Walking the Tightrope between Expressiveness and Uncomputability: AGM Contraction beyond the Finitary Realm

Dominik Klumpp[1,*], Jandson S Ribeiro[2]

[1]University of Freiburg, Germany
[2]Cardiff University, United Kingdom

Abstract
Although there has been significant interest in extending the AGM paradigm of belief change beyond finitary logics, the computational aspects of AGM have remained almost untouched. We investigate the computability of AGM contraction on non-finitary logics, and show an intriguing negative result: there are infinitely many uncomputable AGM contraction functions in such logics. Drastically, even if we restrict the theories used to represent epistemic states, in all non-trivial cases, the uncomputability remains. On the positive side, we use Büchi automata to construct computable AGM contraction functions on Linear Temporal Logic (LTL).

1. Introduction

The field of *Belief Change* [1, 2, 3] investigates how to keep a corpus of beliefs consistent as it evolves. The field is mainly founded on the AGM paradigm [1], named after its authors' initials, which distinguishes, among others, two main kinds of changes: *belief revision*, which consists in incorporating an incoming piece of information with the proviso that the updated corpus of beliefs is consistent; and *belief contraction* whose purpose is to retract an obsolete piece of information. In either case, the incurred changes should be minimized so that most of the original beliefs are preserved. This is known as *principle of minimal change*. Contraction is central as it can be used to define other forms of belief change. For example, belief revision can be defined in terms of contraction: first, remove information in conflict with the incoming belief via contraction, only then incorporate the incoming belief. When classical negation is at disposal, this recipe for defining revision from contraction is formalised via the Levi identity [4, 5].

The AGM paradigm prescribes rationality postulates that capture the *principle of minimal change*, and constructive functions that satisfy such postulates, called rational functions, as for instance *partial meet* [1], (smooth) *kernel contraction* [6], epistemic entrenchment [2] and Grove's system of spheres [7]. Originally, the AGM paradigm was developed assuming some conditions about the underlying logic [2, 8]. Although these conditions cover some classical logics such as classical Propositional Logic and First Order Logic, they restrict the reach of the AGM paradigm into more expressive logics including several Descriptions Logics [9], Modal Logics [10] and Temporal Logics such as LTL, CTL and CTL* [11]. It turns out that the AGM paradigm is independent of such conditions [8], although rational contraction functions do not exist in every logic [8, 12]. Logics in which rational contraction functions do exist are dubbed *AGM compliant* [8]. As a result, several works have been dedicated to dispense with the AGM assumptions in order to extend the paradigm to more expressive logics: Horn logics [13, 14, 15], para-consistent logics [16], Description Logics [17, 12, 8], and non-compact logics [18]. See [19] for a discussion of several other works in this line.

Although much effort has been put into extending the AGM paradigm to more expressive logics, few works have investigated the computational aspects of the AGM paradigm such as [20, 21, 22]. All these works, however, focused on investigating the complexity of decision problems for some fixed belief change operators on classical propositional logics and Horn. In light of the interest and effort of expanding the AGM paradigm for more expressive non-classical logics, it is paramount to comprehend the computational aspects of belief change in such more expressive logics. In this context, there is a central question that even precedes complexity:

Computability / Effectiveness: Given a belief change operator $\circ$, does there exist a Turing Machine that computes $\circ$, and stops on all inputs?

The answer to the question above depends on two main elements: the underlying logic, and the chosen operator. Clearly, it only makes sense to answer such questions for logics that are AGM compliant. However, independently of the operator, the question is trivial for finitary logics, that is, logics whose language contains only finitely many equivalence classes, as it is the case of classical propositional logics. For non-finitary logics, by contrast, we show a disruptive result: AGM rational contraction functions suffer from uncomputability.

This first result uses all the expressive power of the underlying logic. To control computability, one could limit the space of epistemic states to some specific set of theories (logically closed set of sentences). However, we show that no matter how much we constrain the space of epistemic states, uncomputability still remains, as long as the restriction is not so severe that the space collapses back to the finitary case. Although this shows that uncomputability is unavoidable in all such expressive spaces, it is of extreme importance to identify how, and under which conditions, one can construct specific (families of) contraction functions that are computable. We investigate this question for Linear Temporal Logic (LTL), and we show that when representing epistemic states via *Büchi automata* [23], we can construct families of contraction functions that are computable within such a space. LTL is a very expressive logic used in a plethora of applications in Computer Science and AI. For example, LTL has been used for specification and verification of software and hardware systems [11], in business process models such as DECLARE [24], in planning and

22nd International Workshop on Nonmonotonic Reasoning, November 2-4, 2024, Hanoi, Vietnam
*Corresponding author.
klumpp@informatik.uni-freiburg.de (D. Klumpp); ribeiroj@cardiff.ac.uk (J. S. Ribeiro)
https://dominik-klumpp.net/ (D. Klumpp); https://jandsonribeiro.github.io/home/ (J. S. Ribeiro)
0000-0003-4885-0728 (D. Klumpp); 0000-0003-1614-6808 (J. S. Ribeiro)

reasoning about actions [25, 26], and extending Description Logics with temporal knowledge [27, 28]. Büchi automata are endued with closure properties which allow for both effective reasoning and computable contraction functions.

Roadmap: In Section 2, we review basic concepts regarding logics, including LTL and Büchi automata. We briefly review AGM contraction in Section 3. Section 4 discusses the question of finite representation for epistemic states, and presents our first contribution, namely, we introduce a general notion to capture all forms of finite representations, and show a negative result: for a wide class of so-called *compendious* logics, not all epistemic states can be represented finitely. In Section 5, we present an expressive method of finite representation for LTL which is based on Büchi automata, and discuss how it supports reasoning. Section 6 introduces the notion of *AGM closedness*, i.e., every rational contraction outcome on a finitely representable belief state should again be finitely representable. We show that, under certain weak conditions, closedness cannot be satisfied for compendious logics. In Section 7, we establish our third negative result for compendious logics: even if we restrict ourselves to contraction functions whose output can be represented, uncomputability of contraction is inevitable in the non-finitary case, i.e., there always exist uncountably many uncomputable contraction functions. On the positive side, in Section 8, we show that computable contractions do exist for LTL theories represented via Büchi automata, and we identify the conditions needed for computability. Section 9 discusses the impact of our results and provides an outlook on future work.

A version of the paper including proofs is available [29].

2. Logics and Automata

We review a general notion of logics that will be used throughout the paper. We use $\mathcal{P}(X)$ to denote the power set of a set X. A *logic* is a pair $\mathbb{L} = (Fm, Cn)$ comprising a countable[1] set of *formulae* Fm, and a *consequence operator* $Cn : \mathcal{P}(Fm) \to \mathcal{P}(Fm)$ that maps each set of formulae to the conclusions entailed from it. We sometimes write $Fm_{\mathbb{L}}$ and $Cn_{\mathbb{L}}$ for brevity.

We consider logics that are *Tarskian*, that is, logics whose consequence operator Cn is monotone (if $X_1 \subseteq X_2$ then $Cn(X_1) \subseteq Cn(X_2)$), extensive ($X \subseteq Cn(X)$) and idempotent ($Cn(Cn(X)) = Cn(X)$). We say that two formulae $\varphi, \psi \in Fm$ are logically equivalent, denoted $\varphi \equiv \psi$, if $Cn(\varphi) = Cn(\psi)$. $Cn(\emptyset)$ is the set of all tautologies. A *theory* of $\mathbb{L}$ is a set of formulae $\mathcal{K}$ such that $Cn(\mathcal{K}) = \mathcal{K}$. The expansion of a theory $\mathcal{K}$ by a formula φ is the theory $\mathcal{K} + \varphi := Cn(\mathcal{K} \cup \{\varphi\})$. Let $\mathsf{Th}_{\mathbb{L}}$ denote the set of all theories of $\mathbb{L}$. If $\mathsf{Th}_{\mathbb{L}}$ is finite, we say that $\mathbb{L}$ is *finitary*; otherwise, $\mathbb{L}$ is *non-finitary*. Equivalently, $\mathbb{L}$ is finitary if $\mathbb{L}$ has only finitely many formulae up to logical equivalence.

A theory $\mathcal{K}$ is *consistent* if $\mathcal{K} \neq Fm$, and it is *complete* if for all formulae $\varphi \notin \mathcal{K}$, we have $\mathcal{K} + \varphi = Fm$. The set of all complete consistent theories of $\mathbb{L}$ is denoted as $CCT_{\mathbb{L}}$. The set of all CCTs that do not contain φ is given by $\overline{\omega}(\varphi)$.

