# Revisiting Ability-Based Bisimulation

**Carlos Areces**[1,2,3] , **Raul Fervari**[1,2] , **Antonio Mondejar**[1]
[1]Universidad Nacional de Córdoba, FAMAF, Argentina
[2]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
[3]Guangdong Technion – Israel Institute of Technology (GTIIT), China
{carlos.areces, rfervari}@unc.edu.ar, antonio.mondejar@mi.unc.edu.ar

**Abstract**

Bisimulation is a crucial tool for investigating and understanding the semantic properties of labeled transition systems (LTSs) and relational models in general. In particular, it plays a fundamental role in characterizing model equivalence with respect to a given logical language and in guiding the construction of minimal models. In this paper, we study bisimulation in the context of a logic for expressing knowing-how assertions, which are related to an agent's ability to achieve a given goal. We begin by revisiting an existing notion of bisimulation for this logic and reformulating it using purely semantic clauses. We then establish adequacy results for this new notion. Next, we provide a computational analysis of the problem of checking whether two models are bisimilar. In particular, we show that this problem is PSpace-complete. We also investigate two approaches to model minimization in this setting, each exhibiting different computational properties. Along the way, our systematic study of bisimulation yields additional by-product results, w.r.t., for example, the complexity of the definability problem for this logic.

## 1 Introduction

Among the formalisms available for modeling system dynamics, *Labeled Transition Systems* (LTSs) are a widely adopted option. By representing execution steps as action-labeled transitions between states, LTSs offer a flexible framework utilized across diverse disciplines, including process calculi, concurrency theory, and various branches of modal and temporal logic.

In the context of formal verification—specifically model-checking—an LTS serves as the computational substrate upon which logical specifications are evaluated (cf. (Baier and Katoen 2008)). Given that these abstractions often encounter the state-explosion problem, characterizing *behavioral equivalence* is a critical necessity. The objective is to identify a *minimal* representation that preserves the logical integrity of the original system, a task that requires a mathematically robust definition of operational "sameness".

To address this, structural approaches favor *bisimulation* (Milner 1971; Park 1981; Milner 1989; Sangiorgi and Rutten 2012) over simpler, trace-based metrics. Formally, a bisimulation is a binary relation $Z$ that enforces a strict coinductive symmetry between transition systems. A pair $(s, s') \in Z$ must satisfy the following core properties:

1. **Label Consistency:** In state-labeled variants, $s$ and $s'$ must share identical atomic valuations.
2. **Transfer Properties (Zig-Zag):** Any transition $s \xrightarrow{a} t$ in the first system must be matched by a transition $s' \xrightarrow{a} t'$ in the second such that the resulting states remain related $(t, t') \in Z$, and vice-versa.

The standard notion of bisimulation provides a structural characterization of equivalence: two systems are considered identical if their operational capabilities perfectly mimic each other at every branching point. This allows for significant state-space compression while ensuring that the reduced model remains behaviorally indistinguishable from the original. Practical applications of structural characterization and model contraction via bisimulations can be found in very diverse areas, such as planning (see e.g. (Nissim, Hoffmann, and Helmert 2011; Bolander, Burigana, and Montali 2025)), verification of voting protocols (see e.g. (Belardinelli et al. 2017; Belardinelli et al. 2021)), or referring expressions generation in natural language processing (see e.g. (Areces, Figueira, and Gorín 2011)), to name a few.

In the specific context of *Basic Modal Logic* (BML), bisimulation provides the exact structural counterpart to logical equivalence (van Benthem 1976; Blackburn, de Rijke, and Venema 2002). Hennessy-Milner-style results establish that, under appropriate finiteness conditions, two states are bisimilar if and only if they satisfy the same set of BML formulas. Consequently, bisimulation serves as the standard tool for determining when two models are indistinguishable from the perspective of the basic modal language.

However, the choice of an equivalence notion is fundamentally dictated by the expressive power of the language under consideration. If one moves from BML to logics with different expressive power—such as, e.g., *graded modal logics* (Barnaba and Caro 1985)—the standard definition of bisimulation no longer suffices to capture logical indistinguishability. In these cases, similar challenges regarding model minimality and representation arise, but they must be addressed through refined variants of bisimulation (see (de Rijke 2000) for the case of graded modal logic). These variants typically involve additional structural constraints or modified transfer properties specifically engineered to match the distinctive operators of the target language.

In modern modal logic, there exists a vast number of

modal languages with different levels of expressivity, each requiring its own notion of bisimulation. Examples of alternative bisimulations for variants of modal logic can be found in (Figueira, Figueira, and Areces 2015; Abriola et al. 2016; Abriola et al. 2018) for data-aware logics, in (Areces, Fervari, and Hoffmann 2015) for dynamic logics, or in (Belardinelli, Dima, and Murano 2018) for strategy logics. Our focus, in this article, is to investigate the computational properties of bisimulation for the logic of knowing-how, $\mathsf{L_{Kh}}$, based on linear plans (Wang 2015; Wang 2018). Knowing-how is an epistemic concept motivated by scenarios in AI related to strategic abilities and planning. Within this approach, the interpretation of knowing how is *ability-based*: an agent knows how to achieve a goal $\psi$ from an initial condition $\varphi$ if there exists a sequence of actions (a *plan*) that can be executed to completion by the agent from every state satisfying $\varphi$, leading unerringly to states satisfying $\psi$.

The notion of bisimulation for $\mathsf{L_{Kh}}$ was first introduced in previous literature (Fervari, Velázquez-Quesada, and Wang 2022). However, a distinctive feature of these existing notions is the presence of a syntactic clause, stating that for two models to be bisimilar, plans executed at so-called *propositionally definable* sets in one model should be mimicked in the other model. A set is propositionally definable if it coincides with the extension of a propositional formula. This reliance on syntactic objects makes it difficult to provide purely procedural ways to operate over bisimulations. In this article, we recast the "propositional definability" condition into a purely semantic clause that depends simply on the valuation function of the model, and establish its adequacy. We then proceed to investigate different computational properties of this new notion of bisimulation for $\mathsf{L_{Kh}}$.

Our first contribution in this article is to reformulate the definition of bisimulations for $\mathsf{L_{Kh}}$ in terms of structural conditions. We call the new notion $\mathsf{L^*_{Kh}}$-bisimulation and show that $\mathsf{L^*_{Kh}}$-bisimilar models satisfy the same $\mathsf{L_{Kh}}$-formulas. Crucially, we prove that the class of finite-valuation models ($\mathbf{M_{FV}}$) constitutes a Hennessy-Milner class for this notion.

The structural nature of $\mathsf{L^*_{Kh}}$-bisimulation is useful for computational analysis. In this regard, we characterize the complexity of the model comparison problem under $\mathsf{L^*_{Kh}}$-bisimulation. We prove that checking whether two models are $\mathsf{L^*_{Kh}}$-bisimilar is PSpace-complete. For the lower bound, we provide a reduction from the model-checking problem for $\mathsf{L_{Kh}}$ shown to be PSpace-complete in (Demri and Fervari 2023), while the upper bound is obtained by a non-deterministic algorithm running in polynomial space. The algorithm also enables us to build a formula that distinguishes two models if they are not bisimilar.

Finally, we investigate the problem of model minimization and show that models can be contracted while preserving $\mathsf{L_{Kh}}$ properties. First, we study a novel way of defining contractions via $\mathsf{L^*_{Kh}}$-bisimulations. We show that this transformation provides us a form of *offline compilation*, that enables us to perform model-checking in polynomial time, but at the cost of having possibly an exponential number of edges. Moreover, we prove that while BML-bisimulation is not maximal for $\mathsf{L_{Kh}}$, it remains an auto-bisimulation, allowing for the use of efficient, existing partition-refinement algorithms to minimize models.

Our results enable us to revisit results about the expressivity of $\mathsf{L_{Kh}}$, and to understand associated computational properties. In order to do so, we take as a starting point the notion of bisimulation recently introduced in (Areces, Fervari, and Mondejar 2026) for an *uncertainty-based knowing-how logic* (see (Areces et al. 2021; Areces et al. 2025)), in which LTSs are enriched with an indistinguishability relation between plans. For this reason, bisimulations in (Areces, Fervari, and Mondejar 2026) are defined relative to *certain* plans. In this paper, we adapt this notion to take into consideration *every possible* plan, which entails a potentially much more complex behavior. Compared to that work, the problem of deciding whether two models are bisimilar jumps from coNP-completeness to PSpace-completeness, which naturally requires the use of different techniques. This illustrates that none of the previous results trivially transfer to our case. Additionally, we also obtain here a Henessy-Milner result (Thm. 1) that is more general than the ones previously known, and which applies to both semantics.

**Outline.** In Sec. 2 we introduce the syntax and semantics of the knowing-how logic $\mathsf{L_{Kh}}$, together with the existing notion of bisimulation from (Fervari, Velázquez-Quesada, and Wang 2022) and results about model-checking from (Demri and Fervari 2023). In Sec. 3 we reformulate the clauses of bisimulation, show a correspondence with the previously existing definition, and prove adequacy results for finite-valuation LTSs. Moreover, we present results that illustrate the power of our new bisimulations. Sec. 4 is devoted to proving that the problem of deciding if two LTSs are $\mathsf{L^*_{Kh}}$-bisimilar is PSpace-complete, including a way to construct formulas distinguishing non-bisimilar LTSs. Additionally, we discuss the problem of model minimization. Finally, in Sec. 5 we provide some final remarks.

# 2 Preliminaries

## 2.1 Syntax and Semantics

We start by presenting the logical framework introduced in (Wang 2015; Wang 2018) for knowing-how, based on linear plans. In what follows, let Prop be a countable set of propositional symbols.

**Definition 1.** *Formulas of the* ***language*** $\mathsf{L_{Kh}}$ *are defined by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}(\varphi, \varphi),$$

*where* $p \in \mathsf{Prop}$*. Other Boolean connectives are defined as usual. The formula* $\mathsf{Kh}(\varphi, \psi)$ *is read as* "when $\varphi$ is the case, the agent knows how to make $\psi$ true".

In (Wang 2015; Wang 2018), formulas are interpreted over *labeled transition systems* (LTSs), i.e., relational models in which each relation indicates the source and target of a particular action the agent can perform.

**Definition 2** (LTS)**.** *A* ***Labeled Transition System (LTS)*** *over* Prop *is a tuple* $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V} \rangle$ *where* $\mathrm{S} \neq \emptyset$ *is a set of states,* $\mathsf{Act} \neq \emptyset$ *is a countable set of action names,* $\mathrm{R} = \{\mathrm{R}_a \subseteq \mathrm{S} \times \mathrm{S} \mid a \in \mathsf{Act}\}$ *is a collection of binary relations on* S *and* $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathsf{Prop})$ *is a valuation function. We say that* $\mathcal{M}$ *is* ***finite-domain*** *if its set of states* S *is*

*finite, and that $\mathcal{M}$ is **finite** if also* $\mathsf{Act}$ *and each* $\mathrm{V}(s)$ *are finite. We denote by* $\mathbf{M_{FD}} \stackrel{def}{=} \{\mathcal{M} \mid \mathcal{M}$ *is finite-domain*$\}$ *and* $\mathbf{M_{F}} \stackrel{def}{=} \{\mathcal{M} \mid \mathcal{M}$ *is a finite LTS*$\}$*. If* $s \in \mathrm{S}$*, we call* $\mathcal{M}, s$ *a* pointed LTS.

