Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Effective AGM Belief Contraction: A Journey beyond the Finitary Realm

Dominik Klumpp[1,2] , **Jandson S. Ribeiro**[3]
[1]University of Freiburg, Germany
[2]LIX - CNRS - École Polytechnique, France
[3]Cardiff University, United Kingdom
klumpp@lix.polytechnique.fr, ribeiroj@cardiff.ac.uk

Abstract

Despite significant efforts towards 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, we also show that the current *de facto* standard strategies to control computability, which rely on restricting the space of epistemic states, fail: uncomputability remains in all non-finitary cases. Motivated by this disruptive result, we propose new approaches to controlling computability beyond the finitary realm. Using Linear Temporal Logic (LTL) as a case study, we identify an infinite class of fully-rational AGM contraction functions that are computable by design. We use Büchi automata to construct such functions, and to represent and reason about LTL beliefs.

1 Introduction

Evolving a knowledge base is a crucial problem that has been intensively investigated in several research areas such as in *ontology evolution*, *ontology repair*, *data integration*, and *inconsistency handling*. The field of *belief change* (Alchourrón, Gärdenfors, and Makinson, 1985; Gärdenfors, 1988) investigates this problem from the lense of minimal change: removal of information must be minimised, so most of the original beliefs are preserved. The area is founded on the AGM paradigm (Alchourrón, Gärdenfors, and Makinson, 1985), which prescribes rationality postulates of minimal change and defines classes of operations that abide by such postulates. The removal of obsolete information is investigated under the name of *contraction*. Contraction is central, as it underpins most of other kinds of operations and is the core for understanding minimal change. For example, to accommodate a new piece of information α, one must first remove the potential conflicts with α and then incorporate α. The key aspect here is the removal of conflicting information, that is, contraction. Minimal change can, therefore, be understood from the lense of contraction itself. In this paper, we investigate the *computational aspects* of contraction in non-classical logics.

Although originally developed for classical logics, such as classical propositional logic and first order logic, significant efforts have been expended to extend AGM to more expressive non-classical logics used in knowledge representation and reasoning, such as *Horn logics* (Delgrande and Peppas, 2015; Delgrande and Wassermann, 2010; Booth et al., 2014), *para-consistent logics* (da Costa and Bueno, 1998), *description logics* (Ribeiro and Wassermann, 2009; Ribeiro et al., 2013; Flouris, 2006), and *non-compact logics* (Ribeiro, Nayak, and Wassermann, 2018).

Despite all these efforts, computational aspects of AGM belief change have received little attention. The few works on this topic are confined to classical propositional logics and the sub-classical case of Horn logics (Nebel, 1998; Eiter and Gottlob, 1992; Schwind et al., 2020). As the majority of the logics in knowledge representation are non-classical, for belief change to be properly handled, it is paramount that its computational aspects are investigated in such logics. In this paper, we consider a central question:

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

This question is trivially answered in the affirmative for the classical finitary case, that is, when the underlying logic can only distinguish finitely many equivalence classes of formulae, as is the case of classical propositional logic and propositional Horn logic. For the non-finitary case, however, this question is much harder to answer. We provide, in this paper, a severe and disruptive answer: *AGM contraction suffers from uncomputability, in all non-finitary logics.*

The *de facto* standard strategy to control computability rests on limiting "what can be expressed", that is, limiting the space of epistemic states, in favour of tractability. For instance, families of description logics (Baader et al., 2017) have been constructed by depriving the object language of the logic of certain connectives, in favor of taming time and space complexity of some reasoning problems.

We show that, for AGM contraction, uncomputability is inherent to non-finitary logics and therefore, this strategy of limiting epistemic states has no effect in securing computability. This highlights the need for a shift in perspective towards handling computability, which entails devising a *novel machinery to attain computability within AGM*. For this, it is paramount to identify how, and under which conditions, one can construct families of computable AGM contraction functions. Towards this direction, we examine *Lin-*

ear Temporal Logic (Pnueli, 1977), LTL for short. 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 (Clarke et al., 2018), in business process models such as DECLARE (van der Aalst, Pesic, and Schonenberg, 2009), in planning and reasoning about actions (Cerrito and Mayer, 1998; Giacomo and Vardi, 1999), and extending Description Logics with temporal knowledge (Gutiérrez-Basulto, Jung, and Ozaki, 2016; Gutiérrez-Basulto, Jung, and Schneider, 2015). We devise a novel machinery for accommodating computability of AGM contraction in LTL. We explore *Büchi automata* (Richard Büchi, 1966) as a structure to support knowledge representation and reasoning in LTL, and construct contraction operators upon such automata. Our results pave the way for achieving computability of AGM in more general logics used in knowledge representation. In particular for LTL, this opens the door to practical applications, for instance in the repair of unrealizable specifications or the repair of incorrect systems.

