Computational Aspects of Plan-Dependent Model Equivalence: The Case of Knowing-How Bisimulations

Carlos Areces
CONICET and FAMAF-UNC
Córdoba, Argentina
carlos.areces@unc.edu.ar

Raul Fervari
CONICET and FAMAF-UNC
Córdoba, Argentina
Université Paris-Saclay, LMF
Gif-Sur-Yvette, France
rfervari@unc.edu.ar

Antonio Mondejar
FAMAF-UNC
Córdoba, Argentina
antonio.mondejar@mi.unc.edu.ar

ABSTRACT

We investigate computational aspects of model equivalence for labeled transition systems (LTSs) relative to sets of plans. We focus on the case when model equivalence is defined by a notion of bisimulation in the context of an uncertainty-based knowing-how logic. We start by reformulating such a notion of bisimulation based on purely structural conditions of the LTSs involved, and show adequacy results. Then, we elaborate a computational profile of the new bisimulations. First, we show that the problem of checking whether two LTSs are bisimilar is coNP-complete. Then, we investigate model contractions in order to obtain minimal LTSs, and show that they can be computed in polynomial time.

KEYWORDS

Model Comparison; Knowing How; Bisimulation; Complexity; Plans

ACM Reference Format:

Carlos Areces, Raul Fervari, and Antonio Mondejar. 2026. Computational Aspects of Plan-Dependent Model Equivalence: The Case of Knowing-How Bisimulations. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/OKUJ9671

1 INTRODUCTION

Labeled transition systems (LTSs) are the de facto formalism for describing the behavior of a system. An LTS focuses on how the execution of an action transforms one state of the system into another. These structures are central in different areas of computer science, such as concurrency theory, process algebra, and modal logics (including temporal and epistemic variants), among others.

In general, LTSs act as abstractions of actual systems in order to analyze their properties. For instance, they play a key role in model-checking, where a logical formula expressing some desired behavior of the system is verified over the LTS that represents it (see, e.g., [9]). As a result, the problem of determining whether two system representations have an equivalent behavior becomes crucial. From a computational or cognitive perspective, for example, working with *minimal* representations of equivalent systems is often desirable, once a suitable definition of "equivalence" is fixed.

Most current approaches introduce structural notions of model equivalence, i.e., notions that are based solely on intrinsic characteristic of the models. In this regard, the standard notion of *bisimulation* [26, 27, 30, 31] was introduced, inspired by the classical notion of *automata equivalence* (see, e.g., [24]). In short, a bisimulation is a relation Z between states of two LTSs satisfying certain structural conditions, namely that if two states s and s' are related by Z (i.e., sZs'), and it is possible to perform a step from s to another state t via an action a, it is also possible to perform a step via a from s' to another state t' such that tZt', and vice-versa. For the case of LTSs where states are also labeled, it is additionally required that states related by Z share the same labeling. Intuitively, bisimulations capture the idea that, two LTSs represent equivalent systems, if every state of an LTS can be *simulated* by a state in the other system, in the sense that the execution of actions in one system can be mimicked by the other.

The origin of bisimulations can be traced back to a collection of works emerging from different areas (see, e.g., [31] for a discussion), ranging from algebraic theory of automata [22], to program/process comparison [26, 30]. Moreover, bisimulations have become a standard tool to characterize the expressivity of modal languages (see [13, 34]). In this context, it was proved that bisimilar models satisfy the same modal formulas (in what is usually known as the basic modal language), thus guaranteeing that they exhibit the same "modal behavior". Remarkably, this notion of equivalence was introduced even earlier with a more operational definition, by means of what are now called Ehrenfeucht-Fraïssé games [17, 21].