While LTSs are the crucial structures for interpreting $\mathsf{L_{Kh}}$-formulas, knowing-how does not depend only on the execution of basic actions, but on how these actions can be composed in so-called *plans*. We introduce these notions below.

**Definition 3** (Actions and plans)**.** *Let* $\mathsf{Act}^*$ *be the set of finite sequences over a set of actions* $\mathsf{Act}$*. Elements of* $\mathsf{Act}^*$ *are called **plans**, with* $\epsilon$ *being the **empty plan**. Given* $\sigma \in \mathsf{Act}^*$*, we use* $|\sigma|$ *to denote its length (with* $|\epsilon| = 0$*). For* $0 \leq k \leq |\sigma|$*, the plan* $\sigma_k$ *is* $\sigma$*'s initial segment up to (and including) the* $k$*th position (with* $\sigma_0 \stackrel{def}{=} \epsilon$*). For* $0 < k \leq |\sigma|$*, the action* $\sigma[k]$ *is the one in* $\sigma$*'s* $k$*th position.*

We provide some useful notation to manipulate relations.

**Definition 4.** *Given* $\mathrm{R} = \{\mathrm{R}_a \subseteq \mathrm{S} \times \mathrm{S} \mid a \in \mathsf{Act}\}$ *and* $\sigma \in \mathsf{Act}^*$*, define* $\mathrm{R}_\sigma \subseteq \mathrm{S} \times \mathrm{S}$ *inductively as:* $\mathrm{R}_\epsilon \stackrel{def}{=} \{(s,s) \mid s \in \mathrm{S}\}$ *and* $\mathrm{R}_{\sigma a} \stackrel{def}{=} \mathrm{R}_\sigma \circ \mathrm{R}_a$*. For* $T \cup \{t\} \subseteq \mathrm{S}$*, define* $\mathrm{R}_\sigma(t) \stackrel{def}{=} \{u \in \mathrm{S} \mid (t,u) \in \mathrm{R}_\sigma\}$*, and* $\mathrm{R}_\sigma(T) \stackrel{def}{=} \bigcup_{u \in T} \mathrm{R}_\sigma(u)$*.*

In what follows, we introduce the notion of strong executability of plans (Wang 2015), which determines when a given plan is appropriate for achieving a certain goal.

**Definition 5** (Strong executability)**.** *Let* $\mathcal{M}{=}\langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ *be an* LTS*, and let* $\sigma \in \mathsf{Act}^*$ *be a plan. We say that* $\sigma$ *is **strongly executable** (SE) at* $s \in \mathrm{S}$ *if and only if, for all* $k \in [0 \dots |\sigma| - 1]$*,* $t \in \mathrm{R}_{\sigma_k}(s)$ *implies* $\mathrm{R}_{\sigma[k+1]}(t) \neq \emptyset$*. We define the set* $\mathrm{SE}^{\mathcal{M}}(\sigma) \stackrel{def}{=} \{s \in \mathrm{S} \mid \sigma$ *is SE at* $s\}$*.*

In words, *a plan* is strongly executable at a given state when *all its partial executions* from such a state can be completed without aborting. We are now in position to introduce the semantics of the logic.

**Definition 6.** *Let* $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ *be an* LTS *and* $s \in \mathrm{S}$*. The **satisfiability relation** $\models$ between pointed* LTS*s and* $\mathsf{L_{Kh}}$*-formulas is inductively defined as:*

$$
\begin{array}{lcl}
\mathcal{M}, s \models p & \stackrel{def}{\Leftrightarrow} & p \in \mathrm{V}(s) \\
\mathcal{M}, s \models \neg\varphi & \stackrel{def}{\Leftrightarrow} & \mathcal{M}, s \not\models \varphi \\
\mathcal{M}, s \models \varphi \vee \psi & \stackrel{def}{\Leftrightarrow} & \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models \mathsf{Kh}(\varphi, \psi) & \stackrel{def}{\Leftrightarrow} & \text{there is } \sigma \in \mathsf{Act}^* \text{ such that:} \\
 & & \quad (1)\ [\![\varphi]\!]^{\mathcal{M}} \subseteq \mathrm{SE}^{\mathcal{M}}(\sigma), \text{ and} \\
 & & \quad (2)\ \mathrm{R}_\sigma([\![\varphi]\!]^{\mathcal{M}}) \subseteq [\![\psi]\!]^{\mathcal{M}},
\end{array}
$$

*where* $[\![\chi]\!]^{\mathcal{M}} \stackrel{def}{=} \{t \in \mathrm{S} \mid \mathcal{M}, t \models \chi\}$*. Any plan* $\sigma$ *making true the existential statement in the semantic clause of* $\mathsf{Kh}(\varphi, \psi)$ *is called a **witness** for* $\mathsf{Kh}(\varphi, \psi)$*.*

*The universal modality (see (Goranko and Passy 1992)) can be expressed in* $\mathsf{L_{Kh}}$ *as* $\mathsf{A}\varphi \stackrel{def}{=} \mathsf{Kh}(\neg\varphi, \bot)$*, and its dual as* $\mathsf{E}\varphi \stackrel{def}{=} \neg\mathsf{A}\neg\varphi$*. Notice that* $\mathcal{M}, s \models \mathsf{A}\varphi$ *iff* $[\![\varphi]\!]^{\mathcal{M}} = \mathrm{S}$*.*

## 2.2 A Notion of Bisimulation for $\mathsf{L_{Kh}}$

It is a well-known fact that a proper notion of *bisimulation* is fundamental to understand the expressive power of a modal language. For $\mathsf{L_{Kh}}$, bisimulation was first defined in (Fervari, Velázquez-Quesada, and Wang 2022). In this section we will review the definitions introduced therein.

**Definition 7.** *Let* $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ *be an* LTS *over* Prop*. Take* $\sigma \in \mathsf{Act}^*$*,* $X, T \subseteq \mathrm{S}$*. We define:*

- $X \stackrel{\sigma}{\Rightarrow} T \stackrel{def}{\Leftrightarrow} X \subseteq \mathrm{SE}^{\mathcal{M}}(\sigma)$ *and* $\mathrm{R}_\sigma(X) \subseteq T$*,*
- $X \Rightarrow T \stackrel{def}{\Leftrightarrow}$ *there is* $\sigma \in \mathsf{Act}^*$ *such that* $X \stackrel{\sigma}{\Rightarrow} T$*, and*
- $X$ *is propositionally definable in* $\mathcal{M}$ *if and only if there is a propositional formula* $\varphi$ *such that* $X = [\![\varphi]\!]^{\mathcal{M}}$*.*

Now we introduce the notion of bisimulation for $\mathsf{L_{Kh}}$.

**Definition 8** ($\mathsf{L_{Kh}}$-bisimulation)**.** *Let* $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ *and* $\mathcal{M}' = \langle \mathrm{S}', \mathsf{Act}', \mathrm{R}', \mathrm{V}'\rangle$ *be* LTS*s. A non-empty* $Z \subseteq \mathrm{S} \times \mathrm{S}'$ *is called an* $\mathsf{L_{Kh}}$***-bisimulation** between* $\mathcal{M}$ *and* $\mathcal{M}'$ *if and only if* $sZs'$ *implies all of the following.*

**(Atom):** $\mathrm{V}(s) = \mathrm{V}'(s')$*.*

**(Kh-zig):** *for any propositionally definable set* $P \subseteq \mathrm{S}$*, if* $P \Rightarrow T$ *for some* $T \subseteq \mathrm{S}$*, then there is* $T' \subseteq \mathrm{S}'$ *s.t.:* *1)* $Z(P) \Rightarrow T'$*, and 2)* $T' \subseteq Z(T)$*.*

**(Kh-zag):** *for any propositionally definable set* $P' \subseteq \mathrm{S}'$*, if* $P' \Rightarrow T'$ *for some* $T' \subseteq \mathrm{S}'$*, then there is* $T \subseteq \mathrm{S}$ *s.t.:* *1)* $Z^{-1}(P') \Rightarrow T$*, and 2)* $T \subseteq Z^{-1}(T')$*.*

**(A-zig):** *for all* $t \in \mathrm{S}$ *there is a* $t' \in \mathrm{S}'$ *such that* $tZt'$*.*

**(A-zag):** *for all* $t' \in \mathrm{S}'$ *there is a* $t \in \mathrm{S}$ *such that* $tZt'$*.*

*($Z(X)$ and $Z^{-1}(X')$ have the expected meaning.)*

*We write* $\mathcal{M}, s \leftrightarroweq \mathcal{M}', s'$ *whenever there exists an* $\mathsf{L_{Kh}}$*-bisimulation* $Z$ *between* $\mathcal{M}$ *and* $\mathcal{M}'$ *such that* $sZs'$*.*

The next proposition states a classical adequacy result.

**Proposition 1** ((Fervari, Velázquez-Quesada, and Wang 2022))**.** *Let* $\mathcal{M}, s$ *and* $\mathcal{M}', s'$ *be two pointed* LTS*s such that* $\mathcal{M}, s \leftrightarroweq \mathcal{M}', s'$*. Then,* $\mathcal{M}, s \models \varphi$ *iff* $\mathcal{M}', s' \models \varphi$*, for all* $\mathsf{L_{Kh}}$*-formulas* $\varphi$*. Moreover, if* $\mathcal{M}, \mathcal{M}' \in \mathbf{M_{FD}}$*, the converse also holds.*

Prop. 1 states the classical *invariance under bisimulation* result for $\mathsf{L_{Kh}}$, establishing that bisimilar LTSs satisfy the same formulas. Moreover, the converse holds over finite-domain LTSs, usually called a *Hennessy-Milner* result. The latter was proved to fail over arbitrary LTSs in (Fervari, Velázquez-Quesada, and Wang 2022). This proposition shows that bisimulations as in Def. 8 act as expected w.r.t. the logic $\mathsf{L_{Kh}}$. However, bisimulations are in general defined over structural properties of the models, i.e., over purely semantic conditions. Instead, Def. 8 contains syntactic clauses, since (Kh-zig/zag) rely on a property involving propositional formulas. This also makes difficult to provide a procedural way to operate over bisimulations. For this reason we will revisit the notion of bisimulation and introduce a purely semantic definition. We will follow the ideas investigated in (Areces, Fervari, and Mondejar 2026) for an *uncertainty-based* knowing-how logic, this time over LTSs. Moreover, we will revisit some of the model-theoretic properties of $\mathsf{L_{Kh}}$, completing the view provided in (Fervari, Velázquez-Quesada, and Wang 2022).

### 2.3 Model Checking and Plan Characterization

In (Demri and Fervari 2023) the model-checking problem for $\mathsf{L_{Kh}}$ was investigated. The problem was shown therein to be PSpace-complete. This result will be crucial in our complexity characterization for model-comparison. We define below the model-checking problem for $\mathsf{L_{Kh}}$.

