae $\mathcal{L}_{\text{ae}}$ is defined as

$$\varphi ::= a \mid f(\varphi, \ldots, \varphi) \mid L\varphi,$$

where a is a proposition and f is a Boolean function, and the relation $\models$ is extended to simply treat formulae starting with an L as atomic. Similarly to default logic, the semantics of autoepistemic logic are defined in terms of fixed points, which in the context of autoepistemic logic are called stable expansions:

Definition 4.1 ([45]) Let $\Sigma \subseteq \mathcal{L}_{\text{ae}}$ be a set of autoepistemic formulae. A set $\Delta \subseteq \mathcal{L}_{\text{ae}}$ is a *stable expansion* of Σ if it satisfies the equation

$$\Delta = \text{Th}(\Sigma \cup L(\Delta) \cup \neg L(\mathcal{L}_{\text{ae}} \setminus \Delta)),$$

where $L(\Delta) := \{L\varphi \mid \varphi \in \Delta\}$ and $\neg L(\mathcal{L}_{\text{ae}} \setminus \Delta) := \{\neg L\varphi \mid \varphi \notin \Delta\}$.

Example 4.2 Consider the set $\Sigma = \{Lx \vee y, x \vee Ly, L(x \vee y) \rightarrow z\}$ of autoepistemic formulae. We claim that Σ has two stable expansions, each of which containing z. Sticking with our informal interpretation of autoepistemic logic as the logic of an ideally rational agent's beliefs, observe that we cannot deduce x from Σ. Hence, we would assign Lx the value 0 and consequently be able to derive y from $Lx \vee y$. This in turn allows us to conclude z from $L(x \vee y) \rightarrow z$, as $x \vee y$ is derivable from y. Indeed, the set $\bigcup_{i \in \mathbb{N}} \Delta_i$ of formulae

recursively defined via $\Delta_0 := \{x\}$ and $\Delta_i := \mathrm{Th}(\Sigma \cup L(\Delta_{i-1}) \cup \neg L(\mathcal{L}_{\mathrm{ae}} \setminus \Delta_{i-1}))$ is a stable expansion of Σ that contains z.

On the other hand, we are not able to deduce y from Σ either. Hence, we could also continue to assign to Ly the value 0 and be therefore able to derive x. Again, we may conclude z from $L(x \vee y) \rightarrow z$. And just as above, $\bigcup_{i \in \mathbb{N}} \Delta'_i$ with $\Delta'_0 := \{y\}$ and Δ'_i defined as Δ_i is a stable expansion of Σ that contains z.

There is an important difference to default logic as stable expansions need not be minimal fixed points:

Example 4.3 Consider $\Sigma' := \{Lp \rightarrow p\}$. The set Σ' has two stable expansions, one stable expansion containing $\neg Lp$ and the other one containing Lp. As an iterative construction as in Example 4.2 is deemed to fail for the latter, it may be considered *ungrounded* in the set of premises Σ.

Clearly, sets of autoepistemic formulae can also posses no or a single stable expansion. Hence, the *expansion existence problem*, the *credulous reasoning problem* and the *skeptical reasoning problem* arise just as in default logic. The first treatment of the complexity of these problems has been performed by Niemelä [49]. In his paper, he gave a finite characterization of stable expansions in terms of *full sets*: Let $\mathrm{SF}^L(\Sigma)$ denote the set of L-prefixed subformulae of formulae in Σ.

Definition 4.4 ([49]) Let $\Sigma \subseteq \mathcal{L}_{\mathrm{ae}}$ be a set of autoepistemic formulae. A set $\Lambda \subseteq \mathrm{SF}^L(\Sigma) \cup \{\neg L\varphi \mid L\varphi \in \mathrm{SF}^L(\Sigma)\}$ is Σ-*full* if for all $L\psi \in \mathrm{SF}^L(\Sigma)$:

- $\Sigma \cup \Lambda \models \psi$ iff $L\psi \in \Lambda$.
- $\Sigma \cup \Lambda \not\models \psi$ iff $\neg L\psi \in \Lambda$.

Proposition 4.5 ([49]) *Let $\Sigma \subseteq \mathcal{L}_{\mathrm{ae}}$ be a set of autoepistemic formulae. If Λ is a Σ-full set, then there exists exactly one stable expansion of Δ such that $\Lambda \subseteq L(\Delta) \cup \neg L(\mathcal{L}_{\mathrm{ae}} \setminus \Delta)$. Vice versa, if Δ is a stable expansion of Σ, then there exists exactly one Σ-full set Λ such that such that $\Lambda \subseteq L(\Delta) \cup \neg L(\mathcal{L}_{\mathrm{ae}} \setminus \Delta)$.*

Using full sets as finite representations for stable expansions, Niemelä obtained a Σ_2^{p} upper bound for the expansion existence problem and the credulous reasoning problem, and a Π_2^{p} upper bound for the skeptical reasoning problem: for the expansion existence problem, simply guess a candidate for a full set and verify the conditions given in Definition 4.4 using an oracle for formula implication. To extend this idea to the credulous and skeptical reasoning problem, one still needs to define a consequence relation $\models_L$ that, given $\Sigma \subseteq \mathcal{L}_{\mathrm{ae}}$ and a Σ-full set Λ implies exactly those formulae contained

in the stable expansion corresponding to Λ. Niemelä shows that $\models_L$ can be defined such that the problem of deciding the relation Turing reduces to the implication problem. From this it is easy to see that the credulous and skeptical reasoning problem are contained in Σ_2^{p} and Π_2^{p} respectively. The matching lower bounds were later established by Gottlob in [24].

Theorem 4.6 ([49, 24]) *The expansion existence problem and the credulous reasoning problem for autoepistemic logic are Σ_2^{p}-complete, whereas the skeptical reasoning problem for autoepistemic logic is Π_2^{p}-complete.*

The hardness is obtained using a surprisingly simple reduction form the validity problem for quantified Boolean formulae of the form $\exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m \psi$, where ψ is a propositional formula with $\mathrm{Vars}(\psi) = \{x_i \mid 1 \leq i \leq n\} \cup \{y_i \mid 1 \leq i \leq m\}$. Given a formula of the above form, we transform it into a set Σ of autoepistemic formulae defined as

$$\Sigma := \{Lx_i \leftrightarrow x_i \mid 1 \leq i \leq n\} \cup \{L\psi\}.$$

The idea behind the reduction is to mimic existential quantification using different sets of beliefs such that an assignment satisfying $\forall y_1 \cdots \forall y_m \psi$ results in a stable expansion: If σ is an assignment that satisfies $\forall y_1 \cdots \forall y_m \psi$, then the $\Sigma \cup \Lambda$ with $\Lambda = \{L\psi\} \cup \{Lx_i \mid 1 \leq i \leq n\} \cup \{\neg Lx_1 \mid 1 \leq i \leq n\}$ entails ψ; therefore, Λ is a Σ-full set. On the other hand, if Λ is a full set, then we can reconstruct from it an assignment satisfying $\forall y_1 \cdots \forall y_m \psi$.