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 based on Büchi automata. In Section 6, we establish our second negative result, for all compendious logics: uncomputability of contraction is inevitable in the non-finitary case. Towards attaining computability, in Section 7, we identify a large class of computable contraction functions on LTL theories represented via Büchi automata. Computability stems from the fact that the underlying epistemic preference relations are represented as a special kind of automata: *Büchi-Mealy automata*. Section 8 discusses the impact of our results and provides an outlook on future work.

Detailed proofs of our results can be found in the corresponding technical report (Klumpp and Ribeiro, 2025).

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, $\mathcal{K} = \bigcap\{\, \mathcal{K}' \in CCT_{\mathbb{L}} \mid \mathcal{K} \subseteq \mathcal{K}' \,\}$.

We omit subscripts whenever the meaning is clear. Given a binary relation $<$ on some domain D, the maximal elements of a set $X \subseteq D$ w.rt. the relation $<$ are given by

$$\max_<(X) := \{\, x \in X \mid \text{there is no } y \in X \text{ s.t. } x < y \,\}.$$

2.1 Linear Temporal Logic

We recall the definition of *linear temporal logic* (Pnueli, 1977), LTL for short. For the remainder of the paper, we fix a 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 $\pi = a_0 a_1 \cdots$, where each $a_i \in \mathcal{P}(AP)$ is the 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 (Clarke et al., 2018), which describe possible traces.

Definition 2 (Kripke Structure)**.** *A* Kripke structure *is a tuple $M = (S, I, T, \lambda)$ where 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 $\pi = \lambda(s_0)\lambda(s_1)\lambda(s_2)\cdots$ 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 of a Kripke structure M is given by $Traces(M)$. Figure 1 shows an example of a Kripke structure, in graphical notation.

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

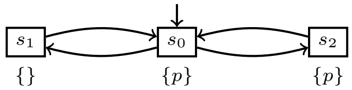


Figure 1: A Kripke structure on $AP = \{p\}$, with an initial state s_0. The labels $\lambda(s_i)$ are shown below each state s_i.

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

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

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 $\varphi \in X$.

Example 4. *Let the atomic proposition* p *denote* "Mauricio swims"*, and let each time step represent one day. The LTL formula* $\mathbf{G}\,\mathbf{F}\,p$ *means* "Mauricio swims infinitely often" *(it always holds that he eventually swims again), and is satisfied by the Kripke structure in Fig. 1. Conversely, the LTL formula* $\mathbf{G}\,p$ *means* "Mauricio swims every day"*. This formula is* not *satisfied by the Kripke structure in Fig. 1.*

We return to this example throughout the paper.

The consequence operator Cn_{LTL} is defined from the satisfaction relation.

Definition 5 (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 6. *LTL is Tarskian and Boolean.*

2.2 Büchi Automata

Büchi automata are finite automata widely used in formal specification and verification of systems, especially in LTL model checking (Clarke et al., 2018). Büchi automata have also been used in planning to synthesise plans when goals are in LTL (Giacomo and Vardi, 1999; Patrizi et al., 2011).

Definition 7 (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 $R \subseteq Q$*.*

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. Figure 2 shows an example of a Büchi automaton.

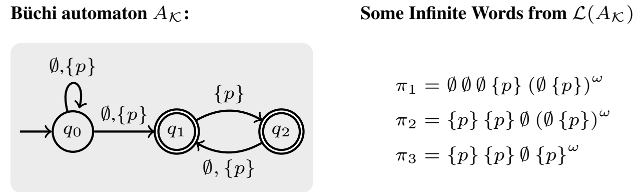


Figure 2: A Büchi automaton $A_\mathcal{K}$ over the alphabet $\Sigma = \{\emptyset, \{p\}\}$. Double circles indicate recurrence states. The initial state q_0 is marked by an incoming arrow. On the right, some infinite words accepted by $A_\mathcal{K}$. By contrast, the word $\emptyset^\omega$ is not accepted.

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.