**Definition 9.** *The **model-checking** problem for $\mathsf{L_{Kh}}$ (notation* MCKh*) is defined as:*

$$\texttt{MCKh} \stackrel{def}{=} \{\langle \mathcal{M}, s, \varphi\rangle \mid \textit{where } \mathcal{M} \in \mathbf{M_F}\textit{, } s \textit{ is a state, and } \varphi \textit{ is an } \mathsf{L_{Kh}} \textit{ formula s.t. } \mathcal{M}, s \models \varphi\}.$$

PSpace-membership for MCKh relies in a characterization based on Finite-State Automata (FSA) of the semantic clauses (1) and (2) in Def. 6 (for basic notions on FSA, see e.g. (Hopcroft, Motwani, and Ullman 2006)). These constructions will be also of use in establishing our results. We introduce below all this necessary machinery.

**Definition 10.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*, and let $X, X' \subseteq \mathrm{S}$. Define the FSA $\mathcal{A}_{(X,X')} \stackrel{def}{=} \langle Q, \mathsf{Act}, \delta, I, F\rangle$ as:*

- $Q \stackrel{def}{=} \mathrm{S}$*,* $I \stackrel{def}{=} X$*,* $F \stackrel{def}{=} \mathrm{S} \setminus X'$*.*
- *For all $t, t' \in Q$ and $a \in \mathsf{Act}$, $t \xrightarrow{a} t' \in \delta \stackrel{def}{\Leftrightarrow} (t, t') \in \mathrm{R}_a$.*

The next lemma establishes that $\mathcal{A}_{(X,X')}$ serves to characterize those plans that satisfy condition (2) from Def. 6, by instantiating $X = [\![\varphi]\!]^{\mathcal{M}}$ and $X' = [\![\psi]\!]^{\mathcal{M}}$. Notice that, in what follows, $\mathrm{L}(\mathcal{A})$ denotes the language of the automaton $\mathcal{A}$, and $\bar{\mathcal{A}}$ denotes the complement automaton of $\mathcal{A}$.

**Lemma 1.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*, and let $X, X' \subseteq \mathrm{S}$. Then, $\mathrm{R}_\sigma(X) \subseteq X'$ iff $\sigma \in \mathrm{L}(\bar{\mathcal{A}}_{(X,X')})$.*

Def. 10 slightly differs from the one in (Demri and Fervari 2023) in that here we use arbitrary sets $X, X'$, but it preserves its correctness. This turns out to be handy in Sec. 4. Now, let us move to condition (1) from Def. 6.

**Definition 11.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*, and let $s \in \mathrm{S}$. Define the FSA $\mathcal{A}^\star_s \stackrel{def}{=} \langle Q, \mathsf{Act}, \delta, I, F\rangle$ as:*

- *$Q \stackrel{def}{=} \mathcal{P}(\mathrm{S}) \times \{acc, rej\}$, $I \stackrel{def}{=} \{(\{s\}, acc)\}$, $F \stackrel{def}{=} \mathcal{P}(\mathrm{S}) \times \{acc\}$ ('acc' stands for 'acceptance', 'rej' for 'rejection').*
- *For all $X \in \mathcal{P}(\mathrm{S})$ and $a \in \mathsf{Act}$, $(X, rej) \xrightarrow{a} (X, rej) \in \delta$.*
- *For all $X, X' \in \mathcal{P}(\mathrm{S})$ and $a \in \mathsf{Act}$, $(X, acc) \xrightarrow{a} (X', acc) \in \delta \stackrel{def}{\Leftrightarrow}$ ($\star$) $\mathrm{R}_a(X) = X'$ and ($\star\star$) for all $s \in X$, there is $s' \in X'$ such that $(s, s') \in \mathrm{R}_a$.*
- *For all $X, X' \in \mathcal{P}(\mathrm{S})$ and $a \in \mathsf{Act}$, $(X, acc) \xrightarrow{a} (X', rej) \in \delta \stackrel{def}{\Leftrightarrow}$ ($\star$) and not ($\star\star$).*

We get the intended result to characterize condition (1).

**Lemma 2.** $\{\sigma \mid s \in \mathrm{SE}^{\mathcal{M}}(\sigma)\} = \mathrm{L}(\mathcal{A}^\star_s)$.

Summing up, $\sigma$ is a witness for a formula $\mathsf{Kh}(\varphi, \psi)$ iff $\sigma$ belongs to the following language:

$$\bigcap\{\mathrm{L}(\mathcal{A}^\star_s) \mid s \in [\![\varphi]\!]^{\mathcal{M}}\} \cap \mathrm{L}(\bar{\mathcal{A}}_{([\![\varphi]\!]^{\mathcal{M}}, [\![\psi]\!]^{\mathcal{M}})}).$$

With these results at hand, we obtain:

**Proposition 2** ((Demri and Fervari 2023))**.** MCKh *is* PSpace*-complete.*

*Proof sketch.* Hardness follows by reducing the non-emptiness problem for intersection of deterministic finite-state automata that is a PSpace-complete problem, see e.g. (Galil 1976; Kozen 1977). The problem MCKh can be decided by a classical labelling algorithm. The critical case is that of checking whether $\mathcal{M}, s \models \mathsf{Kh}(p, q)$, which amounts to check *on-the-fly* the non-emptiness of the product FSA

$$\times\{\mathcal{A}^\star_s \mid s \in [\![p]\!]^{\mathcal{M}}\} \times \bar{\mathcal{A}}_{([\![p]\!]^{\mathcal{M}}, [\![q]\!]^{\mathcal{M}})}.$$

The number of states of this FSA is bounded by $2^{|\mathrm{S}|.(|\mathrm{S}|+2)}$ (Demri and Fervari 2023). This yields a non-deterministic procedure running in polynomial space. By Savitch's theorem (Savitch 1970), MCKh is in PSpace. □

It is worth noticing that in the proof sketch above, as stated in (Demri and Fervari 2023, Lemma 1), checking $\mathcal{M}, s \models \mathsf{Kh}(p, q)$ with $p, q \in \mathsf{Prop}$ is already PSpace-hard. We will use this particular instance (as well as the general results) later on the paper.

## 3 Redefining Bisimulations for $\mathsf{L_{Kh}}$

In this section we reformulate the notion of bisimulation for $\mathsf{L_{Kh}}$ while maintaining the logical properties of Def. 8, but relying on structural characteristics only. This definition will be shown to be more suitable for defining model-comparison procedures and establishing other model-theoretic results. To do so, we adapt the ideas recently introduced in (Areces, Fervari, and Mondejar 2026) for a variant of knowing-how.

We start by introducing a natural property that will be of use in the forthcoming results.

**Lemma 3.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*, and $s, t \in \mathrm{S}$ be such that $\mathrm{V}(s) = \mathrm{V}(t)$. Then, for all propositional formulas $\varphi$, $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}, t \models \varphi$.*

Grouping states in terms of their valuation function defines a partition on the model that will be crucial in the definition of our new notion of bisimulation.

**Definition 12.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*, define the relation $\mathcal{Z}_{\mathcal{M}} \stackrel{def}{=} \{(s, t) \in \mathrm{S} \times \mathrm{S} \mid \mathrm{V}(s) = \mathrm{V}(t)\}$. Let $[s] \stackrel{def}{=} \{t \in \mathrm{S} \mid (s, t) \in \mathcal{Z}_{\mathcal{M}}\}$, define $\mathfrak{C}_{\mathcal{M}} \stackrel{def}{=} \{[s] \mid s \in \mathrm{S}\}$.*

*We say that $\mathcal{M}$ is **finite-valuation** if $\mathfrak{C}_{\mathcal{M}}$ has a finite number of elements (i.e., there is a finite number of equivalence classes). Denote $\mathbf{M_{FV}} \stackrel{def}{=} \{\mathcal{M} \mid \mathcal{M}$ is finite-valuation$\}$.*

Notice that it holds that $\mathbf{M_F} \subset \mathbf{M_{FD}} \subset \mathbf{M_{FV}}$. The class $\mathbf{M_{FV}}$ will be of interest in the rest of the section, as it will allow us to obtain results in a more general way compared to existing ones. We now reformulate the property of "propositional definability" in terms of structural properties of the model, for which the next definition will be of use.

**Definition 13.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*. We say that a set $P \subseteq \mathrm{S}$ is $\mathfrak{C}_{\mathcal{M}}$-**saturated** if for all $[s] \in \mathfrak{C}_{\mathcal{M}}$, either $[s] \cap P = \emptyset$ or $[s] \subseteq P$*

Now we state the intended result:

**Lemma 4.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*, with $P \subseteq \mathrm{S}$ a propositionally definable set. Then, $P$ is $\mathfrak{C}_{\mathcal{M}}$-saturated. If $\mathcal{M} \in \mathbf{M_{FV}}$, the converse holds.*

*Proof.* Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V} \rangle$ be an LTS, and let $P \subseteq \mathrm{S}$ be a propositionally definable set, i.e., there exists a propositional formula $\psi$ s.t. $[\![\psi]\!]^{\mathcal{M}} = P$. To prove the lemma, we assume for $[s] \in \mathfrak{C}_{\mathcal{M}}$, $[s] \cap P \neq \emptyset$ and prove that $[s] \subseteq P$. Since $[s] \cap P \neq \emptyset$, then there is $t \in P$ s.t. $[t] = [s]$ and $\mathcal{M}, t \models \psi$. For all $u \in [s]$, we know $\mathrm{V}(u) = \mathrm{V}(t)$, thus by Lemma 3, $\mathcal{M}, u \models \psi$. Then, $u \in P$, and thus $[s] \subseteq P$.

For the converse, assume $\mathcal{M} \in \mathbf{M_{FV}}$, i.e., that $\mathfrak{C}_{\mathcal{M}}$ is finite. Suppose that $P$ is $\mathfrak{C}_{\mathcal{M}}$-saturated. We proceed by case analysis. If $P = \emptyset$, the formula $\psi = \bot$ defines $P$, thus the proof is concluded. Then assume $P \neq \emptyset$. Since $\mathfrak{C}_{\mathcal{M}}$ contains a finite number of elements, let us denote it as $\{[s_1], \ldots, [s_n]\}$, and denote $\mathrm{V}([s])$ as $\mathrm{V}(s)$. Then, for each $j \neq k$, there is $p \in \mathsf{Prop}$ s.t. $p \in \mathrm{V}([s_j])$ and $p \notin \mathrm{V}([s_k])$, or viceversa. Let us denote such a variable as $p_{j,k}$. Define $\psi_j \stackrel{\text{def}}{=} \bigwedge_{k=1,k\neq j}^{n} l(p_{j,k})$, where $l(p_{j,k}) = p_{j,k}$ if $p_{j,k} \in \mathrm{V}([s_j])$, and $l(p_{j,k}) = \neg p_{j,k}$ otherwise. It can be proved that the propositional formula $\bigvee_{[s_j] \subseteq P} \psi_j$ defines $P$, concluding the proof. □

Finally, we proceed to define a novel notion of bisimulation for $\mathsf{L_{Kh}}$ with the desired characteristics.