A logic $\mathbb{L}$ is *Boolean*, if $Fm_{\mathbb{L}}$ is closed under the classical boolean operators and they are interpreted as usual. In particular, for a logic to be Boolean, we require every theory $\mathcal{K} \in \mathsf{Th}_{\mathbb{L}}$ to coincide with the intersection of all the CCTs containing $\mathcal{K}$, that is, $\mathcal{K} = \bigcap\{\, \mathcal{K}' \in CCT_{\mathbb{L}} \mid \mathcal{K} \subseteq \mathcal{K}' \,\}$.

[1] A set X is countable if there is an injection from X to the natural numbers.

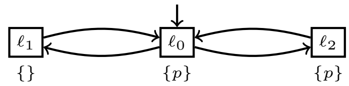


Figure 1: A Kripke structure on $AP = \{p\}$, with an initial state ℓ_0. The labels $\lambda(\ell_i)$ are shown below each state ℓ_i.

We omit subscripts whenever the meaning is clear.

2.1. Linear Temporal Logic

We recall the definition of *linear temporal logic* [11], LTL for short. For the remainder of the paper, we fix an arbitrary finite, nonempty set AP of atomic propositions.

Definition 1 (LTL Formulae)**.** *Let p range over AP.* The formulae of LTL *are generated by the following grammar:*

$$\varphi ::= \bot \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

Fm_{LTL} denotes the set of all LTL formulae.

In LTL, time is interpreted as a linear timeline that unfolds infinitely into the future. The operator $\mathbf{X}$ states that a formula holds in the *next* time step, while $\varphi\,\mathbf{U}\,\psi$ means that φ holds *until* ψ holds (and ψ does eventually hold). We define the usual abbreviations for boolean operations ($\top$, $\wedge$, $\to$) as well as the temporal operators $\mathbf{F}\,\varphi := \top\,\mathbf{U}\,\varphi$ (*finally*, at some point in the future), $\mathbf{G}\,\varphi := \neg\mathbf{F}\,\neg\varphi$ (*globally*, at all points in the future), and $\mathbf{X}^k\,\varphi$ for repeated application of $\mathbf{X}$, where $k \in \mathbb{N}$.

Formally, timelines are modelled as *traces*. A trace is an infinite sequence $\pi = a_0 a_1 \ldots$, where each $a_i \in \mathcal{P}(AP)$ is a set of atomic propositions that hold at time step i. The infinite suffix of π starting at time step i is denoted by $\pi^i = a_i a_{i+i} \cdots$. The set of all traces is denoted by $\mathcal{P}(AP)^\omega$.

The semantics of LTL is defined in terms of Kripke structures [11], which describe possible traces.

Definition 2 (Kripke Structure)**.** *A* Kripke structure *is a tuple $M = (S, I, T, \lambda)$ such that S is a finite set of states; $I \subseteq S$ is a non-empty set of initial states; $T \subseteq S \times S$ is a left-total transition relation, i.e., for all $s \in S$ there exists $s' \in S$ such that $(s, s') \in T$; and $\lambda : S \to \mathcal{P}(AP)$ labels states with sets of atomic propositions.*

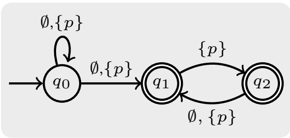
A trace of a Kripke structure M is a sequence $\pi = \lambda(s_0)\lambda(s_1)\lambda(s_2)\ldots$ with $s_0 \in I$, and for all $i \geq 0$, $s_i \in S$ and $(s_i, s_{i+1}) \in T$. The set of all traces from a Kripke structure M is given by $Traces(M)$. Figure 1 shows an example of a Kripke structure, in graphical notation.

The satisfaction relation between Kripke structure and LTL formulae is defined in terms of the satisfaction between the Kripke structure's traces and LTL formulae.

Definition 3 (Satisfaction)**.** *The* satisfaction relation *between traces and LTL formulae is the least relation $\models\ \subseteq \mathcal{P}(AP)^\omega \times Fm_{LTL}$ such that, for all traces $\pi = a_0 a_1 \ldots \in \mathcal{P}(AP)^\omega$:*

$\pi \not\models \bot$		
$\pi \models p$	*iff*	$p \in a_0$
$\pi \models \neg\varphi$	*iff*	$\pi \not\models \varphi$
$\pi \models \varphi_1 \vee \varphi_2$	*iff*	$\pi \models \varphi_1$ *or* $\pi \models \varphi_2$
$\pi \models \mathbf{X}\,\varphi$	*iff*	$\pi^1 \models \varphi$
$\pi \models \varphi_1\,\mathbf{U}\,\varphi_2$	*iff*	*there exists $i \geq 0$ s.t. $\pi^i \models \varphi_2$ and for all $j < i$, $\pi^j \models \varphi_1$*

Büchi automaton $A_{\mathcal{K}}$:



Some Infinite Words from $\mathcal{L}(A_{\mathcal{K}})$:

$\pi_1 = \emptyset\,\emptyset\,\emptyset\,\{p\}\,(\emptyset\,\{p\})^\omega$
$\pi_2 = \{p\}\,\{p\}\,\emptyset\,(\emptyset\,\{p\})^\omega$
$\pi_3 = \{p\}\,\{p\}\,\emptyset\,\{p\}^\omega$

Figure 2: A Büchi automaton (on the right), and some infinite words accepted by this automaton (on the left).

A Kripke structure M satisfies a formula φ, denoted $M \models \varphi$, iff all traces of M satisfy φ. M satisfies a set X of formulae, $M \models X$, iff $M \models \varphi$ for all $\varphi \in X$. The consequence operator Cn_{LTL} is defined from the satisfaction relation.

Definition 4 (Consequence Operator)**.** *The* consequence operator Cn_{LTL} *maps each set* X *of LTL formulae to the set of all formulae* ψ*, such that for all Kripke structures* M*, if* $M \models X$ *then also* $M \models \psi$*.*

Observation 5. *LTL is Tarskian and Boolean.*

2.2. Büchi Automata

Büchi automata are finite automata widely used in formal specification and verification of systems, specially in LTL model checking [11]. Büchi automata have also been used in planning to synthesize plans when goals are in LTL [26, 30].

Definition 6 (Büchi Automata)**.** *A* Büchi automaton *is a tuple* $A = (Q, \Sigma, \Delta, Q_0, R)$ *consisting of a finite set of states* Q*; a finite, nonempty alphabet* Σ *(whose elements are called* letters*); a transition relation* $\Delta \subseteq Q \times \Sigma \times Q$*; a set of initial states* $Q_0 \subseteq Q$*; and a set of* recurrence states $R \subseteq Q$.

We show a concrete Büchi automaton in Example 7.

A Büchi automaton accepts an infinite word over a finite alphabet Σ, if the automaton visits a recurrence state infinitely often while reading the word. Formally, an infinite word is a sequence $a_0 a_1 \ldots$ with $a_i \in \Sigma$ for all i. For a finite word $\rho = a_0 \ldots a_n$, with $n \geq 0$, let ρ^ω denote the infinite word corresponding to the infinite repetition of ρ. The set of all infinite words is denoted by Σ^ω. An infinite word $a_0 a_1 a_2 \ldots \in \Sigma^\omega$ is *accepted* by a Büchi automaton $A = (Q, \Sigma, \Delta, Q_0, R)$ if there exists a sequence $q_0, q_1, q_2, \ldots$ of states $q_i \in Q$ such that $q_0 \in Q_0$ is an initial state, for all i we have that $(q_i, a_i, q_{i+1}) \in \Delta$ and there are infinitely many $i \in \mathbb{N}$ with $q_i \in R$. The set $\mathcal{L}(A)$ of all accepted words is the *language* of A.

In this work, unless otherwise noted, we always consider Büchi automata over the alphabet $\Sigma = \mathcal{P}(AP)$, where letters are sets of atomic propositions and infinite words are traces. The following example presents such an automaton.