Beyond this, the complexity of these problems for fragments of autoepistemic logic has seemingly only been studied in [12]. There, it was shown that already autoepistemic logic using only $\wedge$ and $\vee$ is Σ_2^{p}-complete and that tractable fragments occur only for affine sets of Boolean functions. To present the results, say that, for a finite set B of Boolean functions, an *autoepistemic B-formula* is an autoepistemic formula using Boolean functions from a given finite set B only, and denote by $\mathcal{L}_{\mathrm{ae}}(B)$ the set of all autoepistemic B-formulae. Further, let *B-autoepistemic logic* denote autoepistemic logic restricted to autoepistemic B-formulae and define the *credulous (resp. skeptical) reasoning problem for B-autoepistemic logic* as the problem to decide, given a set Σ of autoepistemic B-formulae and an autoepistemic B-formula φ, whether φ is contained in a stable expansion (resp. all stable expansions) of Σ .

Theorem 4.7 ([12]) *Let B be a finite set of Boolean functions. Then the expansion existence problem and the credulous (resp. skeptical) reasoning problem for B-autoepistemic logic are*

1. *Σ_2^{p}-complete (resp. Π_2^{p}-complete) if $\mathsf{D}_2 \subseteq [B]$ or $\mathsf{S}_{00} \subseteq [B]$ or $\mathsf{S}_{10} \subseteq [B]$,*

2. NP-*complete (resp.* coNP-*complete) if* $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$,

3. $\oplus$L-*hard and contained in* P *if* $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$,

4. *in* L *in all other cases (that is, if* $[B] \subseteq \mathsf{E}$*),*

via logspace many-one reductions.

Note that the complexity classification of these problems substantially differs from their analogues in default logic, which can be credited to the different approach to modelling non-monotonicity: while default logic is limited to consistency testing in the justification of a default rule, autoepistemic logic is capable of both positive and negative introspection. As another result, in general the intertranslatability of autoepistemic logic and default logic does not hold for fragments of these logics (unless collapses considered unlikely occur).

We will briefly present the ideas behind Theorem 4.7. To start with, the proof relies on the following lemma, which significantly reduces the number of clones to be considered.

Lemma 4.8 *Let B be a finite set of Boolean functions and $\Sigma \subseteq \mathcal{L}_{\mathrm{ae}}(B \cup \{0,1\})$. Then we can construct in logspace a set $\Sigma' \subseteq \mathcal{L}_{\mathrm{ae}}(B)$ such that the stable expansions of Σ and Σ' coincide on all autoepistemic formulae over* $\mathrm{Vars}(\Sigma)$.

Proof. Let $\Sigma \subseteq \mathcal{L}_{\mathrm{ae}}(B \cup \{0,1\})$ be given. We first eliminate the constant 1 using Lemma 2.1 and transform the resulting set Σ' to Σ'' by substituting all occurrences of 0 with the formula Lf, where f is a fresh proposition. Suppose that Δ is a consistent stable expansion of Σ''. As f cannot be derived from Σ'', Δ has to contain the $\neg Lf$. Hence any satisfying assignment of Δ sets Lf to 0. □

It thus suffices to consider the complexity of the expansion existence problem for sets B such that $[B] \in \{\mathsf{BF}, \mathsf{M}, \mathsf{E}, \mathsf{V}, \mathsf{L}, \mathsf{N}, \mathsf{I}\}$. The key observation in the proof of Theorem 4.7 is that the reductions from the validity problem for quantified Boolean formulae of the form $\exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m \psi$ with ψ in negation normal form does not requires negations: Given a formula φ in the above form, replace all negative literals $\neg x_i$ and $\neg y_i$ in ψ with new propositions x_i' and y_i'. Call the resulting formula φ'. We then construct the set of autoepistemic $\{\wedge, \vee\}$-formulae as

$$\Sigma := \{\mathrm{L}\varphi'\} \cup \{Lx_i \vee x_i', x_i \vee Lx_i' \mid 1 \leq i \leq n\} \cup \{y_i \vee y_i' \mid 1 \leq i \leq m\}.$$

Due to the formulae $Lx_i \vee x_i', x_i \vee Lx_i'$, $1 \leq i \leq n$, any stable expansion Δ of Σ contains either x_i or x_i' (but not both), while the formulae $y_i \vee y_i'$, $1 \leq i \leq m$,

guarantee that either y_i or y_i' is set to 1 in any satisfying assignment of Δ. From this, it is easy to see that φ is valid iff Σ has a stable expansion.Hence, the expansion existence problem is Σ_2^{p}-complete for $\mathsf{S}_{00} \subseteq [B]$, $\mathsf{S}_{10} \subseteq [B]$, or $\mathsf{D}_2 \subseteq [B]$.

This construction can be generalized to also work for quantified Boolean formulae of the form $\exists x_1 \cdots \exists x_n \psi$ with ψ in conjunctive normal form. Hence, we obtain NP-hardness for $\mathsf{V}_2 \subseteq [B]$. The corresponding upper bound follows from the fact the formula implication of B-formulae with $[B] \subseteq \mathsf{V}$ is tractable.

For an argument to obtain a polynomial-time upper bound for the remaining cases, we refer the reader to the original paper. This concludes the discussion of Theorem 4.7.