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 (Richard Büchi, 1966). In the remainder of the paper, we specifically use the construction for union, and denote it with the symbol $\sqcup$. 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 automata-theoretic treatment of LTL is based on the following result:

Proposition 8 (Clarke et al. (2018))**.** *For each LTL formula* φ *and 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*, i.e.,* $\mathcal{L}(A_\varphi) = \{\, \pi \in \mathcal{P}(AP)^\omega \mid \pi \models \varphi \,\}$*, and* $\mathcal{L}(A_M) = \mathit{Traces}(M)$*.*

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 (Gärdenfors, 1988):

$(\mathbf{K_1^-})$ $\mathcal{K} \dot{-} \varphi = Cn(\mathcal{K} \dot{-} \varphi)$ (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 (Alchourrón, Gärdenfors, and Makinson, 1985; Gärdenfors, 1988; Hansson, 1999). A contraction function that satisfies $(\mathbf{K_1^-})$ to $(\mathbf{K_6^-})$ 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 on classical logics (Hansson, 1999). These contraction functions, however, are not suitable for non-classical logics (Flouris, 2006). To embrace more expressive logics, Ribeiro, Nayak, and Wassermann (2018) 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 rationality) and the Blade Contraction Functions (for full rationality). We briefly review these functions.

Definition 9 (Choice Functions)**.** *A* choice function *is a map $\delta : Fm \to \mathcal{P}(CCT)$ taking 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. Choice functions must be syntax-insensitive **(CF3)**.

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.

Theorem 11. *(Ribeiro, Nayak, and Wassermann, 2018) A contraction function $\dot{-}$ is rational iff it is an ECF.*

For full rationality, the choice function must be based on an epistemic preference relation $< \subseteq CCT \times CCT$ on the CCTs. Intuitively, $C < C'$ means that C' is at least as plausible as C. The choice function $\delta_<$ picks the most reliable CCTs w.r.t. the preference relation: $\delta_<(\varphi) = \max_<(\overline{\omega}(\varphi))$. Satisfaction of the postulates $(\mathbf{K_7^-})$ and $(\mathbf{K_8^-})$ depends on two conditions upon the preference relation:

(Maximal Cut): $\max_<(\overline{\omega}(\varphi)) \neq \emptyset$, if φ is not a tautology;

(Mirroring) if $C_1 \not< C_2$ and $C_2 \not< C_1$ but $C_1 < c_3$ then $C_2 < C_3$

The condition **(Maximal Cut)** guarantees that for every non-tautological formula, at least one CCT will be chosen for the contraction, ensuring success. As for **(Mirroring)**, it imposes that every pair of uncomparable CCTs, C_1 and C_2, must mimic each other's preferences, that is, a CCT C_3 that is at least as preferable as C_1 must be at least as preferable as C_2. See (Ribeiro, Nayak, and Wassermann, 2018) for a deep discussion on this property. An ECF whose choice function is based on a binary relation satisfying **(Maximal Cut)** and **(Mirroring)** is called a Blade Contraction Function. They are characterised by all rationality postulates.

Theorem 12. *(Ribeiro, Nayak, and Wassermann, 2018) A contraction function is fully rational iff it is a Blade Contraction Function.*

4 Finite Representation and its Limits

In the AGM paradigm, the epistemic states of an agent are represented as theories which are in general infinite. However, according to Hansson (2012, 2017), 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. We introduce a general notion of finite representation, and show that in non-finitary logics, there is no method of finite representation that captures all epistemic states.

Different strategies of finite representation have been used such as (i) finite bases (Nebel, 1990; Dalal, 1988; Dixon, 1994), and (ii) finite sets of models (van Ditmarsch, van Der Hoek, and Kooi, 2007; Baltag, Moss, and Solecki, 1998). 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 *"Mauricio swims every two days"* cannot be expressed via a finite base in LTL (Wolper, 1983), although it can be expressed via a single Kripke structure (shown in Fig. 1, where p again stands for *"Mauricio swims"*, as in Example 4). On the other hand, *"Mauricio will swim eventually"* is expressible as a single LTL formula ($\mathbf{F}\, p$), but 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 $c_1 :=$ `{a, b}` and $c_2 :=$ `{a, a→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 13 (Encoding)**.** *An* encoding (Σ_{C}, f) *comprises a finite alphabet Σ_{C} and a partial function $f : \Sigma_{\mathrm{C}}^* \rightharpoonup \mathsf{Th}_{\mathbb{L}}$.*

Given an encoding (Σ_C, f), a word $w \in \Sigma_C^*$ 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 *"Mauricio 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 14 (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 $X = Y$.*

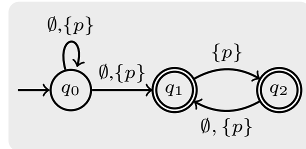
Compendiousness amounts to expressivity in multiple dimensions. Compendious logics can express infinitely many distinct sentences (non-finitary), distinguish between a sentence being true or false (classical negation), and express uncertainty of two or more sentences (disjunction). The property **(Discerning)** 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 15. *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 16. *The logics LTL, CTL, CTL*, μ-calculus and* monadic second-order logic *(MSO) are compendious.*