Example 7. *Figure 2 illustrates (on the left) a Büchi automaton* $A_{\mathcal{K}}$ *over the alphabet* $\Sigma = \{\emptyset, \{p\}\}$*. States are depicted as circles and each transition* (q, a, q') *is depicted as an arrow from* q *to* q' *labelled with* a*. The initial state is* q_0*, and the recurrence states are marked as double circles, i. e.,* q_1 *and* q_2*.*

The right-hand side of Figure 2 shows some of the infinite words accepted by the Büchi automaton $A_{\mathcal{K}}$*. Consider, for instance, the sequence*

$$q_0 \xrightarrow{\emptyset} q_0 \xrightarrow{\emptyset} q_0 \xrightarrow{\emptyset} q_1 \xrightarrow{\{p\}} q_2 \xrightarrow{\emptyset} q_1 \xrightarrow{\{p\}} q_2 \ldots,$$

where each arrow $q \xrightarrow{x} q'$ *indicates the transition* (q, x, q') *in the automaton. By concatenating the letters in this sequence, we get the infinite word* π_1 *defined in Figure 2. The acceptance condition requires some recurrence states to appear infinitely often. As for instance the recurrence state* q_1 *appears infinitely often, the acceptance condition holds and* π_1 *is accepted. Analogously, the infinite words* π_2 *and* π_3 *are also accepted.*

On the other hand, the infinite word $\pi' = \emptyset^\omega$ *is not accepted, as the only sequence of states that produces this word is* $q_0 \xrightarrow{\emptyset} q_0 \xrightarrow{\emptyset} q_0 \ldots$*, where* $q_0 \xrightarrow{\emptyset}$ *loops. The only state in this sequence is* q_0 *which is not a recurrence state and, therefore, the acceptance condition is violated.*

Emptiness of a Büchi automaton's language is decidable. Further, Büchi automata for the union, intersection and complement of the languages of given Büchi automata can be effectively constructed [23]. In the remainder of the paper we specifically make use of the construction for union and intersection, and denote them with the symbol $\sqcup$ resp. $\sqcap$. The automata-theoretic treatment for several crucial reasoning problems in LTL, such as model-checking and satisfiability, is based on the following result:

Proposition 8 ([11])**.** *For every LTL formula* φ *and every Kripke structure* M*, there exist Büchi automata* A_φ *and* A_M *that accept precisely the traces that satisfy* φ *resp. the traces of* M*, that is,* $\mathcal{L}(A_\varphi) = \{\pi \in \mathcal{P}(AP)^\omega \mid \pi \models \varphi\}$*, and* $\mathcal{L}(A_M) = \mathit{Traces}(M)$*.*

The proposition above states that every LTL formula φ can be expressed as a Büchi automaton A_φ, in the sense that A_φ accepts exactly all the traces satisfying φ. This result allows to decide if a formula φ is satisfiable, by deciding emptiness of $\mathcal{L}(A_\varphi)$. Analogously, a Büchi automaton A_M can be used to capture precisely all the traces from a given Kripke structure M, as Proposition 8 states. These two observations make it possible to decide LTL model-checking, by deciding the inclusion $\mathcal{L}(A_M) \subseteq \mathcal{L}(A_\varphi)$. In Section 5, we will exploit Proposition 8 to devise mechanisms that support the computation of belief change operators in LTL.

3. AGM Contraction

In the AGM paradigm, the epistemic state of an agent is represented as a theory. A contraction function for a theory $\mathcal{K}$ is a function $\dot{-} : Fm \to \mathcal{P}(Fm)$ that given an unwanted piece of information φ outputs a subset of $\mathcal{K}$ which does not entail φ. Contraction functions are subject to the following rationality postulates [2]:

$(\mathbf{K_1^-})$	$\mathcal{K} \dot{-} \varphi = Cn(\mathcal{K} \dot{-} \varphi)$	(closure)
$(\mathbf{K_2^-})$	$\mathcal{K} \dot{-} \varphi \subseteq \mathcal{K}$	(inclusion)
$(\mathbf{K_3^-})$	If $\varphi \notin \mathcal{K}$, then $\mathcal{K} \dot{-} \varphi = \mathcal{K}$	(vacuity)
$(\mathbf{K_4^-})$	If $\varphi \notin Cn(\emptyset)$, then $\varphi \notin \mathcal{K} \dot{-} \varphi$	(success)
$(\mathbf{K_5^-})$	$\mathcal{K} \subseteq (\mathcal{K} \dot{-} \varphi) + \varphi$	(recovery)
$(\mathbf{K_6^-})$	If $\varphi \equiv \psi$, then $\mathcal{K} \dot{-} \varphi = \mathcal{K} \dot{-} \psi$	(extensionality)
$(\mathbf{K_7^-})$	$(\mathcal{K} \dot{-} \varphi) \cap (\mathcal{K} \dot{-} \psi) \subseteq \mathcal{K} \dot{-} (\varphi \wedge \psi)$	
$(\mathbf{K_8^-})$	If $\varphi \notin \mathcal{K} \dot{-} (\varphi \wedge \psi)$ then $\mathcal{K} \dot{-} (\varphi \wedge \psi) \subseteq \mathcal{K} \dot{-} \varphi$	

For a detailed discussion on the rationale of these postulates, see [1, 2, 3]. The postulates $(\mathbf{K_1^-})$ to $(\mathbf{K_6^-})$ are called the basic rationality postulates, while $(\mathbf{K_7^-})$ and $(\mathbf{K_8^-})$ are known as supplementary postulates. A contraction function that satisfies the basic rationality postulates is called a rational contraction function. If a contraction function satisfies all the eight rationality postulates, we say that it is fully rational.

There are many different constructions for (fully) rational AGM contraction such as Partial Meet [1], Epistemic Entrenchment [2], and Kernel Contraction [6]. All these functions are characterized by the AGM postulates of contraction. For an overview, see [31, 3]. These contraction functions, however, are rational only in very specific logics, precisely in the presence of the AGM assumptions [8] which includes requiring the logic to be Boolean and compact. See [8] for details about the AGM assumptions.

To embrace more expressive logics, Ribeiro et al. [18] have proposed a new class of (fully) rational contraction functions which only assume the underlying logic to be Tarskian and Boolean: the Exhaustive Contraction Functions (for basic rationality) and the Blade Contraction Functions (for full rationality). We briefly review Exhaustive Contraction Functions. We do not delve into the Blade Contraction Functions, as our results for full rationality do not use such functions, but rather use the supplementary postulates directly.

Definition 9 (Choice Functions)**.** *A* choice function *is a function $\delta : Fm \to \mathcal{P}(CCT)$ maps each formula φ to a set of complete consistent theories satisfying the following:*

(CF1) *$\delta(\varphi) \neq \emptyset$;*

(CF2) *if $\varphi \notin Cn(\emptyset)$, then $\delta(\varphi) \subseteq \overline{\omega}(\varphi)$; and*

(CF3) *for all $\varphi, \psi \in Fm$, if $\varphi \equiv \psi$ then $\delta(\varphi) = \delta(\psi)$.*

A choice function chooses at least one complete consistent theory, for each formula φ to be contracted **(CF1)**. As long as φ is not a tautology, the CCTs chosen must not contain the formula φ **(CF2)**, since the goal is to relinquish φ. The last condition **(CF3)** imposes that a choice function is syntax-insensitive.

Definition 10 (Exhaustive Contraction Functions)**.** *Let δ be a choice function. The* Exhaustive Contraction Function (ECF) *on a theory $\mathcal{K}$ induced by δ is the function $\dot{-}_\delta$ such that $\mathcal{K} \dot{-}_\delta \varphi = \mathcal{K} \cap \bigcap \delta(\varphi)$, if $\varphi \notin Cn(\emptyset)$ and $\varphi \in \mathcal{K}$; otherwise, $\mathcal{K} \dot{-}_\delta \varphi = \mathcal{K}$.*

Whenever the formula φ to be contracted is not a tautology and is in the theory $\mathcal{K}$, an ECF modifies the current theory by selecting some CCTs and intersecting them with $\mathcal{K}$. On the other hand, if φ is either a tautology or is not in the theory $\mathcal{K}$, then all beliefs are preserved. The ECFs are similar in spirit to the standard *partial meet* functions [1]. The main difference is that partial meet relies on the internal structure of the current theory by selecting and intersecting remainders (maximal non-entailing subsets), whilst ECF chooses external structures (CCTs). In the latter, CCTs are used, because, in the absence of compactness, remainders do not exist in general [12, 18, 32].

Theorem 11. *[18] A contraction function $\dot{-}$ is rational iff it is an ECF.*

4. Limits of Finite Representation

In the AGM paradigm, the epistemic states of an agent are represented as theories which are in general infinite. However, according to Hansson [33, 34], the epistemic states of rational agents should have a finite representation. This is motivated from the perspective that epistemic states should resemble the cognitive states of human minds, and Hansson argues that as "finite beings", humans cannot sustain epistemic states that do not have a finite representation. Further, finite representation is crucial from a computational perspective, to represent epistemic states in a computer.

Different strategies of finite representation have been used such as (i) finite bases [35, 36, 37], and (ii) finite sets of models [38, 39]. In the former strategy, each finite set X of formulae, called a *finite base*, represents the theory $Cn(X)$. In the latter strategy, models are used to represent an epistemic state. Precisely, each finite set X of models represents the theory of all formulae satisfied by all models in X, that is, the theory $\{\varphi \in Fm_{\mathbb{L}} \mid M \models \varphi, \text{for all } M \in X\}$. The expressiveness of finite bases and finite sets of models are, in general (depending on the logic), incomparable, that is, some theories expressible in one method cannot be expressed in the other method and vice versa. For instance, the information that *"John swims every two days"* cannot be expressed via a finite base in LTL [40], although it can be expressed via a single Kripke structure (shown in Fig. 1, where p stands for *"John swims"*). On the other hand, *"Anna will swim eventually"* is expressible as a single LTL formula ($\mathbf{F}\, s$, where s stands for *"Anna swims"*), but it cannot be expressed via a finite set of models.