Turning to the problem of counting the number of stable expansions, the reader may easily convince himself that the reductions used to establish the NP-hardness and Σ_2^{p}-hardness above are parsimonious. Hence, the complexity classification of the counting problem is analogous to that of the expansion existence problem:

Theorem 4.9 ([12]) *Let B be a finite set of Boolean functions. Then the problem to count the number of stable expansions in B-autoepistemic logic is*

1. *$\#\cdot$coNP-complete if $\mathsf{D}_2 \subseteq [B]$ or $\mathsf{S}_{00} \subseteq [B]$ or $\mathsf{S}_{10} \subseteq [B]$,*
2. *$\#$P-complete if $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$,*
3. *in FP in all other cases (that is, if $[B] \subseteq \mathsf{L}$ or $[B] \subseteq \mathsf{E}$),*

via parsimonious reductions.

Note that again parsimonious reductions are sufficient to obtain the completeness results in Theorem 4.9.

5 Circumscription

The third non-monotonic logic we will turn to is circumscription, which instead of extending classical logic with default rules or introspection restricts the attention to minimal models. Circumscription was introduced by McCarthy [42] in 1980 to overcome the *qualification problem*, *i.e.*, the problem of listing all preconditions required for an action to have its intended effect. His approach was to allow for the conclusion that the objects that can be shown to have a certain property by reasoning are all objects that satisfy this property. Following [39], this is achieved by considering only those models that are minimal with respect to a preorder on the set of assignments. For ease of notation, we will identify assignments σ with the set $\{p \mid \sigma(p) = 1\}$.

Definition 5.1 Let P, Q, Z partition the set of propositions and let σ, σ': $P \cup Q \cup Z \to \{0,1\}$ be assignments. Define $\leq_{(P,Z)}$ as the preorder defined by

$$\sigma \leq_{(P,Z)} \sigma' \iff \sigma \cap P \subseteq \sigma' \cap P \text{ and } \sigma \cap Q = \sigma' \cap Q.$$

Using $\leq_{(P,Z)}$, we define a consequence relation $\models_{(P,Z)}$ such that for an assignment σ: $P \cup Q \cup Z \to \{0,1\}$ and a set of formulae Γ with $\mathrm{Vars}(\Gamma) \subseteq P \cup Q \cup Z$, $\sigma \models_{(P,Z)} \Gamma$ if σ is minimal w.r.t. $\leq_{(P,Z)}$ among all models of Γ.. In this case, σ is also called a *circumscriptive model* of Γ. Accordingly, we can define the notion of (circumscriptive) implication: $\Gamma \models_{(P,Z)} \varphi$ if φ is satisfied in all circumscriptive models of Γ. It is not hard to see that circumscription coincides with reasoning under the *extended closed world assumption*, in which all formulae involving only propositions from P that cannot be derived from Γ are assumed to be false [27].

Example 5.2 Let $P := \{x\}$, $Q := \emptyset$, $Z := \{y, z\}$ and $\Gamma := \{(x \wedge \neg y) \to z\}$. The models of Γ are $\emptyset$, $\{y\}$, $\{z\}$, $\{x, y\}$, $\{y, z\}$, $\{x, z\}$, $\{y\}$, and $\{x, y, z\}$. Of these, only $\emptyset$, $\{z\}$, and $\{y, z\}$ are minimal with respect to $\leq_{(P,Z)}$. Hence, $\Gamma \cup \{y\} \models_{(P,Z)} z$, while $\Gamma \cup \{y\} \not\models z$.

The notions of circumscriptive models and circumscriptive inference naturally lead to the following decision problems, that received extensive study in the literature:

(*Circumscriptive model checking*) Given a set of formulae Γ, a preorder $\leq_{(P,Z)}$ on the set of its propositions and an assignment σ, does $\sigma \models_{(P,Z)} \Gamma$ hold?

(*Circumscriptive inference*) Given a set of formulae Γ and a formula φ, a preorder $\leq_{(P,Z)}$ on the set of their propositions, does $\Gamma \models_{(P,Z)} \varphi$ hold?

The circumscriptive model checking is dual to a generalization of the minimal satisfiability problem, *i.e.*, the question whether a given formula has a model that is strictly smaller than a given assignment with respect to a given preorder $\leq_{(P,Z)}$, and known to be coNP-complete in general [9], whereas the circumscriptive inference problem was shown to be Π_2^{p}-complete by Eiter and Gottlob in [20]. These results reveal that, alike default logic and autoepistemic logic, circumscription exhibits an increase in the complexity of model checking and reasoning as compared to traditional propositional logic. This increase in the complexity raises the question for restrictions that lower the complexity of these tasks. Accordingly, the complexity of these problems has been studied for both restricted sets of Boolean functions and in Schaefers framework. We will consider the restrictions obtained from *Schaefer's framework* first.

Define the *circumscriptive model checking problem for sets of relations from* S as the problem to decide, given a a set Γ of applications of relations from S, an assignment $\sigma\colon \mathrm{Vars}(\Gamma) \to \{0,1\}$ and a partition (P,Q,Z) of $\mathrm{Vars}(\Gamma)$, whether σ is a minimal model of Γ with respect to $\leq_{(P,Z)}$. In [34], Kirousis and Kolaitis showed that using Schaefer's framework, the circumscriptive model checking problem restricted to $Q = Z = \emptyset$ is dichotomic, a result which was later generalized to the general case in [36]:

Theorem 5.3 ([36]) *Let S be a set of relations. Then the circumscriptive model checking problem for sets of relations from S is*

1. coNP*-complete if S is not Schaefer and*
2. *in* P *in all other cases.*

The tractability if S is Schaefer is easy to verify. In this case, the circumscriptive model checking problem Turing reduces to the satisfiability problem, which in this case is tractable by [57]. To show the coNP-hardness in all remaining cases, Kirousis and Kolaitis give an involved three step reduction from 1-in-3 SAT.

In addition to that, Kirousis and Kolaitis also classified for possible sets of available Boolean functions in an unpublished note. Define the *circumscriptive model checking problem for sets of B-formulae* as the problem to decide, given a set $\Gamma \subseteq \mathcal{L}(B)$, an assignment $\sigma\colon \mathrm{Vars}(\Gamma) \to \{0,1\}$ and a partition (P,Q,Z) of $\mathrm{Vars}(\Gamma)$, whether σ is a minimal model of Γ with respect to $\leq_{(P,Z)}$.