Büchi automaton $A_{\mathcal{K}}$:



Supported Formulae:

$\mathbf{F}\, p \in \mathcal{S}(A_{\mathcal{K}})$

$\mathbf{G}\,\mathbf{F}\, p \in \mathcal{S}(A_{\mathcal{K}})$

$\mathbf{F}\,\mathbf{G}\,(p \rightarrow \mathbf{X}\, p \lor \mathbf{X}^2\, p) \in \mathcal{S}(A_{\mathcal{K}})$

$\mathbf{G}\, p, \neg(\mathbf{G}\, p) \notin \mathcal{S}(A_{\mathcal{K}})$

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

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

Theorem 17. *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 can 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 18 (Finite Representation)**.** *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 crucially determines the tasks that an agent is able to perform. The encoding must be expressive enough to capture a non-trivial space of epistemic states. We present a suitable encoding for epistemic states over LTL and show that it is strictly more expressive than traditional strategies.

LTL is commonly used in model checking and planning. In both these domains, the primary approach to reason about LTL is based on Büchi automata. Thus, Büchi automata are predestined to be the basis for an encoding of epistemic states over LTL. We define the set of LTL formulae represented by a Büchi automaton as follows:

Definition 19 (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 20 (continued from Example 4)**.** *Figure 3 shows a Büchi automaton (on the left), along with three supported formulae (on the right):* "Mauricio will swim eventually", "Mauricio swims infinitely often", *and the more convoluted belief that* "from some point on, if Mauricio swims on a given day, he will also swim the next day or the day after that". *All accepted traces (i.e., for which a run exists that cycles between states q_1 and q_2) satisfy these formulae.*

The formula $\mathbf{G}\, p$ ("Mauricio swims every day"*) is not supported. While the accepted trace $\{p\}^\omega$ satisfies this formula, other accepted traces, 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. We observe an intriguing property of Büchi automata: 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 $^\omega$ 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 formalised by the function $Th_{UP} : UP \to CCT_{LTL}$ such that $Th_{UP}(\pi) = \{\varphi \in Fm_{LTL} \mid \pi \models \varphi\}$.

Lemma 21. *The function Th_{UP} is a bijection.*

We combine Lemma 21 with two classical observations (Clarke et al., 2018): (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 22. *The support of a Büchi automaton A satisfies*

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

Theorem 23. *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* $\mathbb{E}_{\text{Büchi}}$. The Büchi excerpt is strictly more expressive than the classical strategies of finite representation discussed in Section 4:

Theorem 24. *Let $\mathbb{E}_{\text{base}}$ and $\mathbb{E}_{\text{models}}$ denote respectively the excerpts of finite bases and finite sets of models. 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. □

6 The Impossibility of Effective Contraction

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 and its computability aspects. 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 the new epistemic state must be expressible in the excerpt. A solution is to require the excerpt to contain at least one rational outcome for each possible contraction. We say that a contraction $\dot{-}$ *remains in* $\mathbb{E}$ if $\mathsf{img}(\dot{-}) \subseteq \mathbb{E}$.

Definition 25 (Accommodation)**.** *An excerpt* $\mathbb{E}$ accommodates (fully) rational contraction *if 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 (Flouris, 2006). 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. Surprisingly, rational accommodation and fully rational accommodation coincide.

Proposition 26. *An excerpt $\mathbb{E}$ accommodates rational contraction iff $\mathbb{E}$ accommodates fully rational 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 27 (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 $w \in \Sigma_{\text{C}}^*$ 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 28 (Non-Finitary)**.** *A theory $\mathcal{K}$ is* non-finitary *if $\mathcal{K}$ 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 29 (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.*

Example 30. *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 $(x \neq \mathsf{n}) \vee (x \neq \mathsf{m})$, equivalently written as $\neg(x = \mathsf{n} \wedge x = \mathsf{m})$, is a tautology (for constants $\mathsf{n} \neq \mathsf{m}$).*

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 $\varphi \wedge \neg\varphi$).

Lemma 31. *Every non-finitary theory contains a cleaving.*

Example 32. *Returning to our swimming example for LTL, consider the following statement:*

> If Mauricio will swim in n days from today, he will swim on at least two days (overall).