Given the incomparable expressiveness of these two well-established strategies of finite representations, it is not clear whether in general, and specifically in non-finitary logics, there exists a method capable of finitely representing all theories, therefore capturing the whole expressiveness of the logic. Towards answering this question, we provide a broad definition to conceptualise finite representation.

A finite representation for a theory can been seen as a finite word, i.e., a *code*, from a fixed finite alphabet Σ_C. For example, the codes $c_1 := \{\texttt{a, b}\}$ and $c_2 := \{\texttt{a, a}\to\texttt{b}\}$ are finite words in the language of set theory, and both represent the theory $Cn(\{a \wedge b\})$. The set of all codes, i.e., of all finite words over Σ_C, is denoted by Σ_C^*. In this sense, a method of finite representation is a mapping f from codes in Σ_C^* to theories. The pair (Σ_C, f) is called an *encoding*.

Definition 12 (Encoding)**.** *An* encoding (Σ_C, f) *comprises a finite alphabet Σ_C and a partial function $f : \Sigma_\text{C}^* \rightharpoonup \mathsf{Th}_{\mathbb{L}}$.*

Given an encoding (Σ_C, f), a word $w \in \Sigma_\text{C}^*$ represents a theory $\mathcal{K}$, if $f(w)$ is defined and $f(w) = \mathcal{K}$. Observe that a theory might have more than one code, whereas for others there might not exist a code. For instance, in the example above for finite bases, the codes c_1 and c_2 represent the same theory. On the other hand, recall that the LTL theory corresponding to *"John swims every two days"* cannot be expressed in the finite base encoding. Furthermore, the function f is partial, because not all codes in Σ_C^* are meaningful. For instance, for the finite base encoding, the code $\{\{\}\}$ cannot be interpreted as a finite base.

We are interested in logics which are AGM compliant, that is, logics in which rational contraction functions exist. Unfortunately, it is still an open problem how to construct AGM contraction functions in all such logics. The most general constructive apparatus up to date, as discussed in Section 3, are the Exhaustive Contraction functions proposed by Ribeiro et al. (2018) which assume only few conditions on the logic. Additionally, we focus on non-finitary logics, as the finitary case is trivial. We call such logics *compendious*.

Definition 13 (Compendious Logics)**.** *A logic $\mathbb{L}$ is* compendious *if $\mathbb{L}$ is Tarskian, Boolean, non-finitary and satisfies:*

(Discerning) *For all sets $X, Y \subseteq CCT_{\mathbb{L}}$, we have that $\bigcap X = \bigcap Y$ implies $X = Y$.*

Compendiousness amounts to expressivity in multiple dimensions. Compendious logics can express infinitely many distinct sentences (non-finitary), distinguish between a sentence being true or false (classical negation), and express uncertainty of two or more sentences (disjunction). The property **(Discerning)** is related the possible worlds semantics. In a possible world, the truth values of all sentences are known. From this perspective, possible worlds correspond to CCTs. Under the possible worlds semantics, an agent's epistemic state is interpreted as the exact set of all possible worlds in which all the agent's beliefs are true. If the truth value of a formula φ is unknown, the agent considers some possible worlds where φ is true, as well as possible worlds where φ is false. Hence, more possible worlds indicate strictly less information. Equivalently, different sets of possible worlds represent different epistemic states. This is exactly what **(Discerning)** conceptualises.

Example 14 (Discerning)**.** *Yara and Yasmin encounter a large flightless bird. Yara knows that such birds exist in Africa and South America. Hence, Yara considers two possible worlds: the bird is from Africa (it is an ostrich), or the bird is from South America (it is a rhea). Yasmin, who lived in Australia, believes the bird is an emu (from Australia), a rhea or an ostrich. Since Yara and Yasmin consider different sets of possible worlds, their epistemic states differ. Yara believes in the disjunction $ostrich \lor rhea$, Yasmin does not. She believes only in the disjunction $ostrich \lor rhea \lor emu$. As per **(Discerning)**, Yara and Yasmin present different epistemic states, due to the difference in the considered possible worlds.*

The class of compendious logics is broad and includes several widely used logics.

Theorem 15. *The logics LTL, CTL, CTL*, μ-calculus and* monadic second-order logic *(MSO) are compendious.*

It turns out that there is no method of finite representation capable of capturing all theories in a compendious logic.

Theorem 16. *No encoding can represent every theory of a compendious logic.*

Proof Sketch. We show that, since compendious logics are Tarskian, Boolean and non-finitary, there exist infinitely many CCTs. From **(Discerning)**, it follows that there exist uncountably many theories in the logic. However, an encoding can represent only countably many theories. □

As not every theory cann be finitely represented, only some subsets of theories can be used to express the epistemic states of an agent. We call a subset $\mathbb{E}$ of theories an *excerpt* of the logic. Each encoding induces an excerpt.

Definition 17. *The excerpt* induced *by an encoding* (Σ_C, f) *is the set* $\mathbb{E} := \mathsf{img}(f)$*. An excerpt induced by some encoding is called* finitely representable.

5. The Büchi Encoding of LTL

The encoding in which epistemic states are expressed is of fundamental importance. On the one hand, the encoding must be expressive enough to capture a non-trivial space of epistemic states. On the other hand, the encoding must support reasoning. Most fundamentally, an agent should be able to decide whether it believes a given formula φ, i.e., whether φ is entailed by the theory representing the agent's epistemic state. This question might be decidable for one (perhaps less expressive) encoding, but undecidable for a different (more expressive) encoding. We call this question the *entailment problem on an encoding* (Σ_C, f):

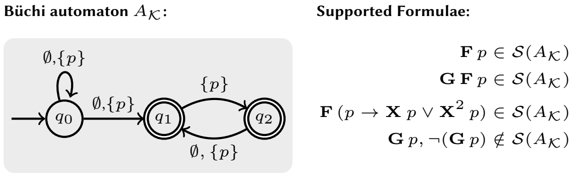


Figure 3: A Büchi automaton, along with some examples of supported (and not supported) LTL formulae.

Input:	$(w, \varphi) \in \Sigma_C^* \times Fm_{\mathbb{L}}$, such that $f(w)$ is defined
Output:	*true* if $\varphi \in f(w)$, otherwise *false*

This problem is a generalisation of several decision problems that support reasoning. For example, on the finite base encoding, it corresponds to the usual entailment problem between formulae. Entailment on the encoding based on finite sets of models corresponds to the model checking problem. For other encodings, as we will see, it can be more general than either of these problems.

We investigate a suitable encoding for epistemic states over LTL, a commonly used compendious logic in model checking and planning. In both these domains, the primary approach to reason about LTL is based on Büchi automata. Thus, from a reasoning standpoint, Büchi automata are predestined to be the basis for an encoding of epistemic states over LTL. We give the following definition for the set of LTL formulae represented by a Büchi automaton:

Definition 18 (Support)**.** *The* support *of a Büchi automaton A is the set $\mathcal{S}(A) := \{ \varphi \in Fm_{LTL} \mid \forall \pi \in \mathcal{L}(A) \,.\, \pi \models \varphi \}$. If $\varphi \in \mathcal{S}(A)$, we say that A* supports *φ.*

Example 19. *Figure 3 shows a Büchi automaton (on the left), along with three supported formulae (on the right). All accepted traces satisfy these formulae. The formula $\mathbf{G}\, p$ is not supported. While some accepted traces, such as $\{p\}^\omega$, satisfy this formula, others, such as $\emptyset\,\{p\}^\omega$ do not. Consequently, the negation $\neg(\mathbf{G}\, p)$ is not supported either.*

It remains to show that the support of a Büchi automaton is a theory. Perhaps surprisingly, the support of an arbitrary language of infinite traces (not represented as a Büchi automaton) does not necessarily form a theory. The disconnect arises from the fact that the semantics of LTL is defined over finite Kripke structures, and arbitrary languages of infinite traces can represent more fine-grained nuances of behaviours. Consider the language $L_{\mathsf{prime}} = \{a_0 a_1 a_2 \ldots\}$, where $a_i = \{p\}$ if i is a prime number and $a_i = \emptyset$ otherwise. The support of L_{prime} prescribes that p holds exactly in prime-numbered steps. Since no Kripke structure satisfies this requirement, the support of L_{prime} is inconsistent, yet L_{prime} does not support $\bot$.