Theorem 5.4 ([33]) *Let B be a finite set of Boolean functions. Then the circumscriptive model checking problem for sets of B-formulae is*

1. coNP*-complete if* $\mathsf{S}_{02} \subseteq [B]$ *or* $\mathsf{S}_{12} \subseteq [B]$ *or* $\mathsf{D}_1 \subseteq [B]$*, and*
2. *in* P *in all other cases (that is, if* $[B] \subseteq \mathsf{M}$*).*

Key to the classification is that if the set B of all available Boolean functions is monotone, then the circumscriptive model checking problem Turing-reduces to the the model checking problem for monotone Boolean formulae. Given Γ and σ, denote by σ^i the assignment obtained by setting all propositions in Z to 1 and the ith proposition in P, which is set to 1 under σ to 0. Then $\sigma \models_{(P,Z)} \Gamma$ iff for all σ^i obtained in this way, $\sigma^i \not\models_{(P,Z)} \Gamma$. Thus circumscriptive model checking problem for sets of B-formulae is tractable if $[B] \subseteq \mathsf{M}$.

On the other hand, the coNP-hardness follows from the fact that all remaining sets B satisfy $[B \cup \{0,1\}] = \mathsf{BF}$, while we can simulate the Boolean

constants: If $B \subseteq \mathsf{R}_1$, then appealing to Lemma 2.1 suffices. If $B \subseteq \mathsf{R}_1$, we use the mapping $\Gamma \mapsto \Gamma' := \Gamma[0/f, 1/t] \cup \{t, f \rightarrow \bigwedge \mathrm{Vars}(\Gamma)\}$, where $\Gamma[0/f, 1/t]$ denotes the set obtained from Γ by replacing all occurrences of 0 by the fresh proposition f and all occurrences of 1 by the fresh proposition t. Notice that any model of Γ' sets t to 1 and that Γ' is satisfied by exactly those assignments that satisfy Γ and additionally the assignment setting all propositions to 1. Thus, if $B \subseteq \mathsf{R}_1$, we may replace any Boolean function in the original Γ with an equivalent $(B \cup \{0, 1\})$-formula.

We point out that the original proof in [33] builds on results from [34], the sketch above provides an alternative proof.

As for the inference problem, the computational complexity of circumscriptive inference was first studied in 1990 by Cadoli and Lenzerini [11], who analyzed the complexity of reasoning under various closed world assumptions. They showed that circumscription for various restrictions on the premises, conclusions and the order still remains intractable and also identified tractable fragments. Yet the exact complexity of circumscriptive inference remained open until Eiter and Gottlob [20] proved its Π_2^{p}-completeness. This result was further refined both in the framework of restricted sets of available Boolean functions as well as in Schaefer's framework. For the latter, Kirousis and Kolaitis [35] proved a dichotomy separating the Π_2^{p}-complete cases and from those in coNP, and conjectured that the cases in coNP could be refined into coNP-complete and tractable ones. While the coNP-completeness for dual Horn relations or bijunctive relations was known from [11] and Durand and Hermann showed that this also holds for for affine relations [16], Nordh finally affirmatively settled Kirousis and Kolaitis' conjecture in [50]. For the restricted case of *basic circumscription* that requires Q or Z or both to be empty, the trichotomy was established in [19]. We state here the result from [50].

Say that a relation R is *negative Horn* if it coincides with the set of models of a Horn formula without positive literals. Define the *circumscriptive inference problem for sets of relations from* S as the problem to decide, given a set Γ of applications of relations from S, a clause φ and a partition (P, Q, Z) of $\mathrm{Vars}(\Gamma \cup \{\varphi\})$, whether $\Gamma \models_{(P,Z)} \varphi$.

Theorem 5.5 ([50]) *Let S be a set of relations. Then the circumscriptive inference problem for sets of relations from S is*

1. Π_2^{p}*-complete if S is not Schaefer,*
2. coNP*-complete if S is Schaefer but neither negative Horn nor both bijunctive and affine nor both Horn and dual Horn, and*

3. *in* P *in all other case (that is, if S is negative Horn or both bijunctive and affine or both Horn and dual Horn).*

For the approach parameterizing by the set available Boolean functions, the circumscriptive inference problem for sets of B-formulae was classified in [64]. Define the *circumscriptive inference problem for sets of B-formulae* as the problem to decide, given a set $\Gamma \subseteq \mathcal{L}(B)$, a clause φ and a partition (P, Q, Z) of $\mathrm{Vars}(\Gamma \cup \{\varphi\})$, whether $\Gamma \models_{(P,Z)} \varphi$.

Theorem 5.6 ([64]) *Let B be a finite set of Boolean functions. Then the circumscriptive inference problem for sets of B-formulae is*

1. Π_2^{p}*-complete if* $\mathsf{S}_{02} \subseteq [B]$ *or* $\mathsf{S}_{12} \subseteq [B]$ *or* $\mathsf{D}_1 \subseteq [B]$,
2. coNP*-complete if* $X \subseteq [B] \subseteq Y$ *for some* $X \in \{\mathsf{V}_2, \mathsf{S}_{10}, \mathsf{D}_2, \mathsf{L}_2\}$ *and* $Y \in \{\mathsf{M}, \mathsf{L}\}$, *and*
3. *in* P *in all other case (that is, if* $[B] \subseteq \mathsf{N}$ *or* $[B] \subseteq \mathsf{E}$*).*

Remark 5.7 Unlike the reasoning problems defined for B-default logic and B-autoepistemic logic, here an arbitrary formula is to be tested for implication. The complexity of deciding the circumscriptive inference of a B-formula from a set of B-formulae is the same as above for all sets B except those satisfying $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$; for these the problem is only known to be $\oplus$L-hard and contained in coNP.

In the remainder of this section, we will consider the counting complexity of circumscription, namely the problem to count the number of minimal models of a given set Γ w.r.t. to a given preorder $\leq_{(P,Z)}$. This problem, henceforth referred to as the *circumscriptive model counting problem*, has recently gained a lot of interest. Its restriction to $Q = Z = \emptyset$ is equivalent to the *minimal model counting problem*, *i.e.*, the problem of counting the number of minimal models w.r.t. the coordinatewise partial order on assignments induced by $0 < 1$. While the problem to count the number of all models of a given formula is well-known to be #P-complete via parsimonious reduction [68], the exact complexity of the minimal model counting problem was open for a long time. The problem is easily seen to belong to #·coNP, since deciding whether a given assignment is among the minimal models of a given set of formulae is in coNP. Similarly, its #P-hardness via parsimonious reductions is apparent: the mapping $\varphi \mapsto \{\varphi \wedge \bigwedge_{x_i \in \mathrm{Vars}(\varphi)} (x_i \oplus y_i)\}$ with $y_i \notin \mathrm{Vars}(\varphi)$ constitutes a parsimonious reduction from the problem to count the number of all models of a given formula. However, #·coNP-hardness could only be established using reductions under which not all of the classes of the counting polynomial hierarchy are closed.