This can be written as the LTL formula ψ_n with

$$\psi_n :\equiv (\mathbf{X}^n\, p) \rightarrow \mathit{twice}(p) \ ,$$

where the LTL formula $\mathit{twice}(p) :\equiv \mathbf{F}\,(p \wedge \mathbf{X}\, \mathbf{F}\, p)$ expresses that Mauricio swims on at least two days. The set of formulae $\{\, \psi_n \mid n \in \mathbb{N} \,\}$ is a cleaving in the theory $\mathcal{S}(A_{\mathcal{K}})$ supported by the Büchi automaton in Fig. 3:

- *Each formula ψ_n is in the theory. As shown in Fig. 3, the formula $\mathbf{G}\,\mathbf{F}\, p$ (*"Mauricio swims infinitely often"*) is in the theory, and it implies (the conclusion of) each ψ_n.*
- *Whenever $n \neq m$, the formulae ψ_n and ψ_m are not equivalent* **(CL1)**.
- *Whenever $n \neq m$, the disjunction $\psi_n \vee \psi_m$ is equivalent to $(\mathbf{X}^n\, p) \wedge (\mathbf{X}^m\, p) \rightarrow \mathit{twice}(p)$, a tautology* **(CL2)***: if Mauricio swims in n days and in m days, he clearly swims on at least two days.*

Given a contraction that remains in an excerpt, cleavings provide a way of generating many contractions that also remain in 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 33 (Composition). *Let δ be a choice function on a theory $\mathcal{K}$, let $\mathcal{C} \subseteq \mathcal{K}$ be a cleaving, and let $\pi : \mathbb{N} \rightarrow \mathcal{C}$ be a permutation of $\mathcal{C}$. The* composition of δ and π *is the function $\delta_\pi : Fm \rightarrow \mathcal{P}(CCT)$ such that*

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

where $\min_\pi(\varphi) = \pi(i)$, for the least $i \in \mathbb{N}$ such that $\varphi \vee \pi(i)$ is non-tautological, or $\min_\pi(\varphi) = \bot$ if no such i exists.

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

Lemma 34. *The composition δ_π of a choice function δ and a permutation π of a cleaving $\mathcal{C} \subseteq \mathcal{K}$ is a choice function.*

Each composition generates a new choice function, which in turn induces a rational contraction function that remains in the excerpt.

Example 35 (continued from Example 32). *Suppose we contract $\varphi \equiv p$ (*"Mauricio swims today"*), and we have $\pi(n) = \psi_n$ for all n. We have $\min_\pi(p) = \psi_1$, as the formula $p \vee \psi_0$ is a tautology, whereas $\varphi \vee \psi_1 \equiv \psi_1$ (*"if Mauricio swims tomorrow, he swims on at least two days"*), which is non-tautological. It follows that $\mathcal{K} \dot{-}_{\delta_\pi} \varphi = \mathcal{K} \dot{-}_{\delta} (\varphi \vee \psi_1)$. We contract* "Mauricio swims today" *with $\dot{-}_{\delta_\pi}$ in the same way as we contract* "if Mauricio swims tomorrow, he swims on at least two days" *with $\dot{-}_{\delta}$.*

Yet, the contraction functions induced by compositions are not necessarily computable.

Theorem 36. *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. $\square$

Theorem 36 makes evident that uncomputability of AGM contraction is inevitable. In fact, there are uncountably many uncomputable contraction functions. Attempting to avoid this uncomputability by restraining the expressiveness of the excerpt leaves only the most trivial case: finitary theories.

7 Effective Contraction in the Büchi Excerpt

Despite the strong negative result of Section 6, computability can still be harnessed in 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, and we present classes of computable contraction functions.

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

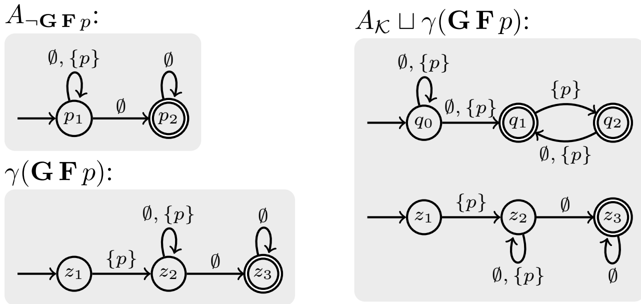


Figure 4: BCF contraction of $\mathbf{G}\,\mathbf{F}\,p$ from $\mathcal{S}(A_{\mathcal{K}})$.

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 21), and the support of a Büchi automaton is determined by the CCTs corresponding to its accepted ultimately periodic traces (Lemma 22), 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.

Definition 37 (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)**. Each Büchi choice function induces a rational contraction function.

Definition 38 (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*

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

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 $\mathcal{K} \cap \mathcal{S}(\gamma(\varphi)) = \mathcal{S}(A \sqcup \gamma(\varphi))$, where $\sqcup$ denotes the union of Büchi automata (cf. Section 2). The class of all rational contraction functions that remain in the Büchi excerpt corresponds exactly to the class of all BCFs.