An intriguing property of Büchi automata however is that their support is fully determined by those accepted traces π that have the property of being *ultimately periodic*, that is, $\pi = \rho\,\sigma^\omega$ for some finite sequences ρ, σ. Recall from Section 2.2 that the superscript ω denotes infinite repetition of the subsequence σ. Ultimately periodic traces are tightly connected to CCTs: each CCT is satisfied by exactly one ultimately periodic trace. Let UP denote the set of all ultimately periodic traces. The correspondence between CCTs and ultimately periodic traces is formalized by the function $Th_{UP} : UP \to CCT_{LTL}$ such that $Th_{UP}(\pi) = \{\varphi \in Fm_{LTL} \mid \pi \models \varphi\}$.

Lemma 20. *The function Th_{UP} is a bijection.*

We combine Lemma 20 with two classical observations [11]: (i) every consistent LTL formula is satisfied by at least one ultimately periodic trace; and (ii) every Büchi automaton with nonempty language accepts some ultimately periodic trace. We arrive at the following characterization:

Lemma 21. *The support of a Büchi automaton A satisfies*

$$\mathcal{S}(A) = \bigcap \{\, Th_{UP}(\pi) \mid \pi \in \mathcal{L}(A) \cap UP \,\}.$$

Theorem 22. *The support of a Büchi automaton is a theory.*

Thus, Büchi automata indeed define an encoding. Every Büchi automaton A, being a finite structure, can be described in a finite code word w_A, which the encoding maps to the theory $\mathcal{S}(A)$. We call this encoding the *Büchi encoding*, denoted $(\Sigma_{\text{Büchi}}, f_{\text{Büchi}})$, and the induced excerpt the *Büchi excerpt* $\mathbb{E}_{\text{Büchi}}$. In terms of expressiveness, the Büchi excerpt strictly subsumes the classical approaches:

Theorem 23. *Let $\mathbb{E}_{\text{base}}$ and $\mathbb{E}_{\text{models}}$ denote respectively the excerpts of finite bases and finite sets of models[2]. It holds that $\mathbb{E}_{\text{base}} \cup \mathbb{E}_{\text{models}} \subsetneq \mathbb{E}_{\text{Büchi}}$.*

Proof Sketch. The expressiveness of the Büchi excerpt follows from Proposition 8. Figure 3 shows an automaton whose support can be expressed neither by a finite base nor a finite sets of models. □

In terms of reasoning, the Büchi encoding benefits from the decidability properties of Büchi automata. Many decision problems, most importantly the entailment problem on the Büchi encoding, can be reduced to the decidable problem of inclusion between Büchi automata.

Theorem 24. *The entailment problem on the Büchi encoding is decidable.*

Proof. Given a word $w \in \Sigma^*_{\text{Büchi}}$ that encodes a Büchi automaton A_w, and an LTL formula φ, one can decide whether $\varphi \in f_{\text{Büchi}}(w) = \mathcal{S}(A_w)$ by deciding the Büchi automata inclusion $\mathcal{L}(A_w) \subseteq \mathcal{L}(A_\varphi)$. □

Beyond ensuring the decidability of key problems, an encoding's suitability for reasoning also involves the question whether modifications of epistemic states can be realized by computations on code words. In particular in the context of the AGM paradigm, it is interesting to see if belief change operations can be performed in such a manner. The Büchi encoding also shines in this respect, since we can employ automata operations to this end. As a first example, consider the *expansion* of a theory $\mathcal{K}$ with a formula φ. This operation corresponds to an intersection operation on Büchi automata, as the support of a Büchi automaton satisfies $\mathcal{S}(A) + \varphi = \mathcal{S}(A \sqcap A_\varphi)$. The intersection automaton $A \sqcap A_\varphi$ can be computed through a product construction.

[2]These excerpts were described in the prologue of Section 4.

By contrast, the following two sections discuss fundamental limitations to effective constructions for rational *contractions*. Nevertheless, we show in Section 8 how the Büchi encoding admits similar automata-based constructions for a large subclass of contraction functions.

6. AGM Accommodation

Assume that the space of epistemic states that an agent can entertain is determined by an excerpt $\mathbb{E}$. In this section, we investigate which properties make an excerpt suitable from the AGM vantage point. Clearly, not every excerpt is suitable for representing the space of epistemic states. For example, if a non-tautological formula φ appears in each theory of $\mathbb{E}$, then φ cannot be contracted. The chosen excerpt should be expressive enough to describe all relevant epistemic states that an agent might hold in response to its beliefs in flux. Precisely, if an agent is confronted with a piece of information and changes its epistemic state into a new one, then this new epistemic state must be expressible in the underlying excerpt. A straightforward option would be to require some sort of closure under AGM rationality, that is, all possible rational contractions involving information in the excerpt should be expressed yet within the excerpt. Such excerpts are *closed under rational contraction* resp. *under fully rational contraction.* We say that a contraction $\dot{-}$ *remains in* $\mathbb{E}$ if $\mathsf{img}(\dot{-}) \subseteq \mathbb{E}$.

Definition 25 (Closedness)**.** *An excerpt $\mathbb{E}$ of a logic $\mathbb{L}$ is* closed under (fully) rational contraction *iff for all theories $\mathcal{K} \in \mathbb{E}$, every (fully) rational contraction operation on $\mathcal{K}$ remains in $\mathbb{E}$.*

Closedness maximises the expressiveness of the excerpt w.r.t. AGM rationality: in each epistemic state of the excerpt, every possible (fully) AGM rational contraction outcome is at disposal. However, although closedness might seem like a reasonable condition, it turns out to be very demanding. For example, as we are dealing with Boolean logics, which are closed under classical negation, an agent should be able to either accept or reject some pieces of information. The excerpt should be broad enough such that there exists some piece of information φ, where both φ and $\neg\varphi$ occur in some, possibly different, epistemic states of the excerpt. We call such excerpts *open-minded.* Even under such an innocuous condition, an agent cannot express its epistemic states in an excerpt that is closed under rational contraction: closedness rules out finite representability.

Theorem 26 (Impossibility of Closedness)**.** *If $\mathbb{E}$ is an open-minded, finitely representable excerpt of a compendious logic, then $\mathbb{E}$ is not closed under rational contraction.*

Proof Sketch. From open-mindedness, it follows that there exists a formula φ in a theory $\mathcal{K}$ of the excerpt, such that $\overline{\omega}(\varphi)$ is infinite. Then there are already uncountably many ways to contract φ. However, the finitely representable excerpt contains only countably many theories. □

The negative result above concerns excerpts that are closed under rational contraction. As full rationality is strictly more demanding than rationality, one could hope to reach closedness by restricting to excerpts closed under fully rational contraction. Unfortunately, rationally closed excerpts and fully rationally closed excerpts coincide.

Proposition 27. *An excerpt is closed under fully rational contraction iff it is closed under rational contraction.*

Instead of insisting on maximising the expressiveness of the excerpts, we impose a weaker condition and require the excerpt only to admit at least one rational outcome for each possible contraction.

Definition 28 (Accommodation). *An excerpt* $\mathbb{E}$ accommodates (fully) rational contraction *iff for each* $\mathcal{K} \in \mathbb{E}$ *there exists a (fully) rational contraction on* $\mathcal{K}$ *that remains in* $\mathbb{E}$.

Accommodation guarantees that an agent can modify its beliefs rationally, in all possible epistemic states covered by the excerpt. There is a clear connection between accommodation and AGM compliance. While AGM compliance concerns existence of rational contraction operations in every theory of a logic, accommodation guarantees that the information in each theory within the excerpt can be rationally contracted and that its outcome can yet be expressed within the excerpt. Analogous to closedness, rational accommodation and fully rational accommodation coincide.

Proposition 29. *An excerpt* $\mathbb{E}$ *accommodates rational contraction iff* $\mathbb{E}$ *accommodates fully rational contraction.*

7. Uncomputability of Contraction

Accommodation is the weakest condition we can impose upon an excerpt to guarantee the existence of AGM rational contractions. Yet, the existence of contractions does not imply that an agent can *effectively* contract information. Thus we investigate the question of *computability* of contraction functions. For this endeavor, the focus on contraction functions that remain in the excerpt is crucial: both input and output of a computation must be finitely representable. We thus fix a finitely representable excerpt $\mathbb{E}$ that accommodates contraction. As an agent has to reason about its beliefs, it should be able to decide whether two formulae are logically equivalent. Hence, we assume that, in the underlying logic, logical equivalence is decidable.