In modern modal logic, there exists a vast number of modal languages with different levels of expressivity, each requiring its own notion of bisimulation (see [13]). Our focus, in this article, is to investigate the computational properties of bisimulation for *knowing-how* logics. Knowing-how is an epistemic concept motivated by scenarios in AI related to strategic abilities and planning. The proposal in [36, 37] constitutes the starting point of a series of works addressing different aspects of knowing-how. Within this approach, we say that an agent *knows how to achieve a goal ψ from an initial condition φ*, if there exists a sequence of actions σ (a *plan*) that can be executed to completion without ever aborting (a requirement usually referred to as *strong executability*) from every state satisfying φ, leading unerringly to states satisfying ψ. In [3, 4], this definition is extended so that an LTS represents the available courses of action for the agents, while each agent's perception is modeled by an indistinguishability relation between plans. Thus, an agent knows how to achieve a goal from a given initial condition if a *set of indistinguishable plans* meet the conditions described above.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/OKUJ9671

The obtained logic is called *uncertainty-based knowing how logic* and denoted $\mathsf{L}_{\mathsf{Kh}_i}$. Notice that the practical or procedural knowledge (ability-based and skill-oriented) of an agent is now *relativized* to sets of plans, that encode her perception about her own abilities/skills (see Def. 2.6). Then, in this article we focus on a notion of model equivalence through bisimulations in this context, where equivalence is not only defined with respect to the structural characteristics of the LTS, but it is also relative to a set of plans or a notion of plan equivalence that is part of the input.

Some proposals of "plan-dependent" bisimulations can be found in previous literature. This is the case, for instance, of *stuttering bisimulation* [15]. Stuttering bisimulations define a notion of model equivalence in which certain intermediate states along a plan can be ignored. What matters for equivalence is the emerging, observable behavior along a plan, disregarding possible repetitions of a given state. Another example is bisimulation for strategy logic [12], useful to determine the ability of an agent to commit to a strategy. In this setting, bisimulations are defined relative to strategies and histories (not just states). In [10, 11], model comparison via bisimulations is explored, considering objective and subjective semantics for ATL^*. While in the former the notion of equivalence relies on the available actions of the LTS, the latter takes into account each agent's information about the system, just as we will do. In [28], bisimulations are used to compute heuristics for automated planning. Therein, model minimization is performed with respect to equivalent paths.

Contributions. Our first contribution is to reformulate the definition of bisimulations for $\mathsf{L}_{\mathsf{Kh}_i}$, in terms of structural conditions. An adequate notion of bisimulation for $\mathsf{L}_{\mathsf{Kh}_i}$ was first introduced in [4], building on the work of [20]. A distinctive feature of these notions is the presence of a syntactic clause, stating that for two models to be bisimilar, plans executed at *propositionally definable* sets in one model should be mimicked in the other model. A set is propositionally definable in a model, if it coincides with the extension of a propositional formula. In this article, we recast the "propositional definability" condition into a clause that depends simply on the valuation function of the model, and we establish its adequacy. We call the new notion $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation, and show that $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimilar models satisfy the same $\mathsf{L}_{\mathsf{Kh}_i}$-formulas, while the converse holds for LTSs with finite sets of states. Interestingly, although LTSs that are $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimilar are also bisimilar under the definition from [4], the converse does not hold in general. However, it holds for finite-state LTSs, which is sufficient for our purposes.

The notion of $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation has some particularities. First, the notion is global, i.e., conditions are established for all states in the model. Second, plan execution is checked with respect to a set of states, instead of a single state. Third, the plans executed on each model can differ, e.g., they can have different lengths, with the status of intermediate states being ignored. These plans, besides modeling the agent's perception, can be abstractly understood as those plans that are known to be more likely to succeed. For instance, this could be the case of a robot that has to traverse a surface, following sets of plans derived from previously recorded paths. Then, these plans could be part of the input in a model-checking procedure.

The structural nature of $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation is useful for computational analysis. We characterize the complexity of the model comparison problem under $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation. In particular, we prove that checking whether two models are $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimilar is coNP-complete. For the upper bound, we provide a non-deterministic algorithm that verifies counter-examples, while the lower bound is obtained by a reduction from the problem of deciding whether a propositional formula in disjunctive normal form is a tautology (see [7]). The algorithm also enables us to build a polynomial size formula that distinguishes the two models, if one exists.

Finally, we investigate the problem of model minimization under $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation and show that minimal models can be obtained via bisimulation-based contractions. It has been established in [3, 4] that the model-checking problem for $\mathsf{L}_{\mathsf{Kh}_i}$ can be solved in polynomial time, with respect to the number of states and the number of plans that the agent is aware of. In this article, we study different ways of contracting a model while minimizing these measures. Crucially, to obtain minimal models the relevant paths between certain states are abstracted into one-step actions, while preserving their meaning with respect to $\mathsf{L}_{\mathsf{Kh}_i}$. Intermediate states can be safely ignored, similar to the so-called *path abstraction* technique [1, 32, 33], introduced to make model-checking algorithms more tractable.