**Definition 14** ($\mathsf{L^*_{Kh}}$-bisimulation)**.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V} \rangle$ and $\mathcal{M}' = \langle \mathrm{S}', \mathsf{Act}', \mathrm{R}', \mathrm{V}' \rangle$ be* LTS*s. A non-empty $Z \subseteq \mathrm{S} \times \mathrm{S}'$ is called an $\mathsf{L^*_{Kh}}$-**bisimulation** between $\mathcal{M}$ and $\mathcal{M}'$ if and only if $sZs'$ implies all of the following.*

**(Atom), (A-zig), (A-zag):** *as in Def. 8.*

**(Kh-zig*):** *for any $\mathfrak{C}_{\mathcal{M}}$-saturated set $P \subseteq \mathrm{S}$, if $P \Rightarrow T$ for some $T \subseteq \mathrm{S}$, then there is $T' \subseteq \mathrm{S}'$ such that: 1) $Z(P) \Rightarrow T'$, and 2) $T' \subseteq Z(T)$.*

**(Kh-zag*):** *for any $\mathfrak{C}_{\mathcal{M}'}$-saturated set $P' \subseteq \mathrm{S}'$, if $P' \Rightarrow T'$ for some $T' \subseteq \mathrm{S}'$, then there is $T \subseteq \mathrm{S}$ such that: 1) $Z^{-1}(P') \Rightarrow T$, and 2) $T \subseteq Z^{-1}(T')$.*

*We write $\mathcal{M}, s \leftrightarrows^* \mathcal{M}', s'$ when there is an $\mathsf{L^*_{Kh}}$-bisimulation $Z$ between $\mathcal{M}$ and $\mathcal{M}'$ such that $sZs'$. Moreover, $\mathcal{M} \leftrightarrows^* \mathcal{M}'$ iff there are $s, s'$ such that $\mathcal{M}, s \leftrightarrows^* \mathcal{M}', s'$.*

Now we state a correspondence between Defs. 8 and 14. Notice that, compared to the analogous result in (Areces, Fervari, and Mondejar 2026) in the context of uncertainty-based knowing-how, here we generalize the class of LTSs for which the converse holds, as $\mathbf{M_{FD}} \subset \mathbf{M_{FV}}$.

**Lemma 5.** *If a relation $Z$ is an $\mathsf{L^*_{Kh}}$-bisimulation between two* LTS*s $\mathcal{M}$ and $\mathcal{M}'$, then it is an $\mathsf{L_{Kh}}$-bisimulation between them. If $\mathcal{M}, \mathcal{M}' \in \mathbf{M_{FV}}$, then the converse also holds.*

*Proof sketch.* It follows by using Lemma 4. □

As a consequence of the previous lemma, we are able to extend the adequacy results obtained in (Areces, Fervari, and Mondejar 2026; Fervari, Velázquez-Quesada, and Wang 2022). Here we establish that the class of models $\mathbf{M_{FV}}$ is also a Hennessy-Milner class, generalizing previous results.

**Theorem 1.** *Let $\mathcal{M}, s$ and $\mathcal{M}', s'$ be two* LTS*s such that $\mathcal{M}, s \leftrightarrows^* \mathcal{M}', s'$. Then, for all $\mathsf{L_{Kh}}$- formulas $\varphi$, $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}', s' \models \varphi$. If $\mathcal{M}, \mathcal{M}' \in \mathbf{M_{FV}}$, the converse also holds.*

*Proof.* The property for arbitrary LTSs follows by Lemma 5 and Prop. 1. For the converse is enough to show that $Z = \{(t, t') \in \mathrm{S} \times \mathrm{S}' \mid \mathrm{V}(t) = \mathrm{V}'(t')\}$ is an $\mathsf{L^*_{Kh}}$-bisimulation that contains the pair $(s, s')$. Suppose $(s, s') \notin Z$, then there exists $p \in \mathsf{Prop}$ s.t. $p \in \mathrm{V}(s)$ and $p \notin \mathrm{V}'(s')$ or viceversa, and therefore, the two pointed LTSs do not coincide in the formula $\varphi = p$, which is a contradiction. Thus, $(s, s') \in Z$. Let us now check that $Z$ is an $\mathsf{L^*_{Kh}}$-bisimulation. Notice that (Atom) is directly satisfied by definition.

**(A-zig):** Suppose $Z$ does not satisfy (A-zig), then there exists $t \in \mathrm{S}$ s.t. $\mathrm{V}(t) \neq \mathrm{V}'(t')$ for all $t' \in \mathrm{S}'$. Since $\mathcal{M}' \in \mathbf{M_{FV}}$, $\mathfrak{C}_{\mathcal{M}'}$ is finite, so we can denote it as $\{[t'_1], ..., [t'_n]\}$. For each $[t'_i]$, there exists $p_i \in \mathsf{Prop}$ s.t. $p_i \in \mathrm{V}(t)$ and $p_i \notin \mathrm{V}'(t'_i)$ or viceversa. Define $\varphi \stackrel{\text{def}}{=} \bigwedge_{i=1}^{n} l(i)$ where $l(i) = p_i$ if $p_i \in \mathrm{V}(t)$, and $l(i) = \neg p_i$ otherwise. Then, it is not difficult to see that $\mathcal{M}, s \models \mathsf{E}\varphi$ and $\mathcal{M}', s' \not\models \mathsf{E}\varphi$, yielding a contradiction. Thus, $Z$ satisfies (A-zig).

**(Kh-zig*):** Suppose $Z$ does not satisfy (Kh-zig*), then there exist $P, T$ as in Def. 14 for which, in particular, $Z(P) \not\Rightarrow Z(T)$. Define $T^+ \stackrel{\text{def}}{=} \bigcup_{t \in T}[t]$. It can be seen that $P \Rightarrow T^+$ and $Z(T) = Z(T^+)$. Moreover, since $\mathcal{M}, \mathcal{M}' \in \mathbf{M_{FV}}$, from Lemma 4 we get that $P, T^+, Z(P)$ and $Z(T^+)$ are all propositionally definable (with formulas $\varphi_P, \varphi_{T^+}, \varphi_{Z(P)}$ and $\varphi_{Z(T^+)}$ respectively). Now, given that $Z$ satisfies (A-zig), it can be proven that the formula $\varphi_P \wedge \varphi_{Z(P)}$ defines $P$ in $\mathcal{M}$ and $Z(P)$ in $\mathcal{M}'$ (analogously $\varphi_{T^+} \wedge \varphi_{Z(T^+)}$). So by considering $\varphi = \mathsf{Kh}(\varphi_P \wedge \varphi_{Z(P)}, \varphi_{T^+} \wedge \varphi_{Z(T^+)})$, we have that $\mathcal{M}, s \models \varphi$ and $\mathcal{M}', s' \not\models \varphi$, which is a contradiction. Thus, $Z$ satisfies (Kh-zig*).

**(A-zag) / (Kh-zag*):** Analogous to the previous cases. □

As a novelty, for $\mathsf{L^*_{Kh}}$-bisimulations, the class $\mathbf{M_{FV}}$ is a so-called Hennessy-Milner class that is more general to those already known, in e.g. (Fervari, Velázquez-Quesada, and Wang 2022). Furthermore, in (Areces, Fervari, and Mondejar 2026, Th. 3.6) the Hennessy-Milner result is also shown for finite-domain models for uncertainty-based knowing-how, but it can be extended to finite-valuation models by using arguments similar to the ones just presented.

**Example 1.** *Let $\mathcal{M}$ and $\mathcal{M}'$ be two* LTS*s whose graphical representation is in Fig. 1. Note that $\mathcal{M}, s_1 \leftrightarrows^* \mathcal{M}', s'_1$, since $Z$ defined by the dashed arrows is an $\mathsf{L^*_{Kh}}$-bisimulation. For instance, we can check that $[s_1] = \{s_1\}$ and that $[s_1] \Rightarrow [s_4] = \{s_4\}$, with witness bc. We can take $T' = [s'_2] = \{s'_2\}$, since it satisfies that 1) $Z([s_1]) \Rightarrow T'$ (witness e), and 2) $T' \subseteq Z([s_4])$ (actually, $T' = [s'_2] = Z([s_4])$).*

It is worth noticing that an $\mathsf{L_{Kh}}$-bisimulation $Z$ is not necessarily an $\mathsf{L^*_{Kh}}$-bisimulation (although this is the case in $\mathbf{M_{FV}}$). Let $\mathsf{Prop}$ be an infinite and countable set of propositional symbols. Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V} \rangle$ be an LTS s.t. $\mathrm{S} = \mathcal{P}(\mathsf{Prop})$ and $\mathrm{V}(s) = s$. Clearly, $\mathcal{P}(\mathsf{Prop})$ is not countable. However, the set of all the propositional formulas built over $\mathsf{Prop}$ is countable, thus there is some $s \in \mathcal{P}(\mathsf{Prop})$ such that $\{s\}$ is not propositionally definable. Hence, the converse of Lemma 4 in the general case does not hold.

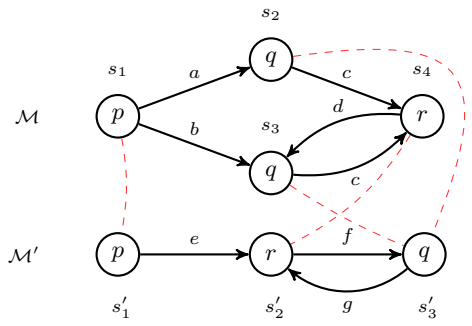


Figure 1: Two $\mathsf{L}^*_{\mathsf{Kh}}$-bisimilar LTSs.

With this at hand, we can define $\mathcal{M}$ and $\mathcal{M}'$ such that they are $\mathsf{L}_{\mathsf{Kh}}$-bisimilar but not $\mathsf{L}^*_{\mathsf{Kh}}$-bisimilar. Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \{\mathrm{R}_a, \mathrm{R}_b\}, \mathrm{V}\rangle$ and $\mathcal{M}' = \langle \mathrm{S}, \mathsf{Act}, \{\mathrm{R}_a, \emptyset\}, \mathrm{V}\rangle$ be s.t. $\mathrm{S} = \mathcal{P}(\mathsf{Prop})$, $\mathsf{Act} = \{a, b\}$, $\mathrm{V}(s) = s$. Define also $\mathrm{R}_a = \mathrm{S} \times \mathrm{S}$, and, for a fix $s$ s.t. $\{s\}$ is not propositionally definable, $\mathrm{R}_b = \{(s, s')\}$ with $s' \in \mathrm{S}$ s.t. $s \neq s'$. Take $P = \{s\}$ as in Def. 14, then $\{s\} \Rightarrow \{s'\}$ via $b$ in $\mathcal{M}$. However, there is no matching $\sigma \in \mathsf{Act}^*$ in $\mathcal{M}'$, since the only possible witnesses are plans of $\{a\}^*$, but $\mathrm{R}_a(X) = \mathrm{S}$ for all $X \subseteq \mathrm{S}$ and $\mathrm{R}_\epsilon(\{s\}) = \{s\} \not\subseteq \{s'\}$. Notice that the action $b$ can only be executed in the set $\{s\}$ (otherwise it would not derive in a strongly executable plan), therefore the behaviors of $\mathcal{M}$ and $\mathcal{M}'$ coincide in any other subset of S. Thus, $\mathcal{M}, s \leftrightarrows \mathcal{M}', s$ but $\mathcal{M}, s \not\leftrightarrows^* \mathcal{M}', s$ does not hold. This situation is natural since $\mathsf{L}_{\mathsf{Kh}}$-bisimulations involve a finite characterization of sets via propositional formulas, while $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulations get rid of such a condition.