In 2000, Durand, Hermann and Kolaitis [18] introduced the notion of *subtractive reductions* which suitably relaxed the notion of parsimonious reductions. Say that a counting problem $\#A$ reduces to a counting problem $\#B$ via strong subtractive reduction if there exists a pair of polynomial-time computable functions f, g such that for all x, $B(g(x)) \subseteq B(f(x))$ and $|A(x)| = |B(f(x))| - |B(g(x))|$. Subtractive reductions are the transitive closure of strong subtractive reductions, *i.e.*, $\#A$ reduces to $\#B$ via subtractive reductions if there exists a finite sequence $(\#C)_{1 \leq i \leq n}$ such that $\#C_1 = \#A$, $\#C_n = \#B$ and $\#C_i$ reduces to $\#C_{i+1}$ via strong subtractive reduction for all $1 \leq i < n$. Clearly, each parsimonious reduction is also a subtractive reduction. And, more importantly, #P and $\#\cdot\Pi_k^{\mathrm{p}}$, for all $k \geq 1$, are closed under subtractive reductions.

Theorem 5.8 ([18]) *The minimal model counting problem and the circumscriptive model counting problem are* $\#\cdot$coNP*-complete via subtractive reductions.*

The counting complexity of the minimal model counting problem was further studied in [17]. There the authors show that, using Schaefer's framework, the restriction to relations that are either dual Horn, bijunctive or affine reduces the complexity of the problem to #P-completeness, whereas the restriction to Horn relations or relations that are both bijunctive an affine yields efficiently computable counting problems. Hence, the counting problem for the case that all relations are both dual Horn and Horn as well as the case that all relations are negative Horn are #P-complete while the underlying decision problems are still tractable.

The counting complexity of both the minimal model counting problem and the circumscriptive model counting problem was further studied in [66], where the complexity of the fragments obtained by restricting the set of Boolean functions is classified. Here, the counting complexity decreases analogously to the complexity of the underlying decision problem.

Theorem 5.9 ([66]) *Let B be a finite set of Boolean functions. Then the minimal model counting problem for B-formulae and the circumscriptive model counting problem for B-formulae is*

1. $\#\cdot$coNP*-complete via subtractive reductions if* $\mathsf{S}_{02} \subseteq [B]$ *or* $\mathsf{S}_{12} \subseteq [B]$ *or* $\mathsf{D}_1 \subseteq [B]$*,*
2. #P*-complete via subtractive reductions if* $\mathsf{S}_{00} \subseteq [B] \subseteq \mathsf{M}$ *or* $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{M}$ *or* $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{M}$*,*
3. #P*-complete via Turing reductions if* $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$ *or* $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$*, and*
4. *in* FP *in all other cases (that is, if* $[B] \subseteq \mathsf{N}$ *or* $[B] \subseteq \mathsf{E}$*).*

6 Abduction

Abduction is a fundamental and important form of non-monotonic reasoning introduced by Peirce [53]. It can be thought of as a form of hypothetical reasoning: to ask what can be abduced from an observation α is to ask for an explanation, which in conjunction with the given background knowledge accounts for α. The importance of this formalism to artificial intelligence was first emphasized by Morgan [46] and has been fruitfully used in many areas of computer science such as medical diagnosis [2], text analysis [30], system diagnosis [63], configuration problems [1], temporal knowledge bases [5], and has connections to default reasoning [60].

Here we will consider logic based abduction in which the background theory is represented by a logical theory, specifically in propositional logic. Hence, the abduction problem can in general be formulated as the problem, given a *knowledge base* $\Gamma \subseteq \mathcal{L}$, a set $A \subseteq \mathrm{Vars}(\Gamma)$ of propositions called *hypothesis*, and an *observation* $q \in \mathrm{Vars}(\Gamma)$, to compute a set $E \subseteq \mathrm{Lits}(A)$ of literals over A such that $\Gamma \cup E$ is consistent and $\Gamma \cup E \models q$. If such a set E exists, then it is called an *explanation* for the *abduction problem* $\mathcal{P} = (\Gamma, A, q)$.

The computational complexity of this problem has first been considered by Selman and Levesque [60], who showed that it is NP-hard to compute an explanation if Γ is restricted to Horn clauses. Independently, Friedrich, Gottlob and Nejdl [23] studied the problem for definite Horn clauses. They show that deciding whether a given proposition is contained in some or all explanations is tractable whereas deciding whether a proposition is contained in a subset-minimal explanation is NP-complete. It is yet tractable to compute some subset-minimal explanation in this case [8]. Further results were obtained by Eshghi [22], who proved that finding subset-minimal explanations becomes tractable if Γ is acyclic Horn and its pseudo-completion is unit-refutable. Finally, the complexity of logic-based abduction was settled by Eiter and Gottlob.

Theorem 6.1 ([21])

1. *To decide, given an abduction problem $\mathcal{P} = (\Gamma, A, q)$ and a set $E \subseteq \mathrm{Lits}(A)$, whether E is an explanation for $\mathcal{P}$ is D^{P}-complete.*
2. *To decide, given an abduction problem $\mathcal{P}$, whether there exists an explanation for $\mathcal{P}$ is Σ_2^{p}-complete.*
3. *To decide, given an abduction problem $\mathcal{P}$ and a proposition p, whether all explanations for $\mathcal{P}$ contain p is Π_2^{p}-complete.*

These results also hold for subset-minimal explanations.

Eiter and Gottlob [21] also studied the complexity of abduction for preference relations other than subset-minimality, but to include these results here would go beyond the scope of this survey. As in most cases the complexity of the first and the third problem in Theorem 6.1 can be derived from the question whether an explanation exists, we will henceforth focus on the complexity of deciding the existence of an explanation.