Definition 30 (AGM Computability). *Let* $\mathcal{K}$ *be a theory in* $\mathbb{E}$*, and let* $\dot{-}$ *be a contraction function on* $\mathcal{K}$ *that remains in* $\mathbb{E}$*. We say that* $\dot{-}$ *is* computable *if there exists an encoding* (Σ_C, f) *that induces* $\mathbb{E}$*, such that the following problem is computed by a Turing machine:*

Input: *A formula* $\varphi \in Fm_{\mathbb{L}}$.

Output: *A word* $w \in \Sigma_C^*$ *such that* $f(w) = \mathcal{K} \dot{-} \varphi$.

In the classical setting of finitary logics, computability of AGM contraction is trivial, as there are only finitely many formulae (up to equivalence), and only a finite number of theories. By contrast, compendious logics have infinitely many formulae (up to equivalence) and consequently infinitely many theories.

In the following, unless otherwise stated, we only consider compendious logics. In such logics, we distinguish two kinds of theories: those that contain infinitely many formulae (up to equivalence), and those that contain only finitely many formulae (up to equivalence). An excerpt that constrains an agent's epistemic states to the latter case essentially disposes of the expressive power of the compendious logic, as in each epistemic state only finitely many sentences can be distinguished. Therefore, such epistemic states could be expressed in a finitary logic. As the computability in the finitary case is trivial, we focus on the more expressive case.

Definition 31 (Non-Finitary). *A theory* $\mathcal{K}$ *is* non-finitary *if it contains infinitely many logical equivalence classes of formulae.*

Note that being non-finitary is a very general condition. Even theories with a finite base can be non-finitary. For instance, the LTL theory $Cn(\mathbf{G}\,p)$ contains the infinitely many non-equivalent formulae $\{p, \mathbf{X}\,p, \mathbf{X}^2\,p, \mathbf{X}^3\,p, \ldots\}$.

In the remainder of this section, we establish a strong link between non-finitary theories and uncomputable contraction functions. To this end, we introduce the notion of *cleavings.*

Definition 32 (Cleaving). *A* cleaving *is an infinite set of formulae* $\mathcal{C}$ *such that for all two distinct* $\varphi, \psi \in \mathcal{C}$ *we have:*

(CL1) φ *and* ψ *are not equivalent* ($\varphi \not\equiv \psi$)*; and*

(CL2) *the disjunction* $\varphi \vee \psi$ *is a tautology.*

From an algebraic perspective, the formulae in a cleaving behave like a kind of weak complement: we require that the disjunction $\varphi \vee \psi$ is a tautology, whereas we do not require the conjunction $\varphi \wedge \psi$ to be inconsistent (as would be the case for the conjunction $\varphi \wedge \neg\varphi$).

Example 33. *Consider the logic of elementary arithmetic over natural numbers. The formulae* $x \neq 0$, $x \neq 1$, $x \neq 2$, *etc. form a cleaving: they are pairwise non-equivalent, and every disjunction* $(x \neq i) \vee (x \neq j)$ *is a tautology (where* i, j *are two different constants).*

Example 34. *Let* $twice(p) :\equiv \mathbf{F}\,(p \wedge \mathbf{X}\,\mathbf{F}\,p)$ *be the LTL formula denoting that proposition* p *holds at least twice, and for* $i \in \mathbb{N}$*, let* $\psi_i :\equiv (\mathbf{X}^i\,p) \rightarrow twice(p)$ *be the formula stating that if* p *holds in time step* i*, it must hold at least one additional time (i.e., at least twice overall). For* $i \neq j$*, the formulae* ψ_i *and* ψ_j *are non-equivalent. Further, the disjunction* $\psi_i \vee \psi_j$ *simplifies to* $(\mathbf{X}^i\,p) \wedge (\mathbf{X}^j\,p) \rightarrow twice(p)$*. The latter formula is a tautology: if* p *holds in time steps* i *and* j*, it holds at least twice. Hence, the set of formulae* $\{\psi_0, \psi_1, \ldots\}$ *is a cleaving.*

Lemma 35. *Every non-finitary theory contains a cleaving.*

Given a contraction that remains in an excerpt, cleavings provide a way of generating many different contractions that remain within the excerpt. This works by ranking the formulae in the cleaving such that each rank has exactly one formula. We reduce the contraction of a formula φ to contracting $\varphi \vee \psi$, where ψ is the lowest ranked formula in the cleaving such that $\varphi \vee \psi$ is non-tautological. Each new contraction depends on the original choice function and the ranking.

Definition 36 (Composition). *Let* δ *be a choice function on a theory* $\mathcal{K}$, $\mathcal{C} \subseteq \mathcal{K}$ *be a cleaving, and* $\pi : \mathbb{N} \to \mathcal{C}$ *be a permutation of* $\mathcal{C}$*. The* composition of δ and π *is the function* $\delta_\pi : Fm \to \mathcal{P}(CCT)$ *such that*

$$\delta_\pi(\varphi) := \delta\big(\varphi \vee \min_\pi(\varphi)\big)$$

where $\min_\pi(\varphi) = \pi(i)$*, for the minimal* $i \in \mathbb{N}$ *such that* $\varphi \vee \pi(i) \not\equiv \top$*, or* $\min_\pi(\varphi) = \bot$ *if no such* i *exists.*

Example 37. *Consider the cleaving* $\{\psi_0, \psi_1, \ldots\}$ *of Example 34, and let* π *be the permutation with* $\pi(i) = \psi_i$ *for all* $i \in \mathbb{N}$*. The formula* $p \vee \psi_0$ *is a tautology. Thus, we have* $\min_\pi(p) = \pi(1) = \psi_1$ *and the choice function chooses* $\delta_\pi(p) = \delta(p \vee \psi_1)$*. If we however consider a permutation* π' *with* $\pi'(0) = \psi_2$ *and* $\pi'(2) = \psi_0$*, then we have* $\min_{\pi'}(p) = \pi'(0) = \psi_2$ *and* $\delta_{\pi'}(p) = \delta(p \vee \psi_2)$*.*

If we consider the formula $\mathbf{F}\,\mathbf{G}\,\neg p$ *stating that* p *only holds finitely often, any disjunction of the form* $(\mathbf{F}\,\mathbf{G}\,\neg p) \vee \psi_i$ *is a tautology: either there are only finitely many occurrences of* p*, or otherwise,* p *holds infinitely often, and so* p *holds at least twice. Hence, we have, for both permutations,* $\delta_\pi(\mathbf{F}\,\mathbf{G}\,\neg p) = \delta_{\pi'}(\mathbf{F}\,\mathbf{G}\,\neg p) = \delta\big((\mathbf{F}\,\mathbf{G}\,\neg p) \vee \bot\big) = \delta(\mathbf{F}\,\mathbf{G}\,\neg p)$*.*

The composition of a choice function δ with a permutation of a cleaving preserves rationality.

Lemma 38. *The composition* δ_π *of a choice function* δ *and a permutation* $\pi : \mathbb{N} \to \mathcal{C}$ *of a cleaving* $\mathcal{C} \subseteq \mathcal{K}$ *is a choice function.*

A composition generates a new choice function which in turn induces a rational contraction function that remains within the excerpt. Yet, the induced contraction function is not necessarily computable.

Theorem 39. *Let* $\mathbb{E}$ *accommodate rational contraction, and let* $\mathcal{K} \in \mathbb{E}$*. The following statements are equivalent:*

1. *The theory* $\mathcal{K}$ *is non-finitary.*
2. *There exists an* uncomputable *rational contraction function on* $\mathcal{K}$ *that remains in* $\mathbb{E}$*.*
3. *There exists an* uncomputable *fully rational contraction function on* $\mathcal{K}$ *that remains in* $\mathbb{E}$*.*

Proof Sketch. Let $\mathcal{K}$ be non-finitary, and δ the choice function of a (fully) rational contraction for $\mathcal{K}$ that remains in $\mathbb{E}$. Each permutation π of a cleaving $\mathcal{C} \subseteq \mathcal{K}$ induces a *distinct* (fully) rational contraction (with choice function δ_π) that remains in $\mathbb{E}$. At most countably many of these uncountably many (fully) rational contractions can be computable.

If $\mathcal{K}$ is finitary, every contraction function is computable, as it only has to consider finitely many formulae. ☐

Theorem 39 makes evident that uncomputability of AGM contraction is inevitable. In fact, there are uncountably many uncomputable contraction functions. The only way to avoid this uncomputability would be to restrain the expressiveness of the excerpt to the most trivial case: only finitary theories.

8. Effective Contraction in the Büchi Excerpt

Despite the strong negative result of Section 7, computability can still be harnessed in very particular excerpts: excerpts $\mathbb{E}$ in which for every theory, there exists at least one computable (fully) rational contraction function that remains in $\mathbb{E}$. We say that such an excerpt $\mathbb{E}$ *effectively accommodates* (fully) rational contraction. If belief contraction is to be computed for compendious logics, it is paramount to identify such excerpts as well as classes of computable contraction functions. In this section, we show that the Büchi excerpt of LTL effectively accommodates (fully) rational contraction.