Outline. In Sec. 2 we introduce the syntax and semantics of the uncertainty-based knowing how logic $\mathsf{L}_{\mathsf{Kh}_i}$, together with the notion of bisimulation from [4]. In Sec. 3 we reformulate the clauses of bisimulation, show a correspondence with the previous definition, and prove adequacy results. Moreover, we investigate the definability problem in $\mathsf{L}_{\mathsf{Kh}_i}$ as a corollary. Sec. 4 is devoted to prove that the problem of deciding if two models are $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimilar is coNP-complete. Moreover, our algorithm enables to extract a formula distinguising the models, if one exists. Then, in Sec. 5 we investigate the problem of model minimization, for which we propose two approaches: one based on $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation and another based on standard bisimulations. In Sec. 6 we provide some final remarks.

2 PRELIMINARIES

2.1 Syntax and Semantics

We start by presenting the logical framework introduced in [3, 4]. In the rest of the article, let Prop be a countable set of propositional symbols and Agt a finite set of agents.

Definition 2.1. Formulas of the **language** $\mathsf{L}_{\mathsf{Kh}_i}$ are defined by the following grammar:

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

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

In [36, 37], 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. This is extended in [3, 4] to include a notion of *uncertainty* or *indistinguishability* between plans (a.k.a. the agents *perceptions*).

Definition 2.2 (Actions and plans). Let Act^* be the set of finite sequences over a set of actions Act. Elements of Act^* are called **plans**, with ϵ being the **empty plan**. Given $\sigma \in \mathsf{Act}^*$, we use $|\sigma|$ to denote its length (with $|\epsilon| = 0$). For $0 \le k \le |\sigma|$, the plan σ_k is σ's initial segment up to (and including) the kth position (with $\sigma_0 \overset{\text{def}}{=} \epsilon$). For $0 < k \le |\sigma|$, the action $\sigma[k]$ is the one in σ's kth position.

Definition 2.3 (Uncertainty-based LTS*).* An **uncertainty-based LTS (LTS^{U})** for Prop and Agt is a tuple $\mathcal{M} = \langle \text{S}, \text{Act}, \text{R}, \text{U}, \text{V} \rangle$ where S is a non-empty set of states, Act is a countable non-empty set of action names, $\text{R} = \{\text{R}_a \subseteq \text{S} \times \text{S} \mid a \in \text{Act}\}$ is a collection of binary relations on S, $\text{U} = \{\text{U}(i) \subseteq \mathcal{P}(\text{Act}^*) \setminus \{\emptyset\} \mid i \in \text{Agt}\}$ assigns to every agent a non-empty collection of pairwise disjoint non-empty sets of plans (i.e., $\text{U}(i) \neq \emptyset$, $\pi_1, \pi_2 \in \text{U}(i)$ with $\pi_1 \neq \pi_2$ implies $\pi_1 \cap \pi_2 = \emptyset$, and $\emptyset \notin \text{U}(i)$) and $\text{V} : \text{S} \to \mathcal{P}(\text{Prop})$ 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 Act, each $\text{V}(s)$, $\text{U}(i)$, and all $\pi \in \text{U}(i)$ are finite. If $s \in \text{S}$, we call $\mathcal{M}, s$ a *pointed* LTS^{U}.

Intuitively, $\text{P}_i \overset{\text{def}}{=} \bigcup_{\pi \in \text{U}(i)} \pi$ is the set of plans agent i has at her disposal (alternatively, she is aware of). Each $\pi \in \text{U}(i)$ is an indistinguishability class: two plans in π are seen as equivalent by the agent (see [3, 4] for a discussion). As discussed therein, there is a one-to-one correspondence between each $\text{U}(i)$ and an 'indistinguishability relation' $\sim_i \subseteq \text{P}_i \times \text{P}_i$ describing the agent's *uncertainty* or *perception* over her available plans ($\sigma_1 \sim_i \sigma_2$ iff there is $\pi \in \text{U}(i)$ such that $\{\sigma_1, \sigma_2\} \subseteq \pi$). The presentation used here simplifies the definitions that will follow.