Due to its applications to knowledge-based systems, it is natural to consider the complexity of this problem in Schaefer's framework. Using this approach Creignou and Zanuttini [15] showed that the complexity of deciding the existence of an explanation forms a trichotomy. Define the *explanation existence problem for sets of relations from* S as the problem to decide, given a set Γ of applications of relations from S, a set $A \subseteq \mathrm{Vars}(\Gamma)$ of propositions, and a proposition $q \in \mathrm{Vars}(\Gamma) \setminus A$, whether there exists a set $E \subseteq \mathrm{Lits}(A)$ such that $\Gamma \cup E$ is consistent and $\Gamma \cup E \models q$. Say that a relation R is *IHS-B−* if if it coincides with the models of a CNF formula whose clauses are all of one of the following types: (x_i), $(\neg x_{i_1} \vee x_{i_2})$, $(\neg x_{i_1} \vee \cdots \vee \neg x_{i_k})$ for some $k > 0$. Analogously say that a relation R is *IHS-B+* if if it coincides with the models of a CNF formula whose clauses are all of one of the following types: $(\neg x_i)$, $(\neg x_{i_1} \vee x_{i_2})$, $(x_{i_1} \vee \cdots \vee x_{i_k})$ for some $k > 0$. Clearly, any IHS-B− formula (resp. IHS-B+) formula is Horn (resp. dual Horn).

Theorem 6.2 ([15]) *Let S be a set of relations. Then the explanation existence problem for sets of relations from S is*

1. Σ_2^{p}*-complete if S is not Schaefer,*
2. NP*-complete if S is either Horn or dual Horn but neither bijunctive nor affine nor definite Horn nor IHS-B+ nor IHS-B−,*
3. *in* P *if S is bijunctive or affine or definite Horn or IHS-B+ or IHS-B−.*

As for restricted sets Γ the restriction of the query q to a positive proposition does no longer come without loss of generality, variants of this problem have been studied, where the observation is a term (a conjunction of literals), a clause or an arbitrary formula. For example, if one allows for the query to be a literal instead of a proposition, then the explanation existence problem for definite Horn relations becomes NP-complete (apart from that, the classification of Theorem 6.2 remains valid). For the case that the observation is a term, a trichotomy has been established in [51]: here only the case that all relations are affine remains tractable, while Horn relations, dual Horn relations and relations expressible using either only $(\neg x_{i_1} \vee x_{i_2})$, or only $(x_{i_1} \leftrightarrow x_{i_2})$ and $(x_{i_1} \vee x_{i_2})$, or only $(x_{i_1} \leftrightarrow x_{i_2})$ and $(\neg x_{i_1} \vee \neg x_{i_2})$ lead to NP-complete fragments. Further loosening the restriction on the observations to allow for

an arbitrary formula leads to a dichotomic classification into NP-complete fragments (if the set of relations is Schaefer) and Σ_2^{p}-complete fragments (in all other cases). These results are summarized and extended to also cover clauses and arbitrary formulae and to several restrictions on the hypothesis in [52].

Seeking further insights into the sources of complexity, abduction has also been studied for restricted sets of Boolean functions. In [13], Creignou, Schmidt and Thomas completely classified the complexity of the *explanation existence problem for B-formulae*, defined as the problem to decide, given a given a set $\Gamma \subseteq \mathcal{L}(B)$ a set $A \subseteq \mathrm{Vars}(\Gamma)$ of propositions, and an observation $q \in \mathrm{Vars}(\Gamma) \setminus A$, whether there exists a set $E \subseteq \mathrm{Lits}(A)$ such that $\Gamma \cup E$ is consistent and $\Gamma \cup E \models q$.

Theorem 6.3 ([13]) *Let B be a finite set of Boolean functions. Then the explanation existence problem for B-formulae is*

1. Σ_2^{p}*-complete if* $\mathsf{S}_{02} \subseteq [B]$ *or* $\mathsf{S}_{12} \subseteq [B]$ *or* $\mathsf{D}_1 \subseteq [B]$,
2. NP*-complete if* $\mathsf{S}_{00} \subseteq [B] \subseteq \mathsf{M}$ *or* $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{M}$ *or* $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{M}$, *and*
3. *in* P *in all other cases (that is, if* $[B] \subseteq \mathsf{E}$ *or* $[B] \subseteq \mathsf{V}$ *or* $[B] \subseteq \mathsf{L}$*).*

The complexity of this problem for observations represented by clauses, terms and B-formula was also classified. The relaxation to clauses does not alter the complexity classification; however, the relaxation to observations represented by terms increases the complexity of the cases satisfying $[B \cup \{0,1\}] = \mathsf{V}$ to NP-completeness. Lastly, if the observation is formalized as a B-formula then the classification becomes dichotomic with the clones above E, V or L being Σ_2^{p}-complete and all remaining clones being tractable; thus skipping the intermediate NP level.

The complexity of propositional abduction has thus been systematically studied and is well understood. The counting complexity of abduction was first studied by Hermann and Pichler [28, 29]. The problems arising in this context are the problem to count the number of all explanations as well as the problem to count the number of minimal explanations with respect to a given preference relation, *e.g.*, the subset-minimal explanations or explanations of minimal cardinality. From the set of counting problems for abduction studied by Hermann and Pichler, we will consider the following two problems:

(*Explanation counting*) Given a set $\Gamma \subseteq \mathcal{L}$, a set $A \subseteq \mathrm{Vars}(\Gamma)$, and a conjunction $q_1 \wedge \cdots \wedge q_n$ or propositions from $\mathrm{Vars}(\Gamma) \setminus A$, to count the number of all sets $E \subseteq A$ such that $\Gamma \cup E$ is consistent and $\Gamma \cup E \models q_1 \wedge \cdots \wedge q_n$.

(*Subset-minimal explanation counting*) Given a set $\Gamma \subseteq \mathcal{L}$, a set $A \subseteq \mathrm{Vars}(\Gamma)$, and a conjunction $q_1 \wedge \cdots \wedge q_n$ or propositions from $\mathrm{Vars}(\Gamma) \setminus A$,

to count the number of all sets $E \subseteq A$ such that $\Gamma \cup E$ is consistent and $\Gamma \cup E \models q_1 \wedge \cdots \wedge q_n$.

Theorem 6.4 ([28]) *The explanation counting problem and the subset-minimal explanation counting problem are* $\#\cdot$coNP*-complete via subtractive reductions.*

However, if one restricts Γ be be a set of applications of relations from a fixed set S of available relations, then the counting complexity drops by at least one level of the counting polynomial hierarchy: The explanation counting problem for sets of relations from S is #P-complete if S is Horn or dual Horn or bijunctive; it is contained in FP if S is affine and the explanations are allowed to contain literals instead of propositions. On the other hand, the subset-minimal explanation counting problem for sets of relations from S is #P-complete in all of the previously mentioned cases.