For a contraction on a theory $\mathcal{K} \in \mathbb{E}_{\text{Büchi}}$ to remain in the Büchi excerpt, the underlying choice function must be designed such that the intersection of $\mathcal{K}$ with the selected CCTs corresponds to the support of a Büchi automaton. As CCTs and ultimately periodic traces are interchangeable (Lemma 20), and the support of a Büchi automaton is determined by the CCTs corresponding to its accepted ultimately periodic traces (Lemma 21), a solution is to design a selection mechanism, analogous to choice functions, that picks a single Büchi automaton instead of an (infinite) set of CCTs.

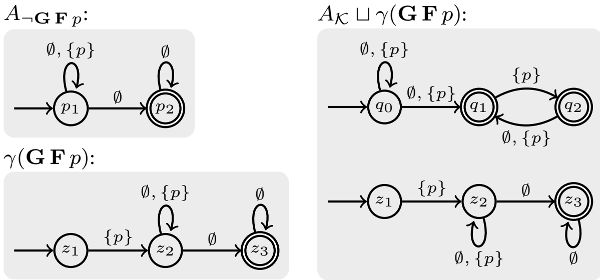


Figure 4: BCF contraction of $\mathbf{G}\,\mathbf{F}\,p$ from $\mathcal{S}(A_\mathcal{K})$ (Example 43).

Definition 40 (Büchi Choice Functions)**.** *A* Büchi choice function γ *maps each LTL formula to a single Büchi automaton, such that for all LTL formulae* φ *and* ψ*,*

(BF1) *the language accepted by* $\gamma(\varphi)$ *is non-empty;*

(BF2) $\gamma(\varphi)$ *supports* $\neg\varphi$*, if* φ *is not a tautology; and*

(BF3) $\gamma(\varphi)$ *and* $\gamma(\psi)$ *accept the same language, if* $\varphi \equiv \psi$*.*

Conditions **(BF1)** - **(BF3)** correspond to the respective conditions **(CF1)** - **(CF3)** in the definition of choice functions. Each Büchi choice function induces a contraction function.

Definition 41 (Büchi Contraction Functions)**.** *Let* $\mathcal{K}$ *be a theory in the Büchi excerpt and let* γ *be a Büchi choice function. The* Büchi Contraction Function (BCF) *on* $\mathcal{K}$ *induced by* γ *is the function*

$$\mathcal{K} \mathbin{\dot{-}}_\gamma \varphi = \begin{cases} \mathcal{K} \cap \mathcal{S}(\gamma(\varphi)) & \textbf{\textit{if}}\ \varphi \notin Cn(\emptyset) \text{ and } \varphi \in \mathcal{K} \\ \mathcal{K} & \textbf{\textit{otherwise}} \end{cases}$$

All such contractions remain in the Büchi excerpt. Indeed, one can observe that if $\mathcal{K} = \mathcal{S}(A)$ for a Büchi automaton A, it holds that $\mathcal{K} \cap \mathcal{S}(\gamma(\varphi)) = \mathcal{S}(A \sqcup \gamma(\varphi))$. Recall from Section 2 that $\sqcup$ denotes the union of Büchi automata. The class of all rational contraction functions that remain in the Büchi excerpt corresponds exactly to the class of all BCFs.

Theorem 42. *A contraction function* $\dot{-}$ *on a theory* $\mathcal{K} \in \mathbb{E}_{\text{Büchi}}$ *is rational and remains within the Büchi excerpt if and only if* $\dot{-}$ *is a BCF.*

Example 43. *Let* $\mathcal{K} = \mathcal{S}(A_\mathcal{K})$*, for the Büchi automaton* $A_\mathcal{K}$ *shown in Fig. 3. To contract the formula* $\mathbf{G}\,\mathbf{F}\,p$*, a Büchi choice function* γ *may select the Büchi automaton* $\gamma(\mathbf{G}\,\mathbf{F}\,p)$ *shown in Fig. 4. This automaton supports* $\neg\mathbf{G}\,\mathbf{F}\,p$*; the automaton* $A_{\neg\mathbf{G}\,\mathbf{F}\,p}$ *is shown for reference. In fact,* $\gamma(\mathbf{G}\,\mathbf{F}\,p)$ *accepts precisely the traces satisfying* $p \wedge \neg\mathbf{G}\,\mathbf{F}\,p$*. The result of the contraction is the theory* $\mathcal{S}(A_\mathcal{K}) \cap \mathcal{S}(\gamma(\mathbf{G}\,\mathbf{F}\,p))$*, which corresponds to the theory* $\mathcal{S}(A_\mathcal{K} \sqcup \gamma(\mathbf{G}\,\mathbf{F}\,p))$*, whose supporting automaton is also shown in Fig. 4. The union* $\sqcup$ *is obtained by simply taking the union of states and transitions. This automaton does not support* $\mathbf{G}\,\mathbf{F}\,p$*, and therefore the contraction is successful.*

As BCFs capture all rational contractions within the excerpt, it follows from Theorem 39 that not all BCFs are computable. Note from Definition 41 that to achieve computability, it suffices to be able to: (i) decide if φ is a tautology, (ii) decide if $\varphi \in \mathcal{K}$, (iii) compute the underlying Büchi choice function γ, and (iv) compute the intersection of $\mathcal{K}$ with the support of $\gamma(\varphi)$. We already know that conditions (i) and (ii) are satisfied (Theorem 24). For condition (iv), recall that the intersection of the support of two automata is equivalent to the support of their union. As γ produces a Büchi automaton, and union of Büchi automata is computable, condition (iv) is also satisfied. Therefore, condition (iii) is the only one remaining. It turns out that (iii) is a necessary and sufficient condition to characterize all computable contraction functions within the Büchi excerpt.

Theorem 44. *Let $\dot{-}$ be a rational contraction function on a theory $\mathcal{K} \in \mathbb{E}_{\text{Büchi}}$, such that $\dot{-}$ remains in the Büchi excerpt. The operation $\dot{-}$ is computable iff $\dot{-} = \dot{-}_\gamma$ for some computable Büchi choice function γ .*

Note that there do indeed exist computable choice functions. As an example, the *full meet* contraction [1, 3] is computable. The corresponding Büchi choice function γ_{fm} is given by $\gamma_{\text{fm}}(\varphi) = A_{\neg\varphi}$ if φ is non-tautological, and $\gamma_{\text{fm}}(\varphi) = A_\top$ otherwise. This function is computable: the automata $A_{\neg\varphi}$ and $A_\top$ can be effectively constructed, and it is decidable whether the given LTL formula φ is a tautology.

As the *full meet* contraction is fully rational, we conclude that the Büchi excerpt effectively accommodates (fully) rational contraction.

9. Conclusion

We have investigated the computability of AGM contraction for the class of compendious logics, which embrace several logics used in computer science and AI. Due to the high expressive power of these logics, not all epistemic states admit a finite representation. Hence, the epistemic states that an agent can assume are confined to a space of theories, which depends on a method of finite representation. We have shown a severe negative result: no matter which form of finite representation we use, as long as it does not collapse to the finitary case, AGM contraction suffers from uncomputability. Precisely, there are uncountably many uncomputable (fully) rational contraction functions in all such expressive spaces. This negative result also impacts other forms of belief change. For instance, belief revision is interdefinable with belief contraction, via the Levi Identity. Therefore, it is likely that revision also suffers from uncomputability. Accordingly, uncomputability might span to iterated belief revision [41], update and erasure [42], and pseudo-contraction [43], to cite a few. It is worth investigating uncomputability of these other operators.

In this work, we have focused on the AGM paradigm, and logics which are Boolean. We intend to expand our results for a wider class of logics by dispensing with the Boolean operators, and assuming only that the logic is AGM compliant. We believe the results shall hold in the more general case, as our negative results follow from cardinality arguments. On the other hand, several logics used in knowledge representation and reasoning are not AGM compliant, as for instance a variety of Description Logics [12]. In these logics, the *recovery* postulate ($\mathbf{K_5^-}$) can be replaced by the *relevance* postulate [44], and contraction functions can be properly defined. Such logics are called relevance-compliant. As relevance is an weakened version of recovery, the uncomputability results in this work translate to various relevance-compliant logics. However, it is unclear if all such logics are affected by uncomputability. We aim to investigate this issue in such logics.