An interesting property of $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulations is that, if two models are bisimilar, then we can always build a relation that acts as a *universal bisimulation* $\mathfrak{B}$ between them, that is in addition the maximal one.

**Lemma 6.** *Let $\mathcal{M}$ and $\mathcal{M}'$ be* LTS*s s.t. $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ and $\mathcal{M}' = \langle \mathrm{S}', \mathsf{Act}', \mathrm{R}', \mathrm{V}'\rangle$. $\mathcal{M} \leftrightarrows^* \mathcal{M}'$ implies that $\mathfrak{B} \stackrel{def}{=} \{(t, t') \in \mathrm{S} \times \mathrm{S}' \mid \mathrm{V}(t) = \mathrm{V}'(t')\}$ is an $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. Moreover, $\mathfrak{B}$ is the maximal bisimulation between $\mathcal{M}$ and $\mathcal{M}'$.*

*Proof sketch.* Similar to (Areces, Fervari, and Mondejar 2026, Lemma 4.6). □

We conclude with an application of our developments. Concretely, we show that with the results introduced above we can solve the *definability problem* (see, e.g., (Areces, Figueira, and Gorín 2011; Arenas and Diaz 2016) for examples of its use) for $\mathsf{L}_{\mathsf{Kh}}$ in polynomial time.

**Corollary 1** (Definability)**.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle \in \mathbf{M_F}$, and let $X \subseteq \mathrm{S}$. The problem of deciding whether there exists an $\mathsf{L}_{\mathsf{Kh}}$-formula $\varphi$ such that $[\![\varphi]\!]^{\mathcal{M}} = X$ is in* P*.*

*Proof sketch.* Similar to (Areces, Fervari, and Mondejar 2026, Cor. 3.7). □

# 4 A Computational Analysis of Bisimulations

This section is devoted to investigate some computational properties of $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulations. Concretely, we focus on two natural problems related to bisimulations: the complexity of model comparison via bisimulations, and model contraction. Even though the notion of $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation is very similar to that from (Areces, Fervari, and Mondejar 2026) (modulo what the relation '$\Rightarrow$' means), the results obtained here differ from the ones in the aforementioned work.

## 4.1 Complexity of Model Comparison

Let us introduce the main problem we tackle in this section.

**Definition 15.** *Define the problem* `KhBisim` *as:*

$$\texttt{KhBisim} \stackrel{def}{=} \{\langle \mathcal{M}, \mathcal{M}'\rangle \mid \mathcal{M}, \mathcal{M}' \in \mathbf{M_F} \text{ and } \mathcal{M} \leftrightarrows^* \mathcal{M}'\}.$$

In (Areces, Fervari, and Mondejar 2026), this problem is approached as a "search and verification problem" on the logic of knowing-how with uncertainty sets or perceptions, relying on finding a potential candidate $Z$ for a bisimulation and checking that $Z$ actually is a bisimulation. This is not needed here, as a consequence of the intrinsic complexity of the problem. In particular, we will define a nondeterministic procedure running in polynomial space, which will be proved to be optimal by finding a corresponding PSpace lower-bound. We start by showing the latter, which follows by reduction of the model-checking problem for $\mathsf{L}_{\mathsf{Kh}}$ (Demri and Fervari 2023).

**Lemma 7.** *Deciding* `KhBisim` *is* PSpace*-hard.*

*Proof.* We proceed by reduction of `MCKh`. Define $f$ : `MCKh` $\rightarrow$ `KhBisim` as the following map between instances of the respective problems. Let $\langle \mathcal{M}, s, \varphi\rangle$ be an instance of `MCKh` where $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ and $\varphi = \mathsf{Kh}(p, q)$ with $p, q \in \mathsf{Prop}$. Then, $f$ maps $\langle \mathcal{M}, s, \varphi\rangle$ to $\langle \mathcal{M}, \mathcal{M}'\rangle$ where:

- $\mathcal{M}' = \langle \mathrm{S}, \mathsf{Act} \cup \{test\}, \mathrm{R}', \mathrm{V}\rangle$, with
- $\mathrm{R}' = \mathrm{R} \cup \{\mathrm{R}_{test}\}$,
- $\mathrm{R}_{test} = \{(t, u) \in \mathrm{S} \times \mathrm{S} \mid t \in [\![p]\!]^{\mathcal{M}} \text{ and } u \in [\![q]\!]^{\mathcal{M}}\}$.

Due to the fact that `MCKh` is already PSpace-hard for formulas of the form $\mathsf{Kh}(p, q)$ (see (Demri and Fervari 2023)), we only focus on reducing the case where $\varphi$ is of that form. Also, since the Kh operator acts globally, the state $s$ does not play any role in the proof.

We also notice that if $[\![p]\!]^{\mathcal{M}} = \emptyset$, then trivially $\mathcal{M}, s \models \mathsf{Kh}(p, q)$. Moreover, if we have that $[\![p]\!]^{\mathcal{M}} \neq \emptyset$ and $[\![q]\!]^{\mathcal{M}} = \emptyset$, then $\mathcal{M}, s, \not\models \mathsf{Kh}(p, q)$. Thus, w.l.o.g., we assume that both $[\![p]\!]^{\mathcal{M}}$ and $[\![q]\!]^{\mathcal{M}}$ are non-empty (otherwise $f$ can be easily modified to map $\mathsf{Kh}(p, q)$ to bisimilar or respectively non-bisimilar models).

It is clear that $|\mathcal{M}| + |\mathcal{M}'|$ is polynomial w.r.t. the size of $\langle \mathcal{M}, s, \varphi\rangle$. Moreover, since computing $[\![p]\!]^{\mathcal{M}}$ and $[\![q]\!]^{\mathcal{M}}$ only requires performing propositional logic model-checking in each state of $\mathcal{M}$, then $f$ can be computed in polynomial time w.r.t. the size of $\langle \mathcal{M}, s, \varphi\rangle$.

We need to prove that $\mathcal{M}, s \models \varphi$ iff there exists an $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$.

$\Rightarrow$) Suppose $\mathcal{M}, s \models \mathsf{Kh}(p, q)$. Let us check that $Z = \{(t, t) \mid t \in \mathrm{S}\}$ is an $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$.

If $tZt'$, then $\mathrm{V}(t) = \mathrm{V}(t')$ since by definition $t = t'$, therefore $Z$ satisfies (Atom). (A-zig/zag) are also directly satisfied by definition of $Z$. For (Kh-zig*), notice that every witness in $\mathcal{M}$ can be mimicked on $\mathcal{M}'$, then this condition directly holds. Let us see that $Z$ satisfies (Kh-zag*).

Let us first prove a claim that will be of use in (Kh-zag*).

**Claim ($\star$):** Let $P \subseteq \mathrm{S}$. For each $\sigma \in \{\mathsf{Act} \cup \{test\}\}^*$ such that $P \subseteq \mathrm{SE}^{\mathcal{M}'}(\sigma)$ there exists $\gamma \in \mathsf{Act}^*$ such that $P \subseteq \mathrm{SE}^{\mathcal{M}}(\gamma)$ and $\mathrm{R}_\gamma(P) \subseteq \mathrm{R}'_\sigma(P)$.

The proof of the claim proceeds by induction on the length of $\sigma$ (recall we are working under the assumption that $\mathcal{M}, s \models \mathsf{Kh}(p, q)$). Notice that if $P = \emptyset$, then any $\gamma \in \mathsf{Act}^*$ does the job. So we will also assume $P \neq \emptyset$.

**Base Case** ($|\sigma|$=0)**:** Note that $\sigma = \epsilon$, then take $\gamma = \epsilon$.

**Inductive Case** ($|\sigma|$=$k+1$)**:** Let $P \subseteq \mathrm{S}$ and $\sigma \in \{\mathsf{Act} \cup \{test\}\}^*$ s.t. $|\sigma| = k+1$ and $P \subseteq \mathrm{SE}^{\mathcal{M}'}(\sigma)$. Note that $P \subseteq \mathrm{SE}^{\mathcal{M}'}(\sigma_k)$, then by IH there is $\gamma' \in \mathsf{Act}^*$ s.t. $P \subseteq \mathrm{SE}^{\mathcal{M}}(\gamma')$ and $\mathrm{R}_{\gamma'}(P) \subseteq \mathrm{R}'_{\sigma_k}(P)$. We proceed by case analysis:

- $\sigma[k+1] \neq test$. In this case $\gamma = \gamma'\sigma[k+1]$ does the job, given that $\mathrm{R}_{\sigma[k+1]} = \mathrm{R}'_{\sigma[k+1]}$.
- $\sigma[k+1] = test$. Notice that necessarily $\mathrm{R}'_{\sigma_k}(P) \subseteq [\![p]\!]^{\mathcal{M}}$ and $\mathrm{R}'_{\sigma_k}(P) \neq \emptyset$, otherwise $P \subseteq \mathrm{SE}^{\mathcal{M}'}(\sigma)$ would not hold. Then, by analyzing the definition of $\mathrm{R}'$ we get that $\mathrm{R}'_\sigma(P) = [\![q]\!]^{\mathcal{M}}$. Since $\mathcal{M}, s \models \mathsf{Kh}(p, q)$, there is $\alpha \in \mathsf{Act}^*$ s.t. $[\![p]\!]^{\mathcal{M}} \subseteq \mathrm{SE}^{\mathcal{M}}(\alpha)$ and $\mathrm{R}_\alpha([\![p]\!]^{\mathcal{M}}) \subseteq [\![q]\!]^{\mathcal{M}}$. Then, it can be proven that $\gamma = \gamma'\alpha$ does the job.

Let us resume the proof of (Kh-zag*). Suppose that $P', T' \subseteq \mathrm{S}$ are as in Def. 14, we will show that there exists $T \subseteq \mathrm{S}$ s.t.: 1) $Z^{-1}(P') \Rightarrow T$ and 2) $T \subseteq Z^{-1}(T')$. Let us see that $T = T'$ does the job. Notice that 2) is directly satisfied since $Z^{-1}(T') = T' = T$.