Definition 2.4. Given $\text{R} = \{\text{R}_a \subseteq \text{S} \times \text{S} \mid a \in \text{Act}\}$ and $\sigma \in \text{Act}^*$, define $\text{R}_\sigma \subseteq \text{S} \times \text{S}$ inductively as: $\text{R}_\epsilon \overset{\text{def}}{=} \{(s, s) \mid s \in \text{S}\}$ and $\text{R}_{\sigma a} \overset{\text{def}}{=} \text{R}_\sigma \circ \text{R}_a$. For $\pi \subseteq \text{Act}^*$ and $T \cup \{t\} \subseteq \text{S}$, define $\text{R}_\sigma(t) \overset{\text{def}}{=} \{u \in \text{S} \mid (t, u) \in \text{R}_\sigma\}$, and $\text{R}_\sigma(T) \overset{\text{def}}{=} \bigcup_{u \in T} \text{R}_\sigma(u)$. Moreover, we denote $\text{R}_\pi \overset{\text{def}}{=} \bigcup_{\sigma \in \pi} \text{R}_\sigma$, $\text{R}_\pi(t) \overset{\text{def}}{=} \bigcup_{\sigma \in \pi} \text{R}_\sigma(t)$, and $\text{R}_\pi(T) \overset{\text{def}}{=} \bigcup_{t \in T} \text{R}_\pi(t)$.

In what follows, we introduce the notion of strong executability of plans (see, e.g., [4, 36]), a condition which determines when a given plan (or a set of them) is appropriate for achieving a certain goal. Given her uncertainty over (a subset of) Act^*, the abilities of an agent i depend not on what a single plan can achieve, but rather on what a set of them can guarantee.

Definition 2.5 (Strong executability). Let $\mathcal{M} = \langle \text{S}, \text{Act}, \text{R}, \text{U}, \text{V} \rangle$ be an LTS^{U}. A plan $\sigma \in \text{Act}^*$ is **strongly executable** (SE) at $s \in \text{S}$ if and only if, for all $k \in [0 \dots |\sigma|-1]$, $t \in \text{R}_{\sigma_k}(s)$ implies $\text{R}_{\sigma[k+1]}(t) \neq \emptyset$. We define the set $\text{SE}^{\mathcal{M}}(\sigma) \overset{\text{def}}{=} \{s \in \text{S} \mid \sigma \text{ is SE at } s\}$. Then, a *set of plans* $\pi \subseteq \text{Act}^*$ is *strongly executable* at s if and only if *every* plan $\sigma \in \pi$ is *strongly executable* at s. Finally, $\text{SE}^{\mathcal{M}}(\pi) \overset{\text{def}}{=} \bigcap_{\sigma \in \pi} \text{SE}^{\mathcal{M}}(\sigma)$ is the set of the states in S where π is SE.

In short, *a plan* is strongly executable (at a state) when *all its partial executions* can be completed, while *a set of plans* is strongly executable when *all its plans* are strongly executable.

Let us introduce now the semantics of the logic.

Definition 2.6. Let $\mathcal{M} = \langle \text{S}, \text{Act}, \text{R}, \text{U}, \text{V} \rangle$ be an LTS^{U} and $s \in \text{S}$. The **satisfiability relation** $\models$ for $\mathsf{L}_{\mathsf{Kh}_i}$ is inductively defined as:

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

where $[\![\chi]\!]^{\mathcal{M}} \overset{\text{def}}{=} \{t \in \text{S} \mid \mathcal{M}, t \models \chi\}$. Any π making true the existential statement in the semantic clause of $\mathsf{Kh}_i(\varphi, \psi)$ is called a **witness** for $\mathsf{Kh}_i(\varphi, \psi)$. The universal modality (see [23]) can be expressed in $\mathsf{L}_{\mathsf{Kh}_i}$ as $\mathsf{A}\varphi \overset{\text{def}}{=} \mathsf{Kh}_i(\neg\varphi, \bot)$ (for i arbitrary, see [4]), and its dual as $\mathsf{E}\varphi \overset{\text{def}}{=} \neg\mathsf{A}\neg\varphi$. Notice that $\mathcal{M}, s \models \mathsf{A}\varphi$ iff $[\![\varphi]\!]^{\mathcal{M}} = \text{S}$.