Even if we have to coexist with uncomputability, we can still identify classes of operators which are guaranteed to be computable. To this end, we have introduced a novel class of contraction functions for LTL using Büchi automata, and identified the conditions needed for computability. This is an initial step towards the application of belief change in other areas, such as methods for automatically repairing systems [45]. The methods devised here for LTL form a foundation for the development of analogous strategies for other expressive logics, such as CTL, μ-calculus and many Description Logics. For example, in these logics, similarly to LTL, decision problems such as satisfiability and entailment have been solved using various kinds of automata, such as tree automata [46, 47].

References

[1] C. E. Alchourrón, P. Gärdenfors, D. Makinson, On the logic of theory change: Partial meet contraction and revision functions, J. Symb. Log. 50 (1985) 510–530. doi:10.2307/2274239.

[2] P. Gärdenfors, Knowledge in flux: Modeling the dynamics of epistemic states., The MIT press, 1988.

[3] S. O. Hansson, A textbook of belief dynamics - theory change and database updating, volume 11 of *Applied logic series*, Kluwer, 1999.

[4] C. E. Alchourrón, D. Makinson, On the logic of theory change: Contraction functions and their associated revision functions, Theoria 48 (1982) 14–37. doi:10.1111/j.1755-2567.1982.tb00480.x.

[5] P. Gärdenfors, An epistemic approach to conditionals, American Philosophical Quarterly 18 (1981) 203–211.

[6] S. O. Hansson, Kernel contraction, J. Symb. Log. 59 (1994) 845–859. doi:10.2307/2275912.

[7] A. Grove, Two modellings for theory change, J. Philos. Log. 17 (1988) 157–170. doi:10.1007/BF00247909.

[8] G. Flouris, On Belief Change and Ontology Evolution, Ph.D. thesis, University of Crete, 2006.

[9] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[10] P. Blackburn, J. F. A. K. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, volume 3 of *Studies in logic and practical reasoning*, North-Holland, 2007.

[11] E. M. Clarke, O. Grumberg, D. Kroening, D. A. Peled, H. Veith, Model checking, 2nd Edition, MIT Press, 2018.

[12] M. M. Ribeiro, R. Wassermann, G. Flouris, G. Antoniou, Minimal change: Relevance and recovery revisited, Artif. Intell. 201 (2013) 59–80. doi:10.1016/J.ARTINT.2013.06.001.

[13] J. P. Delgrande, P. Peppas, Belief revision in Horn theories, Artif. Intell. 218 (2015) 1–22. doi:10.1016/J.ARTINT.2014.08.006.

[14] J. P. Delgrande, R. Wassermann, Horn clause contraction functions: Belief set and belief base approaches, in: KR, AAAI Press, 2010.

[15] R. Booth, T. Meyer, I. Varzinczak, R. Wassermann,

On the link between partial meet, kernel, and infra contraction and its application to Horn logic, CoRR abs/1401.3902 (2014). arXiv:1401.3902.
[16] N. C. da Costa, O. Bueno, Belief change and inconsistency, Logique et Analyse 41 (1998) 31–56.
[17] M. M. Ribeiro, R. Wassermann, Base revision for ontology debugging, J. Log. Comput. 19 (2009) 721–743. doi:10.1093/LOGCOM/EXN048.
[18] J. S. Ribeiro, A. Nayak, R. Wassermann, Towards belief contraction without compactness, in: KR, AAAI Press, 2018, pp. 287–296.
[19] R. Wassermann, On AGM for non-classical logics, J. Philos. Log. 40 (2011) 271–294. doi:10.1007/S10992-011-9178-2.
[20] B. Nebel, How Hard is it to Revise a Belief Base?, Springer Netherlands, 1998, pp. 77–145.
[21] T. Eiter, G. Gottlob, On the complexity of propositional knowledge base revision, updates, and counterfactuals, Artif. Intell. 57 (1992) 227–270. doi:10.1016/0004-3702(92)90018-S.
[22] N. Schwind, S. Konieczny, J. Lagniez, P. Marquis, On computational aspects of iterated belief change, in: IJCAI, ijcai.org, 2020, pp. 1770–1776. doi:10.24963/IJCAI.2020/245.
[23] J. Richard Büchi, Symposium on Decision Problems: On a Decision Method in Restricted Second Order Arithmetic, volume 44 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, 1966, pp. 1–11. doi:10.1016/S0049-237X(09)70564-6.
[24] W. M. P. van der Aalst, M. Pesic, H. Schonenberg, Declarative workflows: Balancing between flexibility and support, Comput. Sci. Res. Dev. 23 (2009) 99–113. doi:10.1007/S00450-009-0057-9.
[25] S. Cerrito, M. C. Mayer, Bounded model search in linear temporal logic and its application to planning, in: TABLEAUX, volume 1397 of *Lecture Notes in Computer Science*, Springer, 1998, pp. 124–140. doi:10.1007/3-540-69778-0_18.
[26] G. D. Giacomo, M. Y. Vardi, Automata-theoretic approach to planning for temporally extended goals, in: ECP, volume 1809 of *Lecture Notes in Computer Science*, Springer, 1999, pp. 226–238. doi:10.1007/10720246_18.
[27] V. Gutiérrez-Basulto, J. C. Jung, A. Ozaki, On metric temporal description logics, in: ECAI, volume 285 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2016, pp. 837–845. doi:10.3233/978-1-61499-672-9-837.
[28] V. Gutiérrez-Basulto, J. C. Jung, T. Schneider, Lightweight temporal description logics with rigid roles and restricted TBoxes, in: IJCAI, AAAI Press, 2015, pp. 3015–3021.
[29] D. Klumpp, J. S. Ribeiro, Walking the Tightrope between Expressiveness and Uncomputability: AGM Contraction beyond the Finitary Realm (Extended Version), Technical Report, 2024. URL: https://dominik-klumpp.net/publications/nmr24/extended.pdf.
[30] F. Patrizi, N. Lipovetzky, G. D. Giacomo, H. Geffner, Computing infinite plans for LTL goals using a classical planner, in: IJCAI, IJCAI/AAAI, 2011, pp. 2003–2008. doi:10.5591/978-1-57735-516-8/IJCAI11-334.
[31] E. L. Fermé, S. O. Hansson, Belief Change - Introduction and Overview, Springer Briefs in Intelligent Systems, Springer, 2018. doi:10.1007/978-3-319-60535-7.
[32] J. S. Ribeiro, A. Nayak, R. Wassermann, Belief change and non-monotonic reasoning sans compactness, in: AAAI, AAAI Press, 2019, pp. 3019–3026. doi:10.1609/AAAI.V33I01.33013019.
[33] S. O. Hansson, Descriptor Revision: Belief Change Through Direct Choice, volume 46, Springer, 2017.
[34] S. O. Hansson, Finite contractions on infinite belief sets, Stud Logica 100 (2012) 907–920. doi:10.1007/S11225-012-9440-9.
[35] B. Nebel, Reasoning and Revision in Hybrid Representation Systems, volume 422 of *Lecture Notes in Computer Science*, Springer, 1990. doi:10.1007/BFB0016445.
[36] M. Dalal, Investigations into a theory of knowledge base revision, in: AAAI, AAAI Press / The MIT Press, 1988, pp. 475–479.
[37] S. E. Dixon, Belief revision: A computational approach, Ph.D. thesis, University of Sydney, 1994.
[38] H. van Ditmarsch, W. van Der Hoek, B. Kooi, Dynamic epistemic logic, volume 337, Springer Science & Business Media, 2007.
[39] A. Baltag, L. S. Moss, S. Solecki, The logic of public announcements and common knowledge and private suspicions, in: TARK, Morgan Kaufmann, 1998, pp. 43–56.
[40] P. Wolper, Temporal logic can be more expressive, Inf. Control. 56 (1983) 72–99. doi:10.1016/S0019-9958(83)80051-5.
[41] A. Darwiche, J. Pearl, On the logic of iterated belief revision, Artif. Intell. 89 (1997) 1–29. doi:10.1016/S0004-3702(96)00038-0.
[42] H. Katsuno, A. O. Mendelzon, On the difference between updating a knowledge base and revising it, in: P. Gärdenfors (Ed.), Belief revision, 2003, pp. 183–203.
[43] S. O. Hansson, Changes of disjunctively closed bases, J. Log. Lang. Inf. 2 (1993) 255–284. doi:10.1007/BF01181682.
[44] S. O. Hansson, Belief contraction without recovery, Stud Logica 50 (1991) 251–260. doi:10.1007/BF00370186.
[45] P. T. Guerra, R. Wassermann, Two AGM-style characterizations of model repair, in: KR, AAAI Press, 2018, pp. 645–646.
[46] O. Kupferman, M. Y. Vardi, P. Wolper, An automata-theoretic approach to branching-time model checking, J. ACM 47 (2000) 312–360. doi:10.1145/333979.333987.
[47] J. Hladik, R. Peñaloza, PSPACE automata for description logics, in: Description Logics, volume 189 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2006.