Since $P' \Rightarrow T'$ in $\mathcal{M}'$, there is $\sigma \in \{\mathsf{Act} \cup \{test\}\}^*$ s.t. $P' \subseteq \mathrm{SE}^{\mathcal{M}'}(\sigma)$ and $\mathrm{R}'_\sigma(P') \subseteq T'$. From the claim ($\star$), there exists $\gamma \in \mathsf{Act}^*$ s.t. $P' \subseteq \mathrm{SE}^{\mathcal{M}}(\gamma)$ and $\mathrm{R}_\gamma(P') \subseteq \mathrm{R}'_\sigma(P')$. Thus, $P' = Z^{-1}(P') \Rightarrow T' = T$ in $\mathcal{M}$, proving 1).

$\Leftarrow$) Suppose that there exists an $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation $Z$ between $\mathcal{M}$ and $\mathcal{M}'$. Let us check that $\mathcal{M}, s \models \mathsf{Kh}(p, q)$. First, notice that $[\![p]\!]^{\mathcal{M}} \Rightarrow [\![q]\!]^{\mathcal{M}}$ in $\mathcal{M}'$ with witness $test$. Also, since $[\![p]\!]^{\mathcal{M}}$ is propositionally definable, from Lemma 4 we get that $[\![p]\!]^{\mathcal{M}}$ is $\mathfrak{C}_{\mathcal{M}'}$-saturated. Since $Z$ satisfies (Kh-zag*), there exists $T$ such that $Z^{-1}([\![p]\!]^{\mathcal{M}}) \Rightarrow T$ in $\mathcal{M}$ and $T \subseteq Z^{-1}([\![q]\!]^{\mathcal{M}})$. Also, given that $Z$ satisfies (A-zig) and (Atom), it holds that $Z^{-1}([\![p]\!]^{\mathcal{M}}) = [\![p]\!]^{\mathcal{M}}$ and $Z^{-1}([\![q]\!]^{\mathcal{M}}) = [\![q]\!]^{\mathcal{M}}$. Therefore, there exists $\sigma \in \mathsf{Act}^*$ s.t. $[\![p]\!]^{\mathcal{M}} \subseteq \mathrm{SE}^{\mathcal{M}}(\sigma)$ and $\mathrm{R}_\sigma([\![p]\!]^{\mathcal{M}}) \subseteq T \subseteq [\![q]\!]^{\mathcal{M}}$. Thus, $\mathcal{M}, s \models \mathsf{Kh}(p, q)$. □

**Example 2.** *Fig. 2 diplays an example of an* LTS $\mathcal{M}$ *and its encoding into the corresponding* $\mathcal{M}'$*, for the formula* $\varphi = \mathsf{Kh}(p, r)$*. Notice that the latter is exactly as the former except that the edges labeled by test are added, which are colored in red. These edges witness the satisfaction of* $\mathsf{Kh}(p, r)$ *at* $\mathcal{M}'$*, which can be mimicked by abc in* $\mathcal{M}$*.*

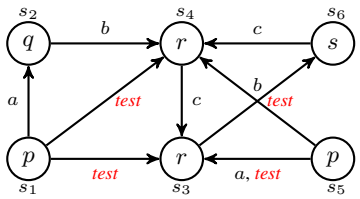


Figure 2: $\mathcal{M}$ and its encoding to $\mathcal{M}'$ for $\varphi = \mathsf{Kh}(p, r)$.

Once the PSpace lower-bound has been established, we will provide an algorithm deciding KhBisim that runs in polynomial space. Again, we will take advantage of the machinery used to obtain the PSpace membership of MCKh, that we introduced in Sec. 2.3.

**Lemma 8.** *Deciding* KhBisim *is in* PSpace.

*Proof.* We will provide a non-deterministic algorithm running in polynomial space, that proceeds as follows over two LTSs $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V} \rangle$ and $\mathcal{M}' = \langle \mathrm{S}', \mathsf{Act}', \mathrm{R}', \mathrm{V}' \rangle$.
**(Part I):** Guess a relation $Z \subseteq \mathrm{S} \times \mathrm{S}'$ that is a candidate to be an $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation. Clearly, $|Z| \leq \mathfrak{f}_1(|\mathrm{S}| + |\mathrm{S}'|)$ for some polynomial $\mathfrak{f}_1$.

**(Part II):** Guess a certificate $c = \langle X, Y, b \rangle$, where $X$ and $Y$ are subset of states from one of the models playing the roles of $P$ and $T$ or resp. $P'$ and $T'$ in Def. 14, and $b$ is a bit indicating to which model we are referring to ($b = 1$ for $\mathcal{M}$ and $b = 0$ for $\mathcal{M}'$). The certificate $c$ will act as a counterexample witnessing that $Z$ is not an $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation. Notice that $|c| \leq \mathfrak{f}_2(|\mathcal{M}| + |\mathcal{M}'| + |Z|)$ for some polynomial $\mathfrak{f}_2$.

**(Part III):** Check that $c$ is indeed a counterexample. Checking (Atom), (A-zig) and (A-zag) can clearly be done in polynomial time w.r.t. the size of the models, and any certificate works as a counterexample. Thus, the critical part is to check the conditions (Kh-zig*/zag*). The function CounterExCheck in Alg. 1 checks if (Kh-zig*) is satisfied over the sets $X$ and $Y$ playing the roles of $P$ and $T$ resp. in Def. 14 ((Kh-zag*) is obtained by inverting the parameters in line 13, whenever $b = 0$). Conditions in lines 7 and 11 require computing $\mathfrak{C}_{\mathcal{M}}$ and $\mathfrak{C}_{\mathcal{M}'}$, and checking saturation, which can be done in polynomial time on the size of the models. The rest of the solution relies on invoking the function NotSimulates, which verifies if the given input is a counterexample (returns 1 if the input is a counterexample for $Z$, and 0 otherwise). In line 2, NotSimulates checks non-emptiness of the language of the product FSA built from Defs. 10 and 11 using $X$ and $Y$ as input, and the emptiness of the analogous FSA for $X'$ and $Y'$ in line 4. As in Prop. 2, these checks can be done on-the-fly using polynomial space.

Parts II and III together give us a procedure that decides whether $Z$ is not a $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation, running in NPSpace. Since NPSpace is closed under complements, deciding whether $Z$ is an $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation is in NPSpace. Correctness of Part III follows by the correctness argument used for MCKh. Then, together with Part I we get a procedure running in NPSpace for deciding KhBisim, and since NPSpace = PSpace (Savitch 1970), KhBisim is in PSpace. □

**Algorithm 1** Certificate Verifier

**function** NOTSIMULATES($\mathcal{M}, \mathcal{M}', Z, X, Y$)
  **if** $\mathrm{L}(\times\{\mathcal{A}^\star_s \mid s \in X\} \times \bar{\mathcal{A}}_{(X,Y)}) \neq \emptyset$ **then**
    $X', Y' \leftarrow Z(X), Z(Y)$
    **if** $\mathrm{L}(\times\{\mathcal{A}^\star_s \mid s \in X'\} \times \bar{\mathcal{A}}_{(X',Y')}) = \emptyset$ **then return** 1
  **return** 0
**function** COUNTEREXCHECK($\mathcal{M}, \mathcal{M}', \langle Z, X, Y, b\rangle$)
  **if** $b$ **then**
    **if** $X$ is not $\mathfrak{C}_{\mathcal{M}}$-saturated **then**
      **return** 0
    **return** NOTSIMULATES($\mathcal{M}, \mathcal{M}', Z, X, Y$)
  **else**
    **if** $X$ is not $\mathfrak{C}_{\mathcal{M}'}$-saturated **then**
      **return** 0
    **return** NOTSIMULATES($\mathcal{M}', \mathcal{M}, Z^{-1}, X, Y$)

**Theorem 2.** *Deciding* `KhBisim` *is* PSpace*-complete.*

*Proof.* It follows by Lemmas 7 and 8. □

As a by-product, it can be shown that the pointed variant of `KhBisim` is also PSpace-complete.

**Corollary 2.** *Deciding* `PKhBisim` *is* PSpace*-complete, where:* $\texttt{PKhBisim} \stackrel{def}{=} \{\langle \mathcal{M}, s, \mathcal{M}', s'\rangle \mid \mathcal{M}, s \leftrightarrows^* \mathcal{M}', s'\}$.

*Proof sketch.* It follows by checking if $(s, s')$ belongs to $Z$ after Part III above. □

We conclude this section with an application of the results we just presented. Over the class $\mathbf{M_F}$, we have that two LTSs are not $\mathsf{L}^*_{\mathsf{Kh}}$-bisimilar *iff* there is an $\mathsf{L}_{\mathsf{Kh}}$-formula that distinguishes them. This is a consequence of Thm. 1 and $\mathbf{M_F} \subset \mathbf{M_{FV}}$. It is in fact possible to show that such a formula can be obtained from Alg. 1. Let $\mathsf{Prop}'$ be the finite set of propositional symbols appearing in the valuation of some of the models under comparison. For a model $\mathcal{M}$ and a set of states $X$, we define $\varphi_X \stackrel{\text{def}}{=} \bigvee_{[s]\in\mathfrak{C}_{\mathcal{M}},[s]\subseteq X} \varphi_{[s]}$, where $\varphi_{[s]} \stackrel{\text{def}}{=} \bigwedge_{p\in \mathrm{V}(s)} p \wedge \bigwedge_{p\in \mathsf{Prop}'\setminus \mathrm{V}(s)} \neg p$. With this at hand, we can build a formula distinguishing the models.

**Theorem 3.** *For $\mathcal{M}, \mathcal{M}' \in \mathbf{M_F}$, if $\mathcal{M}, s \not\leftrightarrows^* \mathcal{M}', s'$, then there is a polynomial size $\mathsf{L}_{\mathsf{Kh}}$-formula $\varphi$ s.t. $\mathcal{M}, s \models \varphi$ and $\mathcal{M}', s' \not\models \varphi$.*

*Proof sketch.* We can use Alg. 1 with $Z = \mathfrak{B}$ ($\mathfrak{B}$ as in Lemma 6) and see in which case the bisimulation fails, to obtain the intended formula. If $(s, s') \notin \mathfrak{B}$ (resp. (A-zig/zag)) then, take $\varphi = \varphi_{[s]}$ (resp. $\varphi = \mathsf{E}\varphi_{[t]}$, with $t$ being the state where the condition fails). If (Kh-zig*) fails ((Kh-zag*) is analogous), take $\varphi = \mathsf{Kh}(\varphi_X, \varphi_{Y^+})$ where $X, Y$ are the input of the algorithm and $Y^+ = \bigcup_{t\in Y}[t]$. □

## 4.2 Bisimulation-Based Contractions

One of the most common applications of bisimulation is the one of model minimization or model contraction. Given a model, the goal is to obtain a smaller one that preserves its properties. Bisimulation is usually the right way to guide this preservation condition. Model contraction is crucial for model-checking, given that the efficiency of verification algorithms depends fundamentally on the size of the model.

We start this section with a model contraction that relies on the relation from Def. 12.

**Lemma 9.** *Let $\mathcal{M}$ be an* LTS*. Then, the relation $\mathcal{Z}_{\mathcal{M}}$ as in Def. 12 is an auto-bisimulation. Moreover, $\mathcal{Z}_{\mathcal{M}}$ is the maximal auto-bisimulation of $\mathcal{M}$.*

Now we proceed to defining a way to minimize models based on $\mathcal{Z}_{\mathcal{M}}$. Similar to what happens in (Areces, Fervari, and Mondejar 2026) the main challenge is defining how to connect the states. Notice that therein plans are given as input, which is not the case here. So, we need to come up with a new way of defining the accessibility relations.