2.2 A Notion of Bisimulation for $\mathsf{L}_{\mathsf{Kh}_i}$

The notion of *bisimulation* is a crucial tool to compare models, and moreover to understand the expressive power of a modal language. Here we review the notion of bisimulation for $\mathsf{L}_{\mathsf{Kh}_i}$ over LTS^{U}s introduced in [4]. We start by providing some useful abbreviations.

Definition 2.7. Let $\mathcal{M} = \langle \text{S}, \text{Act}, \text{R}, \text{U}, \text{V} \rangle$ be an LTS^{U} over Prop and Agt. Take $\pi \in \mathcal{P}(\text{Act}^*)$, $X, T \subseteq \text{S}$ and $i \in \text{Agt}$.

- Write $X \overset{\pi}{\Rightarrow} T \overset{\text{def}}{\Leftrightarrow} X \subseteq \text{SE}^{\mathcal{M}}(\pi)$ and $\text{R}_\pi(X) \subseteq T$.
- Write $X \overset{i}{\Rightarrow} T \overset{\text{def}}{\Leftrightarrow}$ there is $\pi \in \text{U}(i)$ such that $X \overset{\pi}{\Rightarrow} T$.

Additionally, $X \subseteq \text{S}$ is propositionally definable in $\mathcal{M}$ if and only if there is a propositional formula φ such that $X = [\![\varphi]\!]^{\mathcal{M}}$.

Now we introduce the notion of bisimulation for $\mathsf{L}_{\mathsf{Kh}_i}$.

Definition 2.8 ($\mathsf{L}_{\mathsf{Kh}_i}$-bisimulation). Let $\mathcal{M} = \langle \text{S}, \text{Act}, \text{R}, \text{U}, \text{V} \rangle$ and $\mathcal{M}' = \langle \text{S}', \text{Act}', \text{R}', \text{U}', \text{V}' \rangle$ be $\text{LTS}^{\text{U}}s$. A non-empty $Z \subseteq \text{S} \times \text{S}'$ is called an $\mathsf{L}_{\mathsf{Kh}_i}$**-bisimulation** between $\mathcal{M}$ and $\mathcal{M}'$ if and only if sZs' implies all of the following.

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

(Kh_i-zig): for any *propositionally* definable $P \subseteq \text{S}$, if $P \overset{i}{\Rightarrow} T$ for some $T \subseteq \text{S}$, then there is $T' \subseteq \text{S}'$ such that:
1) $Z(P) \overset{i}{\Rightarrow} T'$, and 2) $T' \subseteq Z(T)$.

(Kh_i-zag): for any *propositionally* definable $P' \subseteq \text{S}'$, if $P' \overset{i}{\Rightarrow} T'$ for some $T' \subseteq \text{S}'$, then there is $T \subseteq \text{S}$ such that:
1) $Z^{-1}(P') \overset{i}{\Rightarrow} T$, and 2) $T \subseteq Z^{-1}(T')$.

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

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

(Notice that above, $Z(X)$ and $Z^{-1}(X')$ have the expected meaning.)

We write $\mathcal{M}, s \leftrightarroweq \mathcal{M}', s'$ when there is an $\mathsf{L}_{\mathsf{Kh}_i}$-bisimulation Z between $\mathcal{M}$ and $\mathcal{M}'$ such that sZs'.

The following theorem establishes a classical adequacy result.

Theorem 2.9 ([4]). *Let $\mathcal{M}, s$ and $\mathcal{M}', s'$ be two $\text{LTS}^{\text{U}}s$. $\mathcal{M}, s \leftrightarroweq \mathcal{M}', s'$ implies $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}', s' \models \varphi$, for all $\mathsf{L}_{\mathsf{Kh}_i}$-formulas φ. Moreover, if $\mathcal{M}$ and $\mathcal{M}'$ are finite-domain, the converse also holds.*

Thm. 2.9 states that the notion of bisimulation we introduced is in a sense adequate for the logic. However, one can argue that the conditions (Kh_i-zig/zag) can be target of criticism. Usually, a notion of bisimulation relies purely on structural properties of the logic. However, Def. 2.8 contains clauses quantifying over *propositionally definable* sets, i.e., sets that are characterized by a propositional formula (a syntactic object). This also makes difficult to provide a procedural way to operate over bisimulations. In the next section we introduce a novel notion of bisimulation addressing these issues.