In addition to these results, the complexity of counting all explanations in the case that Γ is represented by a set of B-formulae and the observation is a single proposition was also studied in [13]. There both variants, to count the number of (positive) explanations and to count the number of literal explanations, have been studied.

Theorem 6.5 ([13]) *Let B be a finite set of Boolean functions. Then the explanation counting problem for sets of B-formulae is*

1. $\#\cdot$coNP*-complete if* $\mathsf{S}_{02} \subseteq [B]$ *or* $\mathsf{S}_{12} \subseteq [B]$ *or* $\mathsf{D}_1 \subseteq [B]$,
2. #P*-complete if* $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{M}$ *or* $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{M}$ *or* $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{M}$, *and*
3. *in* FP *in all other cases (that is, if* $[B] \subseteq \mathsf{E}$ *or* $[B] \subseteq \mathsf{L}$*),*

via subtractive reductions. If explanations are restricted to contain positive literals only, then the problem is contained in FP *for* $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$ *and only known to be in* #P *for* $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$.

The open case in Theorem 6.5 is equivalent to the case where Γ is restricted to a set of affine relations. This case was already left open in [28].

References

[1] J. Amilhastre, H. Fargier, and P. Marquis. Consistency restoration and explanations in dynamic CSPs. *Artif. Intell.*, 135(1-2):199–234, 2002.

[2] T. Bylander, D. Allemang, M. C. Tanner, and J. R. Josephson. Some results concerning the computational complexity of abduction. In *Proc. 1st KR*, pages 44–54, 1989.

[3] M. Bauland, E. Böhler, N. Creignou, S. Reith, H. Schnoor, and H. Vollmer. The complexity of problems for quantified constraints. *Theory Comput. Syst.*, 47(2):454–490, 2010.

[4] R. Ben-Eliyahu-Zohary. Yet some more complexity results for default logic. *Artif. Intell.*, 139(1):1–20, 2002.

[5] M. Bouzid and A. Ligeza. Temporal causal abduction. *Constraints*, 5(3):303–319, 2000.

[6] O. Beyersdorff, A. Meier, M. Thomas, and H. Vollmer. The complexity of propositional implication. *Inf. Process. Lett.*, 109(18):1071–1077, 2009.

[7] O. Beyersdorff, A. Meier, M. Thomas, and H. Vollmer. The complexity of reasoning for fragments of default logic. In *Proc. 12th SAT*, volume 5584 of *LNCS*, pages 51–64, Berlin Heidelberg New York, 2009. Springer.

[8] T. Bylander. The monotonic abduction problem: A functional characterization on the edge of tractability. In *Proc. 2nd KR*, pages 70–77, 1991.

[9] M. Cadoli. The complexity of model checking for circumscriptive formulae. *Inf. Process. Lett.*, 44:113–118, 1992.

[10] P. Chapdelaine, M. Hermann, and I. Schnoor. Complexity of default logic on generalized conjunctive queries. In *Proc. 9th LPNMR*, volume 4483 of *Lecture Notes in Computer Science*, pages 58–70. Springer, 2007.

[11] M. Cadoli and M. Lenzerini. The complexity of propositional closed world reasoning and circumscription. *J. Comput. Syst. Sci.*, 48(2):255–310, 1994.

[12] N. Creignou, A. Meier, M. Thomas, and H. Vollmer. The complexity of reasoning for fragments of autoepistemic logic. In *Circuits, Logic, and Games*, volume 10061 of *Dagstuhl Seminar Proceedings*, 2010.

[13] N. Creignou, J. Schmidt, and M. Thomas. Complexity of propositional abduction for restricted sets of Boolean functions. In *Proc. 12th KR*. AAAI, 2010.

[14] Nadia Creignou and Heribert Vollmer. Boolean constraint satisfaction problems: When does post's lattice help? In N. Creignou, Ph. G. Kolaitis, and H. Vollmer, editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 3–37, Berlin Heidelberg New York, 2008. Springer.

[15] N. Creignou and B. Zanuttini. A complete classification of the complexity of propositional abduction. *SIAM J. Comput.*, 36(1):207–229, 2006.

[16] A. Durand and M. Hermann. The inference problem for propositional circumscription of affine formulas is coNP-complete. In *Proc. 20th STACS*, volume 2607 of *LNCS*, pages 451–462. Springer, 2003.

[17] A. Durand and M. Hermann. On the counting complexity of propositional circumscription. *Inf. Process. Lett.*, 106(4):164–170, 2008.

[18] A. Durand, M. Hermann, and P. G. Kolaitis. Subtractive reductions and complete problems for counting complexity classes. *Theoretical Computer Science*, 340(3):496–513, 2005.

[19] A. Durand, M. Hermann, and G. Nordh. Trichotomy in the complexity of minimal inference. In *Proc. 24th LICS*, pages 387–396. IEEE Computer Society, 2009.

[20] T. Eiter and G. Gottlob. Propositional circumscription and extended closed world reasoning are Π_2^p-complete. *Theoretical Computer Science*, 114(2):231–245, 1993. Addendum in TCS 118 (1993), page 15.

[21] T. Eiter and G. Gottlob. The complexity of logic-based abduction. *J. ACM*, 42(1):3–42, 1995.

[22] K. Eshghi. A tractable class of abduction problems. In *Proc. 13th IJCAI*, pages 3–8. Morgan Kaufmann, 1993.

[23] G. Friedrich, G. Gottlob, and W. Nejdl. Hypothesis classification, abductive diagnosis and therapy. In *Expert Systems in Engineering*, volume 462 of *LNCS*, pages 69–78. Springer, 1990.

[24] G. Gottlob. Complexity results for nonmonotonic logics. *J. Log. Comput.*, 2(3):397–425, 1992.

[25] G. Gottlob. NP trees and Carnap's modal logic. *J. ACM*, 42:421–457, 1995.

[26] G. Gottlob. Translating default logic into standard autoepistemic logic. *J. ACM*, 42(4):711–740, 1995.

[27] M. Gelfond, H. Przymusinska, and T. C. Przymusinski. On the relationship between circumscription and negation as failure. *Artificial Intelligence*, 38(1):75–94, 1989.