**Definition 16.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*. Its **valuation contraction** is defined as $\mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}} = \langle \mathrm{S}', \mathsf{Act}', \mathrm{R}', \mathrm{V}'\rangle$, where:*

- $\mathrm{S}' \stackrel{def}{=} \mathfrak{C}_{\mathcal{M}}$, $\mathrm{V}'([s]) \stackrel{def}{=} \mathrm{V}(s)$, $\mathrm{R}' \stackrel{def}{=} \bigcup_{a\in\mathsf{Act}'} \mathrm{R}'_a$ *with:*
- $\mathsf{Act}' \stackrel{def}{=} \{a_{P,T} \mid P, T \subseteq \mathrm{S}$ *are $\mathfrak{C}_{\mathcal{M}}$-saturated sets, $P \Rightarrow T$, and $P \not\subseteq T$*$\}$,
- $\mathrm{R}'_{a_{P,T}} \stackrel{def}{=} \{([s],[t]) \in \mathrm{S}' \times \mathrm{S}' \mid [s] \subseteq P$ *and* $[t] \subseteq T\}$.

**Theorem 4.** *Let $\mathcal{M} \in \mathbf{M_F}$ with $\mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}$ as in Def. 16. Then, for all states $s$, we have $\mathcal{M}, s \leftrightarrows^* \mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}, [s]$. Moreover, $\mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}$ can be computed in polynomial space.*

*Proof.* It suffices to show that $Z = \{(s, [s]) \mid s \in \mathrm{S}\}$ is an $\mathsf{L}^*_{\mathsf{Kh}}$-bisimulation. Notice that, by definition, conditions (Atom), (A-zig) and (A-zag) are directly satisfied. Let us focus on proving (Kh-zig*) and (Kh-zag*).

(Kh**-zig*):** Let $P, T \subseteq \mathrm{S}$ be as in Def. 14. Let us show that there is $T' \subseteq \mathrm{S}'$ s.t. 1) $Z(P) \Rightarrow T'$ and 2) $T' \subseteq Z(T)$. Consider $T' = Z(T)$, then 2) is directly satisfied. Since $P \Rightarrow T$, then $P \Rightarrow T^+$ with $T^+ = \bigcup_{t\in T}[t]$. Also, by definition of $Z$, it is clear that $Z(T^+) = Z(T)$. We proceed by case analysis:

- $P \subseteq T^+$: We have that $Z(P) \subseteq Z(T^+) = Z(T)$. Hence, $\epsilon$ serves as a witness for $Z(P) \Rightarrow Z(T)$.
- $P \not\subseteq T^+$: Since $P \Rightarrow T^+$ and both $P, T^+$ are $\mathfrak{C}_{\mathcal{M}}$-saturated, we have that $a_{P,T^+} \in \mathsf{Act}'$ and $([s],[t]) \in \mathrm{R}'_{a_{P,T^+}}$ for every $[s] \subseteq P$ and $[t] \subseteq T^+$. By analyzing the definition of $Z$, it is not difficult to see that the states involved in these edges are exactly the ones of $Z(P)$ and $Z(T^+)$, therefore we have that $Z(P) \Rightarrow Z(T^+) = Z(T)$ with witness $a_{P,T^+}$.

(Kh**-zag*):** Let us first prove a claim that will be of use in the analysis of this case. Below, suppose $P' \subseteq \mathrm{S}'$.

> **Claim ($\star\star$):** For all $\sigma \in \mathsf{Act}'^*$ s.t. $P' \subseteq \mathrm{SE}^{\mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}}(\sigma)$, there exists $\gamma \in \mathsf{Act}^*$ s.t. $Z^{-1}(P') \subseteq \mathrm{SE}^{\mathcal{M}}(\gamma)$ and $\mathrm{R}_\gamma(Z^{-1}(P')) \subseteq Z^{-1}(\mathrm{R}'_\sigma(P'))$.

The proof of the claim proceeds by induction on the length of $\sigma$. Notice that if $P' = \emptyset$, then any $\gamma \in \mathsf{Act}^*$ does the job. We will then assume that $P' \neq \emptyset$.

**Base Case** ($|\sigma|{=}0$)**:** Note that $\sigma = \epsilon$, then take $\gamma = \epsilon$.

**Inductive Case** ($|\sigma|{=}k+1$)**:** Let $P' \subseteq \mathrm{S}'$ and $\sigma \in \mathsf{Act}'^{*}$ s.t. $|\sigma| = k+1$ and $P' \subseteq \mathrm{SE}^{\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}}(\sigma)$. Since $P' \subseteq \mathrm{SE}^{\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}}(\sigma_k)$, by IH there exists $\gamma' \in \mathsf{Act}^{*}$ s.t. $Z^{-1}(P') \subseteq \mathrm{SE}^{\mathcal{M}}(\gamma')$ and $\mathrm{R}_{\gamma'}(Z^{-1}(P')) \subseteq Z^{-1}(\mathrm{R}'_{\sigma_k}(P'))$. Let $\sigma[k+1] = a_{X,T}$ with $X, T \subseteq \mathrm{S}$, then necessarily for each $[s] \in \mathrm{R}'_{\sigma_k}(P')$ we have that $[s] \subseteq X$, otherwise $P' \subseteq \mathrm{SE}^{\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}}(\sigma)$ would not hold. Therefore, $\mathrm{R}_{\gamma'}(Z^{-1}(P')) \subseteq Z^{-1}(\mathrm{R}'_{\sigma_k}(P')) \subseteq X$. Since $\mathrm{R}'_{\sigma_k}(P') \neq \emptyset$, we have that $\mathrm{R}'_{\sigma}(P') = \{[s] \in \mathrm{S}' \mid [s] \subseteq T\}$, thus $Z^{-1}(\mathrm{R}'_{\sigma}(P')) = T$.

By definition of $\mathsf{Act}'$, we get that $X \Rightarrow T$ in $\mathcal{M}$. Then, there exists $\alpha \in \mathsf{Act}^{*}$ s.t. $X \subseteq \mathrm{SE}^{\mathcal{M}}(\alpha)$ and $\mathrm{R}_\alpha(X) \subseteq T$. Therefore, it can be shown that $\gamma = \gamma'\alpha$ does the job.

Let us resume the proof of (Kh-zag*). Take $P', T' \subseteq \mathrm{S}'$ as in Def. 14. We will show that there is $T \subseteq \mathrm{S}$ s.t. 1) $Z^{-1}(P') \Rightarrow T$ and 2) $T \subseteq Z^{-1}(T')$. Let us see that $T = Z^{-1}(T')$ does the job. Notice that 2) is directly satisfied. Since $P' \Rightarrow T'$ in $\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}$, there exists $\sigma \in \mathsf{Act}'^{*}$ s.t. $P' \subseteq \mathrm{SE}^{\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}}(\sigma)$ and $\mathrm{R}'_\sigma(P') \subseteq T'$. From the claim ($\star\star$), we get that there exists $\gamma \in \mathsf{Act}^{*}$ s.t. $Z^{-1}(P') \subseteq \mathrm{SE}^{\mathcal{M}}(\gamma)$ and $\mathrm{R}_\gamma(Z^{-1}(P')) \subseteq Z^{-1}(\mathrm{R}'_\sigma(P')) \subseteq Z^{-1}(T')$. Therefore, $Z^{-1}(P') \Rightarrow Z^{-1}(T')$ in $\mathcal{M}$ with witness $\gamma$, proving 1).

It is worth noticing that $\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}$ is a well-defined LTS, which is clear since $\mathcal{M} \in \mathbf{M_F}$. To prove that $\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}$ can be computed in polynomial space w.r.t. $\mathcal{M}$, it is enough to design a procedure iterating over every pair of $\mathfrak{C}_\mathcal{M}$-saturated subsets $P, T \subseteq \mathrm{S}$ and checking whether $P \Rightarrow T$. The latter can be decided by checking the non-emptiness of the language of the product FSA built from Defs. 10 and 11, which can be done in polynomial space as explained in Lemma 8. To ensure the desired complexity we should clear the space used by the procedure after each of the iterations. □

An important observation is that even though $\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}$ may have an exponential number of action names and edges, these are directly written into an *output* tape throughout the computation. Therefore, the used space in the *working* tape remains polynomially bounded w.r.t. the size of $\mathcal{M}$. Furthermore, the model obtained might contain an exponential number of edges, but it also makes explicit all the information required to do model-checking in a simpler way. Formally, define the problem $\texttt{MCKhSA} \stackrel{\text{def}}{=} \{\langle \mathcal{M}, s, \varphi\rangle \mid \mathcal{M}, w \models \varphi$ and if $\varphi = \mathsf{Kh}(\psi, \chi)$, its witness is some $a \in \mathsf{Act} \cup \{\epsilon\}\}$, a model-checking variant where each formula $\mathsf{Kh}(\psi, \chi)$ is witnessed by a single-action plan $a$ or $\epsilon$. This problem can be decided in $\mathsf{P}$, similar to what is done in (Demri and Fervari 2023, Th. 2). Thus, we can prove the following result:

**Theorem 5.** *Let $\mathcal{M}$, $\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}$ and $s$ as in Thm. 4 and $\varphi$ an $\mathsf{L_{Kh}}$-formula. Then, $\langle \mathcal{M}, s, \varphi\rangle \in \texttt{MCKh}$ iff $\langle \mathcal{M}_{/\mathcal{Z}_\mathcal{M}}, [s], \varphi\rangle \in \texttt{MCKhSA}$.*

The transition from a PSpace algorithm operating on the original model, to a P algorithm acting on a possible exponentially larger expansion $\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}$ represents a strategic shift from query complexity to data complexity. This approach is particularly beneficial when the cost of the expansion can be amortized over repeated verification tasks. By performing a one-time "offline" compilation, subsequent queries are reduced to simpler problems. This transformation might allow for the use of highly optimized, explicit-state model-checking tools that are often more robust and easier to parallelize than the recursive depth-first search strategies usually required for PSpace-complete reasoning.

We conclude by discussing another approach for model minimization. In particular, it is possible to contract our LTSs relying on the notion of bisimulation for BML (see, e.g., (Blackburn, de Rijke, and Venema 2002)). This is remarkable since BML-bisimulation between two models do not imply they satisfy the same $\mathsf{L_{Kh}}$-formulas. However, using BML-bisimulation works fine for auto-bisimulations.