3 REDEFINING BISIMULATIONS FOR $\mathsf{L}_{\mathsf{Kh}_i}$

In this section we introduce a novel notion of bisimulation that maintains the logical properties of Def. 2.8. Interestingly, this new notion relies on structural characteristics only and it is more suitable for defining model-comparison procedures. We start by introducing a natural property that will be of use in the forthcoming results.

Lemma 3.1. *Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ be an* $\mathrm{LTS}^{\mathrm{U}}$*, and $s, t \in \mathrm{S}$ be such that $\mathrm{V}(s) = \mathrm{V}(t)$. Then, for all propositional formulas φ, $\mathcal{M}, s \models \varphi$ 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 3.2. Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ be an $\mathrm{LTS}^{\mathrm{U}}$, define the relation $\mathcal{Z}_{\mathcal{M}} \overset{\text{def}}{=} \{(s,t) \in \mathrm{S} \times \mathrm{S} \mid \mathrm{V}(s) = \mathrm{V}(t)\}$. Let $[s] \overset{\text{def}}{=} \{t \in \mathrm{S} \mid (s,t) \in \mathcal{Z}_{\mathcal{M}}\}$, define $\mathfrak{C}_{\mathcal{M}} \overset{\text{def}}{=} \{[s] \mid s \in \mathrm{S}\}$.

We now reformulate the property of "propositionally definability" in terms of structural properties of the model. Consider first the following result:

Lemma 3.3. *Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ be an* $\mathrm{LTS}^{\mathrm{U}}$*, with $P \subseteq \mathrm{S}$ a propositionally definable set. Then, for all $[s] \in \mathfrak{C}_{\mathcal{M}}$, either $[s] \cap P = \emptyset$ or $[s] \subseteq P$. If $\mathcal{M}$ is finite-domain, the converse also holds.*

Proof. Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ be an $\mathrm{LTS}^{\mathrm{U}}$, and let $P \subseteq \mathrm{S}$ be a propositionally definable set, i.e., there exists a propositional formula ψ s.t. $[\![\psi]\!]^{\mathcal{M}} = P$. To prove the lemma, we assume for all $[s] \in \mathfrak{C}_{\mathcal{M}}$, $[s] \cap P \neq \emptyset$ 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 3.1, $\mathcal{M}, u \models \psi$. Then, $u \in P$, and therefore we obtain $[s] \subseteq P$.

For the converse, assume $\mathcal{M}$ is finite-domain, i.e., S is finite. Suppose that for all $[s] \in \mathfrak{C}_{\mathcal{M}}$, either $[s] \cap P = \emptyset$ or $[s] \subseteq P$. We proceed by case analysis. If $P = \emptyset$, the formula $\psi = \bot$ defines P, thus the proof is concluded. Then assume $P \neq \emptyset$. Notice that since S is finite, each $[s]$ is finite, and moreover, $\mathfrak{C}_{\mathcal{M}}$ contains a finite number of elements. Thus, let us denote $\mathfrak{C}_{\mathcal{M}}$ as $\{[s_1], \ldots, [s_n]\}$, and denote $\mathrm{V}([s])$ as $\mathrm{V}(s)$ (since for all $u, v \in [s]$, $\mathrm{V}(u) = \mathrm{V}(v)$). Then, for each $j \neq k$, there is $p \in \mathrm{Prop}$ 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 $\psi_j \overset{\text{def}}{=} \bigwedge_{k=1, k\neq j}^{n} l(p_{j,k})$, 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 $\mathsf{L}_{\mathsf{Kh}_i}$ with the desired characteristics.

Definition 3.4 ($\mathsf{L}^{*}_{\mathsf{Kh}_i}$-bisimulation). Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ and $\mathcal{M}' = \langle \mathrm{S}', \mathrm{Act}', \mathrm{R}', \mathrm{U}', \mathrm{V}'\rangle$ be $\mathrm{LTS}^{\mathrm{U}}$s. A non-empty $Z \subseteq \mathrm{S} \times \mathrm{S}'$ is called an **$\mathsf{L}^{*}_{\mathsf{Kh}_i}$-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. 2.8.