Theorem 39. *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 40. *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 $p \wedge \neg\mathbf{G}\,\mathbf{F}\,p$. In our swimming example (cf. Example 4), this corresponds to* "Mauricio swims today, but does not swim infinitely often."

The result of the contraction is the belief state $\mathcal{S}(A_{\mathcal{K}} \sqcup \gamma(\mathbf{G}\,\mathbf{F}\,p))$, 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. The other supported formulae listed in Fig. 3 are still supported (see Example 20 for a discussion of their meaning).

As BCFs capture all rational contractions within the excerpt, it follows from Theorem 36 that not all BCFs are computable. Note from Definition 38 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 $\gamma(\varphi)$. Conditions (i) and (ii) can be realised with standard reasoning methods for LTL and Büchi automata (Clarke et al., 2018). For condition (iv), we observe above 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 characterise all computable contraction functions within the Büchi excerpt.

Theorem 41. *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 .*

In the following, we define a large class of computable Büchi choice functions. As outlined in Section 3, a choice function is an extra-logical mechanism that realises the epistemic preferences of an agent, which can be formalised as a preference relation on CCTs. Due to the tight connection between CCTs and ultimately periodic traces, we can equivalently formalise the epistemic preferences as a relation on ultimately periodic traces. To attain computability, we finitely represent such a (potentially infinite) relation on traces using a special kind of Büchi automata:

Definition 42 (Büchi-Mealy Automata)**.** *A* Büchi-Mealy automaton *is a Büchi automaton on $\Sigma_{\text{BM}} = \mathcal{P}(AP) \times \mathcal{P}(AP)$.*

A Büchi-Mealy automaton B accepts infinite sequences of pairs $(a_1, b_1)(a_2, b_2) \cdots (a_i, b_i) \cdots$ with $a_i, b_i \in \mathcal{P}(AP)$, for all $i \geq 1$. Such an infinite sequence corresponds to a pair of traces (π_1, π_2) where $\pi_1 = a_1 a_2 \cdots a_i \cdots$ and $\pi_2 = b_1 b_2 \cdots b_i \cdots$. Therefore, a Büchi-Mealy automaton B recognises the binary relation

$$\mathcal{R}(B) := \{(a_1 \cdots, b_1 \cdots) \mid (a_1, b_1)(a_2, b_2) \cdots \in \mathcal{L}(B)\}$$

If $(\pi_1, \pi_2) \in \mathcal{R}(B)$ then π_2 is at least as plausible as π_1.

Example 43. *Consider again the swimming example (cf. Example 4), and an epistemic preference that deems scenarios in which Mauricio swims later to be less plausible than those where he swims sooner. This preference is expressed by the Büchi-Mealy automaton B shown in Fig. 5 (on the left). The automaton B recognises the relation*

$$\mathcal{R}(B) = \{\, (\pi, \pi') \in \Sigma^\omega \times \Sigma^\omega \mid \mathit{first}_p(\pi) > \mathit{first}_p(\pi') \,\}$$

where $\mathit{first}_p(\cdot)$ is the index of the first occurrence of proposition p in the given trace (and ∞ if p never occurs). In

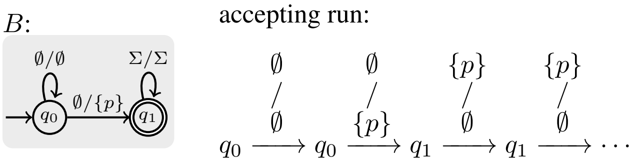


Figure 5: A Büchi-Mealy automaton B on $AP = \{p\}$. By convention, we write a/b rather than (a, b). A label containing Σ denotes transitions for both $\emptyset$ and $\{p\}$. On the right, an accepting run of B.

other words, the earlier p occurs in a trace, the more plausible is such a trace. For instance, the accepting run on the right-hand side of Fig. 5 is the reason that the trace $\emptyset\emptyset\{p\}^\omega$ (where Mauricio swims only in two days) is considered less plausible than $\emptyset\{p\}\emptyset^\omega$ (where Mauricio already swims tomorrow), and hence the pair $(\emptyset\emptyset\{p\}^\omega, \emptyset\{p\}\emptyset^\omega)$ is in $\mathcal{R}(B)$.

An epistemic preference relation induces a choice function which always selects the *maximal*, i.e., the most plausible CCTs that do not contain the given formula. In order to analogously define the Büchi choice function induced by a Büchi-Mealy automaton, we show that the set of most plausible CCTs can be represented by a Büchi automaton.