**Definition 17.** *Let $\mathcal{M} = \langle \mathrm{S}, \mathsf{Act}, \mathrm{R}, \mathrm{V}\rangle$ be an* LTS*, and let $Z$ its maximal auto-bisimulation for BML. Its **BML contraction** is defined as $\mathcal{M}_{\text{BML}} = \langle \mathrm{S}', \mathsf{Act}, \mathrm{R}', \mathrm{V}'\rangle$, where:*

- $\mathrm{S}' \stackrel{\text{def}}{=} \mathrm{S}_{/Z}$, $\mathrm{V}'([s]) \stackrel{\text{def}}{=} \mathrm{V}(s)$.
- $\mathrm{R}'_a \stackrel{\text{def}}{=} \{([s],[t]) \mid$ *exist* $s' \in [s], t' \in [t]$ *s.t.* $(s', t') \in \mathrm{R}_a\}$.

The auto-bisimulation obtained through this notion is not maximal for $\mathsf{L_{Kh}}$, but still useful since efficient methods to compute it exist, see, e.g., (Paige and Tarjan 1987; Fernandez 1989).

**Theorem 6.** *Let $\mathcal{M}$ be an* LTS *with $\mathcal{M}_{\text{BML}}$ as in Def. 17. Then, for all states $s$, we have $\mathcal{M}, s \leftrightarrows^{*} \mathcal{M}_{\text{BML}}, [s]$.*

*Proof sketch.* Similar to (Areces, Fervari, and Mondejar 2026, Th. 5.5). □

$\mathcal{M}_{\text{BML}}$ is not the minimal contraction w.r.t. the number of states, but Thm. 6 enables us to use efficient methods to compute a model that is considerably smaller than the original one. For instance, (Paige and Tarjan 1987) establishes that for a finite $\mathcal{M}$, computing $\mathcal{M}_{\text{BML}}$ can be done in $\mathcal{O}(m{\cdot}\log(n))$ time, where $n$ and $m$ are respectively the number of states and edges of $\mathcal{M}$. We conclude with an example of contractions obtained via Defs. 16 and 17.

**Example 3.** *Let $\mathcal{M}$ be the* LTS *in Fig. 3. $\mathcal{M}_{/\mathcal{Z}_\mathcal{M}}$ is shown below $\mathcal{M}$, where $a_1 \stackrel{\text{def}}{=} a_{[s_1],[s_5]}$, $a_2 \stackrel{\text{def}}{=} a_{[s_1],[s_3]\cup[s_5]}$, $a_3 \stackrel{\text{def}}{=} a_{[s_1],[s_3]}$, $a_4 \stackrel{\text{def}}{=} a_{[s_3],[s_5]}$ and $a_5 \stackrel{\text{def}}{=} a_{[s_3],[s_1]\cup[s_5]}$. Finally, $\mathcal{M}_{\text{BML}}$ is displayed at the bottom of Fig. 3.*

## 5 Final Remarks

We studied computational aspects of bisimulations for the ability-based knowing-how logic over linear plans $\mathsf{L_{Kh}}$. Following the methodology applied to other variants, we redefined bisimulation using purely structural conditions, introducing $\mathsf{L^{*}_{Kh}}$-bisimulation, and proved adequacy for finite-valuation models. We showed that the model-comparison problem is PSpace-complete, a notable increase over uncertainty-based variants, which are coNP-complete. We also analyzed model minimization via $\mathsf{L_{Kh}}$ and BML auto-bisimulations, and discussed their properties.

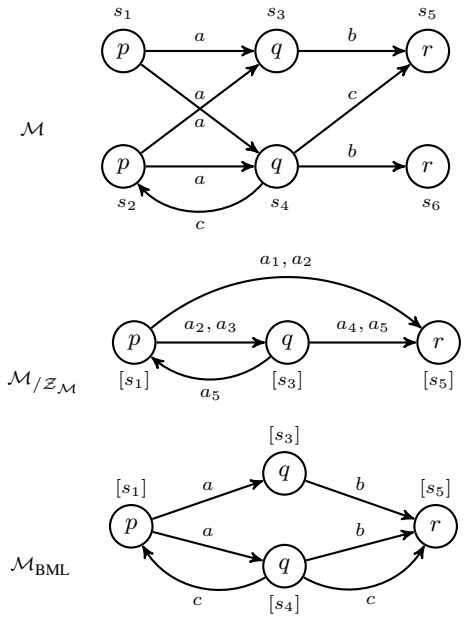


Figure 3: An LTS $\mathcal{M}$ and its corresponding contractions.

Future work includes characterizing the complexity of alternative behavioral equivalences tailored for linear-plan environments, such as simulation and trace equivalence, adapted to preserve *strong executability* and *goal-reachability*. It would be interesting also to investigate *goal-sensitive minimization* techniques that preserve planning-relevant metrics like optimal plan length. Finally, these structural methods could be extended to other ability-based logics, like probabilistic knowing-how (Castro, D'Argenio, and Fervari 2025) and knowing how with budgets (Demri, Doyen, and Fervari 2025). This will contribute to provide a unified framework for understanding model equivalence across the broad landscape of knowing-how logics.

## Acknowledgments

We would like to thank the reviewers for their comments and suggestions that helped us to improve the quality of the document. This work was partially supported by Agencia I+D+i grants PICT 2020-3780 and 2021-00400, the EU H2020 R&I programme under the MSC agreement 101008233 (MISSION), the IRP SINFIN, and SeCyT-UNC grant 33620230100178CB. Part of this work was carried out during a visit by Carlos Areces to GTIIT, China, whose support is gratefully acknowledged.

## AI Declaration

The authors have not employed any Generative AI tools.

## References

Abriola, S.; Barceló, P.; Figueira, D.; and Figueira, S. 2016. Bisimulations on data graphs. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016*, 309–318. AAAI Press.

Abriola, S.; Barceló, P.; Figueira, D.; and Figueira, S. 2018. Bisimulations on data graphs. *Journal of Artificial Intelligence Research* 61:171–213.

Areces, C.; Fervari, R.; Saravia, A. R.; and Velázquez-Quesada, F. R. 2021. Uncertainty-based semantics for multi-agent knowing how logics. In *Proceedings Eighteenth Conference on Theoretical Aspects of Rationality and Knowledge, TARK 2021*, volume 335 of *EPTCS*, 23–37.

Areces, C.; Fervari, R.; Saravia, A. R.; and Velázquez-Quesada, F. R. 2025. Uncertainty-based knowing how logic. *Journal of Logic and Computation* 35(1):1–35.

Areces, C.; Fervari, R.; and Hoffmann, G. 2015. Relation-changing modal operators. *Logic Journal of the IGPL* 23(4):601–627.

Areces, C.; Fervari, R.; and Mondejar, A. 2026. Computational aspects of plan-dependent model equivalence: The case of knowing-how bisimulations. In *The 25th International Conference on Autonomous Agents and Multi-Agent Systems*.

Areces, C.; Figueira, S.; and Gorín, D. 2011. Using logic in the generation of referring expressions. In *Logical Aspects of Computational Linguistics - 6th International Conference, LACL 2011*, volume 6736 of *LNCS*, 17–32. Springer.

Arenas, M., and Diaz, G. I. 2016. The exact complexity of the first-order logic definability problem. *ACM Transactions on Database Systems* 41(2):13:1–13:14.

Baier, C., and Katoen, J. 2008. *Principles of model checking*. MIT Press.

Barnaba, M. F., and Caro, F. D. 1985. Graded modalities. *Studia Logica* 44(2):197–221.

Belardinelli, F.; Condurache, R.; Dima, C.; Jamroga, W.; and Jones, A. V. 2017. Bisimulations for verifying strategic abilities with an application to threeballot. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017*, 1286–1295. ACM.

Belardinelli, F.; Condurache, R.; Dima, C.; Jamroga, W.; and Knapik, M. 2021. Bisimulations for verifying strategic abilities with an application to the threeballot voting protocol. *Information and Computation* 276:104552.

Belardinelli, F.; Dima, C.; and Murano, A. 2018. Bisimulations for logics of strategies: A study in expressiveness and verification. In *Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning, KR 2018*, 425–434. AAAI Press.

Blackburn, P.; de Rijke, M.; and Venema, Y. 2002. *Modal Logic*. Cambridge University Press.

Bolander, T.; Burigana, A.; and Montali, M. 2025. Depth-Bounded Epistemic Planning. In *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning*, 729–739. ijcai.org.

Castro, P. F.; D'Argenio, P. R.; and Fervari, R. 2025. How lucky are you to know your way? a probabilistic approach to knowing how logics. In *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning, KR 2025*, 229–239. ijcai.org.

de Rijke, M. 2000. A note on graded modal logic. *Studia Logica* 64(2):271–283.

Demri, S., and Fervari, R. 2023. Model-checking for ability-based logics with constrained plans. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*, 6305–6312. AAAI Press.

Demri, S.; Doyen, L.; and Fervari, R. 2025. Knowing-how reasoning with budgets recasted: Universal reachability problem on VASS. In *Reachability Problems - 19th International Conference, RP 2025*, LNCS, 140–155. Springer.

Fernandez, J. 1989. An implementation of an efficient algorithm for bisimulation equivalence. *Science of Computer Programming* 13(1):219–236.

Fervari, R.; Velázquez-Quesada, F. R.; and Wang, Y. 2022. Bisimulations for knowing how logics. *Review of Symbolic Logic* 15(2):450–486.

Figueira, D.; Figueira, S.; and Areces, C. 2015. Model theory of XPath on data trees. part I: bisimulation and characterization. *Journal of Artificial Intelligence Research* 53:271–314.

Galil, Z. 1976. Hierarchies of complete problems. *Acta Informatica* 6:77–88.

Goranko, V., and Passy, S. 1992. Using the universal modality: Gains and questions. *Journal of Logic and Computation* 2(1):5–30.

Hopcroft, J. E.; Motwani, R.; and Ullman, J. D. 2006. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc.

Kozen, D. 1977. Lower bounds for natural proof systems. In *FOCS'77*, 254–266. IEEE Computer Society.

Milner, R. 1971. An algebraic definition of simulation between programs. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence*, IJCAI'71, 481–489. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Milner, R. 1989. *Communication and Concurrency*. Prentice Hall.

Nissim, R.; Hoffmann, J.; and Helmert, M. 2011. Computing perfect heuristics in polynomial time: On bisimulation and merge-and-shrink abstraction in optimal planning. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011*, 1983–1990. IJCAI/AAAI.

Paige, R., and Tarjan, R. E. 1987. Three partition refinement algorithms. *SIAM Journal on Computing* 16(6):973–989.

Park, D. 1981. Concurrency and automata on infinite sequences. In Deussen, P., ed., *Theoretical Computer Science*, 167–183. Berlin, Heidelberg: Springer Berlin Heidelberg.

Sangiorgi, D., and Rutten, J. J. M. M., eds. 2012. *Advanced Topics in Bisimulation and Coinduction*, volume 52 of *Cambridge tracts in theoretical computer science*. Cambridge University Press.

Savitch, W. 1970. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences* 4(2):177–192.

van Benthem, J. 1976. *Modal Correspondence Theory*. Ph.D. Dissertation, University of Amsterdam. Department of Mathematics.

Wang, Y. 2015. A logic of knowing how. In *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015*, volume 9394 of *LNCS*, 392–405. Berlin, Heidelberg: Springer.

Wang, Y. 2018. A logic of goal-directed knowing how. *Synthese* 195(10):4419–4439.