(Kh_i-zig*): for all $P \subseteq \mathrm{S}$ such that for any $[t] \in \mathfrak{C}_{\mathcal{M}}$ either $[t] \cap P = \emptyset$ or $[t] \subseteq P$, if $P \overset{i}{\Rightarrow} T$ for some $T \subseteq \mathrm{S}$, then there is $T' \subseteq \mathrm{S}'$ such that: 1) $Z(P) \overset{i}{\Rightarrow} T'$, and 2) $T' \subseteq Z(T)$.

(Kh_i-zag*): for all $P' \subseteq \mathrm{S}'$ such that for any $[t'] \in \mathfrak{C}_{\mathcal{M}'}$ either $[t]' \cap P' = \emptyset$ or $[t'] \subseteq P'$, if $P' \overset{i}{\Rightarrow} T'$ for some $T' \subseteq \mathrm{S}'$, then there is $T \subseteq \mathrm{S}$ such that:
1) $Z^{-1}(P') \overset{i}{\Rightarrow} T$, and 2) $T \subseteq Z^{-1}(T')$.

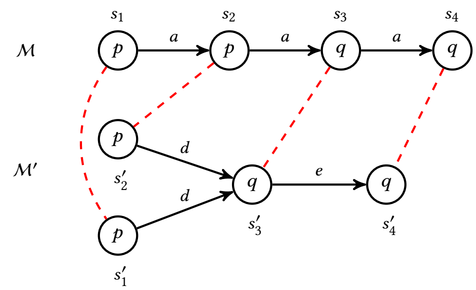

Figure 1: Two $\mathsf{L}^{*}_{\mathsf{Kh}_i}$-bisimilar models.

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

Below we state a correspondence between Defs. 2.8 and 3.4.

Lemma 3.5. *If a relation Z is an $\mathsf{L}^{*}_{\mathsf{Kh}_i}$-bisimulation between two* $\mathrm{LTS}^{\mathrm{U}}$*s $\mathcal{M}$ and $\mathcal{M}'$, then it is an $\mathsf{L}_{\mathsf{Kh}_i}$-bisimulation between them. If $\mathcal{M}$ and $\mathcal{M}'$ are finite-domain, then the converse also holds.*

Proof. Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ and $\mathcal{M}' = \langle \mathrm{S}', \mathrm{Act}', \mathrm{R}', \mathrm{U}', \mathrm{V}'\rangle$ be two $\mathrm{LTS}^{\mathrm{U}}$s, and let $Z \subseteq \mathrm{S} \times \mathrm{S}'$ be an $\mathsf{L}^{*}_{\mathsf{Kh}_i}$-bisimulation. To prove the property it is enough to see that if Z satisfies (Kh_i-zig*) (resp. zag*), then it satisfies (Kh_i-zig) (resp. zag). Let $P \subseteq \mathrm{S}$ be a propositionally definable set s.t. for some $T \subseteq \mathrm{S}$, we have $P \overset{i}{\Rightarrow} T$. We need to find a set $T' \subseteq \mathrm{S}'$ s.t. 1) $Z(P) \overset{i}{\Rightarrow} T'$, and 2) $T' \subseteq Z(T)$. Since P is propositionally definable, by Lemma 3.3, for all $[s] \in \mathfrak{C}_{\mathcal{M}}$, either $[s] \cap P = \emptyset$ or $[s] \subseteq P$. Moreover, Z satisfies (Kh_i-zig*), which guarantees the existence of the set T' as described above. For (Kh_i-zag*) the proof is analogous. The converse for the finite-domain case follows by the converse part in Lemma 3.3. □

As a consequence of the previous lemma, we are able to prove the corresponding adequacy results for $\mathsf{L}^{*}_{\mathsf{Kh}_i}$-bisimulations.