[28] M. Hermann and R. Pichler. Counting complexity of propositional abduction. In *Proc. 20th IJCAI*, volume 4451 of *LNCS*, pages 417–422. Springer, 2007.

[29] M. Hermann and R. Pichler. Counting complexity of minimal cardinality and minimal weight abduction. In *Proc. 11th JELIA*, volume 5293 of *LNCS*, pages 206–218. Springer, 2008.

[30] J. R. Hobbs, M. E. Stickel, D. E. Appelt, and P. A. Martin. Interpretation as abduction. *Artif. Intell.*, 63(1-2):69–142, 1993.

[31] L. Hemaspaandra and H. Vollmer. The satanic notations: counting classes beyond #P and other definitional adventures. *Complexity Theory Column 8, ACM-SIGACT News*, 26(1):2–13, 1995.

[32] T. Janhunen. On the intertranslatability of non-monotonic logics. *Ann. Math. Artif. Intell.*, 27(1-4):79–128, 1999.

[33] L. M. Kirousis and P. Kolaitis. The complexity of minimal satisfiability in Post's lattice. Unpublished notes, 2001.

[34] L. M. Kirousis and P. Kolaitis. The complexity of minimal satisfiability problems. In *Proc. 18th STACS*, volume 2010 of *LNCS*, pages 407–418. Springer, 2001.

[35] L. M. Kirousis and P. G. Kolaitis. A dichotomy in the complexity of propositional circumscription. In *Proc. 16th LICS*, pages 71–80, 2001.

[36] L. M. Kirousis and P. Kolaitis. The complexity of minimal satisfiability problems. *Information and Computation*, 187(1):20–39, 2003.

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

[38] H. A. Kautz and B. Selman. Hard problems for simple default logics. *Artif. Intell.*, 49:243–279, 1991.

[39] V. Lifschitz. Computing circumscription. In *Proc. 9th IJCAI*, pages 121–127. Morgan Kaufman, 1985.

[40] F. Laroussinie, N. Markey, and P. Schnoebelen. Model checking CTL^+ and FCTL is hard. In *Proc. 4th FoSSaCS*, volume 2030 of *LNCS*, pages 318–331. Springer, 2001.

[41] P. Liberatore and M. Schaerf. The complexity of model checking for propositional default logics. *Data Knowl. Eng.*, 55(2):189–202, 2005.

[42] J. McCarthy. Circumscription – A form of non-monotonic reasoning. *Artif. Intell.*, 13:27–39, 1980.

[43] D. McDermott. Non-monotonic logic II: nonmonotonic modal theories. *J. ACM*, 29:33–57, 1982.

[44] D. McDermott and J. Doyle. Non-monotonic logic I. *Artif. Intell.*, 13:41–72, 1980.

[45] R. C. Moore. Semantical considerations on modal logic. *Artificial Intelligence*, 25:75–94, 1985.

[46] C. G. Morgan. Hypothesis generation by machine. *Artif. Intell.*, 2(2):179–187, 1971.

[47] V. W. Marek and M. Truszczyński. Relating autoepistemic and default logics. In *Proc. 1st KR*, pages 276–288. Morgan Kaufmann, 1989.

[48] V. W. Marek and M. Truszczyński. Modal logic for default reasoning. *Ann. Math. Artif. Intell.*, 1:175–302, 1990.

[49] I. Niemelä. Towards automatic autoepistemic reasoning. In *Proc. 2th JELIA*, volume 478 of *LNCS*, pages 428–443. Springer, 1990.

[50] G. Nordh. A trichotomy in the complexity of propositional circumscription. In *Proc. 11th LPAR*, volume 3452 of *LNCS*, pages 257–269. Springer, 2005.

[51] G. Nordh and B. Zanuttini. Propositional abduction is almost always hard. In *Proc. 19th IJCAI*, pages 534–539. Professional Book Center, 2005.

[52] G. Nordh and B. Zanuttini. What makes propositional abduction tractable. *Artif. Intell.*, 172(10):1245–1284, 2008.

[53] C. S. Peirce. Abduction and induction. In *Philosophical writings of Peirce*, chapter 11, pages 150–156. Dover, New York, 1955.

[54] N. Pippenger. On simultaneous resource bounds. In *Proc. 20th FOCS*, pages 307–311. IEEE Computer Society Press, 1979.

[55] E. Post. The two-valued iterative systems of mathematical logic. *Ann. Math. Stud.*, 5:1–122, 1941.

[56] R. Reiter. A logic for default reasoning. *Artif. Intell.*, 13:81–132, 1980.

[57] T. J. Schaefer. The complexity of satisfiability problems. In *Proc. 10th STOC*, pages 216–226. ACM Press, 1978.

[58] I. Schnoor. *The Weak Base Mthod for Constraint Satisfaction.* PhD thesis, Leibniz Universität Hannover, 2007.

[59] Grigori Shvarts. Autoepistemic modal logics. In *Proc. 3rd TARK*, pages 97–109. Morgan Kaufmann, 1990.

[60] B. Selman and H. Levesque. Abductive and default reasoning: A computational core. In *Proc. 8th AAAI*, pages 343–348. AAAI Press, 1990.

[61] J. Stillman. It's not my default: The complexity of membership problems in restricted propositional default logics. In *Proc. 8th AAAI*, pages 571–578, 1990.

[62] J. Stillman. The complexity of propositional default logics. In *Proc. 10th AAAI*, pages 794–800, 1992.

[63] M. Stumptner and F. Wotawa. Diagnosing tree-structured systems. *Artif. Intell.*, 127(1):1–29, 2001.

[64] M. Thomas. The complexity of circumscriptive inference in Post's lattice. In *Proc. 10th LPNMR*, volume 5753 of *LNCS*, pages 290–302. Springer, 2009.

[65] M. Thomas. On the applicability of Post's lattice. Technical report, Leibniz Universität Hannover, 2010. Available at arXiv:1007.2924 [cs.CC].

[66] M. Thomas. *On the complexity of fragments of nonmonotonic logics.* PhD thesis, Leibniz Universität Hannover, 2010.

[67] S. Toda. *Computational Complexity of Counting Complexity Classes.* PhD thesis, Tokyo Institute of Technology, Department of Computer Science, Tokyo, 1991.

[68] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.