Lemma 44. *Let B be a Büchi-Mealy automaton, and φ an LTL formula. There exists a Büchi automaton $A^{B,\varphi}_{\max}$ such that $\mathcal{L}(A^{B,\varphi}_{\max}) = \max_{\mathcal{R}(B)}\{\, \pi \in \Sigma^\omega \mid \pi \models \varphi \,\}$.*

The Büchi choice function induced by a Büchi-Mealy automaton B is the function γ_B with $\gamma_B(\varphi) = A^{B,\neg\varphi}_{\max}$, if φ is non-tautological, and $\gamma_B(\varphi) = A_\varphi$ otherwise. The automaton $A^{B,\neg\varphi}_{\max}$ can be constructed from B and φ through a series of effective automata constructions, as detailed in the proof of Lemma 44. Consequently, γ_B is computable.

Proposition 45. *If the relation $\mathcal{R}(B)$ recognised by a Büchi-Mealy automaton B satisfies **(Maximal Cut)**, then γ_B is a computable Büchi choice function.*

To obtain fully rational computable contraction functions, it suffices that the relation recognised by the Büchi-Mealy automaton satisfies **(Mirroring)** as well as **(Maximal Cut)**.

Theorem 46. *Let $\mathcal{K}$ be a theory in the Büchi excerpt, and let B be a Büchi-Mealy automaton such that the relation $\mathcal{R}(B)$ satisfies **(Mirroring)** and **(Maximal Cut)**.*
The BCF $\dot{-}_{\mathcal{R}(B)}$ is fully rational and computable.

Example 47 (continued from Example 43)**.** *Consider the Büchi automaton $A_\mathcal{K}$ in Fig. 3, and the epistemic preference expressed by the Büchi-Mealy automaton B in Fig. 5. Note that the relation $\mathcal{R}(B)$ satisfies both **(Maximal Cut)**, as there always exists an earliest-possible occurrence of p, and **(Mirroring)**. To contract the formula $\varphi :\equiv \mathbf{G}\,\mathbf{F}\,p$ (*"Mauricio swims infinitely often"*) from $\mathcal{S}(A_\mathcal{K})$, we construct the automaton $A^{B,\neg\varphi}_{\max}$ representing only the most plausible CCTs. This automaton is equivalent to the automaton $\gamma(\mathbf{G}\,\mathbf{F}\,p)$ shown in Fig. 4. The most preferrable traces wrt. $\mathcal{R}(B)$ are those where p holds already in the first step (*"Mauricio swims today"*). Therefore, the result of the contraction is the same as in Example 40.*

As there exist Büchi-Mealy automata that satisfy **(Mirroring)** and **(Maximal Cut)**, such as the automaton discussed in Examples 43 and 47, we conclude that the Büchi excerpt effectively accommodates fully rational contraction.

8 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, in the presence of classical negation, revision and contraction are interdefinable via Levi and Harper identities (Santos, 2019). Thus, it is likely that revision also suffers from uncomputability. Accordingly, uncomputability might span to iterated belief revision (Darwiche and Pearl, 1997), update and erasure (Katsuno and Mendelzon, 2003), and pseudo-contraction (Hansson, 1993), 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 (Ribeiro et al., 2013). In these logics, the *recovery* postulate $(\mathbf{K_5^-})$ can be replaced by the *relevance* postulate (Hansson, 1991), 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 computable contraction functions for LTL using Büchi automata. This is an initial step towards the application of belief change in other areas, such as methods for automatically repairing systems (Guerra and Wassermann, 2018). 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 (Kupferman, Vardi, and Wolper, 2000; Hladik and Peñaloza, 2006).

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.* 50(2):510–530.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Baltag, A.; Moss, L. S.; and Solecki, S. 1998. The logic of public announcements and common knowledge and private suspicions. In *TARK*, 43–56. Morgan Kaufmann.

Booth, R.; Meyer, T.; Varzinczak, I.; and Wassermann, R. 2014. On the link between partial meet, kernel, and infra contraction and its application to Horn logic. *CoRR* abs/1401.3902.

Cerrito, S., and Mayer, M. C. 1998. Bounded model search in linear temporal logic and its application to planning. In *TABLEAUX*, volume 1397 of *Lecture Notes in Computer Science*, 124–140. Springer.

Clarke, E. M.; Grumberg, O.; Kroening, D.; Peled, D. A.; and Veith, H. 2018. *Model checking, 2nd Edition*. MIT Press.

da Costa, N. C., and Bueno, O. 1998. Belief change and inconsistency. *Logique et Analyse* 41(161/163):31–56.

Dalal, M. 1988. Investigations into a theory of knowledge base revision. In *AAAI*, 475–479. AAAI Press / The MIT Press.

Darwiche, A., and Pearl, J. 1997. On the logic of iterated belief revision. *Artif. Intell.* 89(1-2):1–29.

Delgrande, J. P., and Peppas, P. 2015. Belief revision in Horn theories. *Artif. Intell.* 218:1–22.

Delgrande, J. P., and Wassermann, R. 2010. Horn clause contraction functions: Belief set and belief base approaches. In *KR*. AAAI Press.

Dixon, S. E. 1994. *Belief revision: A computational approach*. Ph.D. Dissertation, University of Sydney.

Eiter, T., and Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artif. Intell.* 57(2-3):227–270.

Flouris, G. 2006. *On Belief Change and Ontology Evolution*. Ph.D. Dissertation, University of Crete.

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

Giacomo, G. D., and Vardi, M. Y. 1999. Automata-theoretic approach to planning for temporally extended goals. In *ECP*, volume 1809 of *Lecture Notes in Computer Science*, 226–238. Springer.

Guerra, P. T., and Wassermann, R. 2018. Two AGM-style characterizations of model repair. In *KR*, 645–646. AAAI Press.

Gutiérrez-Basulto, V.; Jung, J. C.; and Ozaki, A. 2016. On metric temporal description logics. In *ECAI*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, 837–845. IOS Press.

Gutiérrez-Basulto, V.; Jung, J. C.; and Schneider, T. 2015. Lightweight temporal description logics with rigid roles and restricted TBoxes. In *IJCAI*, 3015–3021. AAAI Press.

Hansson, S. O. 1991. Belief contraction without recovery. *Stud Logica* 50(2):251–260.

Hansson, S. O. 1993. Changes of disjunctively closed bases. *J. Log. Lang. Inf.* 2(4):255–284.

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

Hansson, S. O. 2012. Finite contractions on infinite belief sets. *Stud Logica* 100(5):907–920.

Hansson, S. O. 2017. *Descriptor Revision: Belief Change Through Direct Choice*, volume 46. Springer.

Hladik, J., and Peñaloza, R. 2006. PSPACE automata for description logics. In *Description Logics*, volume 189 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Katsuno, H., and Mendelzon, A. O. 2003. On the difference between updating a knowledge base and revising it. In Gärdenfors, P., ed., *Belief revision*. 183–203.

Klumpp, D., and Ribeiro, J. S. 2025. Effective AGM belief contraction: A journey beyond the finitary realm (technical report). *CoRR* abs/2409.09171.

Kupferman, O.; Vardi, M. Y.; and Wolper, P. 2000. An automata-theoretic approach to branching-time model checking. *J. ACM* 47(2):312–360.

Nebel, B. 1990. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Computer Science*. Springer.

Nebel, B. 1998. How hard is it to revise a belief base? In Dubois, D., and Prade, H., eds., *Belief Change*. Springer Netherlands. 77–145.

Patrizi, F.; Lipovetzky, N.; Giacomo, G. D.; and Geffner, H. 2011. Computing infinite plans for LTL goals using a classical planner. In *IJCAI*, 2003–2008. IJCAI/AAAI.

Pnueli, A. 1977. The temporal logic of programs. In *FOCS*, 46–57. IEEE Computer Society.

Ribeiro, M. M., and Wassermann, R. 2009. Base revision for ontology debugging. *J. Log. Comput.* 19(5):721–743.

Ribeiro, M. M.; Wassermann, R.; Flouris, G.; and Antoniou, G. 2013. Minimal change: Relevance and recovery revisited. *Artif. Intell.* 201:59–80.

Ribeiro, J. S.; Nayak, A.; and Wassermann, R. 2018. Towards belief contraction without compactness. In *KR*, 287–296. AAAI Press.

Richard Büchi, J. 1966. *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. 1–11.

Santos, J. S. R. 2019. *Belief Change without Compactness*. Ph.D. Dissertation, Macquarie University.

Schwind, N.; Konieczny, S.; Lagniez, J.; and Marquis, P. 2020. On computational aspects of iterated belief change. In *IJCAI*, 1770–1776. ijcai.org.

van der Aalst, W. M. P.; Pesic, M.; and Schonenberg, H. 2009. Declarative workflows: Balancing between flexibility and support. *Comput. Sci. Res. Dev.* 23(2):99–113.

van Ditmarsch, H.; van Der Hoek, W.; and Kooi, B. 2007. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media.

Wolper, P. 1983. Temporal logic can be more expressive. *Inf. Control.* 56(1/2):72–99.