Theorem 3.6. *Let $\mathcal{M}, s$ and $\mathcal{M}', s'$ be two* $\mathrm{LTS}^{\mathrm{U}}$*s. $\mathcal{M}, s \underline{\leftrightarrow}^{*} \mathcal{M}', s'$ implies $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}', s' \models \varphi$, for all $\mathsf{L}_{\mathsf{Kh}_i}$-formulas φ. If $\mathcal{M}$ and $\mathcal{M}'$ are finite-domain, the converse also holds.*

Proof. It follows by Lemma 3.5 and Thm. 2.9. □

As an example, let $\mathcal{M}$ and $\mathcal{M}'$ be two $\mathrm{LTS}^{\mathrm{U}}$s with $\mathrm{Agt} = \{i\}$, whose graphical representation is given in Fig. 1, where $\mathrm{U}(i) = \{\{aa\}\}$ and $\mathrm{U}'(i) = \{\{de, d\}\}$. It is easy to check that $\mathcal{M}, s_1 \underline{\leftrightarrow}^{*} \mathcal{M}', s'_1$, since Z defined by the dashed arrows is an $\mathsf{L}^{*}_{\mathsf{Kh}_i}$-bisimulation. For instance, we can check that $[s_1] = \{s_1, s_2\}$ and that $[s_1] \overset{i}{\Rightarrow} [s_3] = \{s_3, s_4\}$, with witness $\{aa\}$. We can take $T' = [s'_3] = \{s'_3, s'_4\}$, since it satisfies 1) $Z([s_1]) \overset{i}{\Rightarrow} T'$ (witness $\{de, d\}$), and 2) $T' \subseteq Z([s_3])$ (actually, $T' = [s'_3] = Z([s_3])$).

It is worth noticing that an $\mathsf{L}_{\mathsf{Kh}_i}$-bisimulation Z is not necessarily an $\mathsf{L}^{*}_{\mathsf{Kh}_i}$-bisimulation (although this is the case in finite models). Suppose that Prop is an infinite and countable set of propositional symbols. Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ be an $\mathrm{LTS}^{\mathrm{U}}$ s.t. $\mathrm{S} = \mathcal{P}(\mathrm{Prop})$ and $\mathrm{V}(s) = s$. Clearly, $\mathcal{P}(\mathrm{Prop})$ is not countable. However, the set of all the propositional formulas built over Prop is countable, thus there is some $s \in \mathcal{P}(\text{Prop})$ such that $\{s\}$ is not propositionally definable. Hence, the converse of Lemma 3.3 in the general case does not hold.

With this at hand, we can define $\mathcal{M}$ and $\mathcal{M}'$ such that they are $\mathsf{L}_{\mathsf{Kh}_i}$-bisimilar but not $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimilar. Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ and $\mathcal{M}' = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}', \mathrm{V}\rangle$ be s.t. $\mathrm{S} = \mathcal{P}(\text{Prop})$, $\mathrm{Act} = \{a, b\}$, $\mathrm{V}(s) = s$, $\mathrm{U}(i) = \{\{a\}, \{b\}\}$ and $\mathrm{U}'(i) = \{\{a\}\}$. Define also $\mathrm{R}_a = \mathrm{S} \times \mathrm{S}$, and, for a fix s s.t. $\{s\}$ is not propositionally definable, $\mathrm{R}_b = \{(s, s)\}$. Take $P = \{s\}$ as in Def. 3.4, then $\{s\} \overset{i}{\Rightarrow} \{s\}$ via $\{b\}$ in $\mathcal{M}$. However, there is no matching $\pi \in \mathrm{U}'(i)$, since the only possible witness in $\mathcal{M}'$ is $\{a\}$, but $\mathrm{R}_a(\{s\}) \nsubseteq \mathrm{R}_b(\{s\})$. Thus, $\mathcal{M}, s \leftrightarrows \mathcal{M}', s$ but $\mathcal{M}, s \leftrightarrows^* \mathcal{M}', s$ does not hold. This situation is natural since $\mathsf{L}_{\mathsf{Kh}_i}$-bisimulations involve a finite characterization of sets via propositional formulas, while $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulations get rid of such a condition.

We conclude this section with a result that already illustrates the impact of our developments. Concretely, we show that with the results introduced above we can solve the *definability problem* (see, e.g., [5, 6] for examples of its use) for $\mathsf{L}_{\mathsf{Kh}_i}$ in polynomial time.

Corollary 3.7 (Definability). *Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ be a finite-domain* $\mathrm{LTS}^{\mathrm{U}}$*, and let $X \subseteq \mathrm{S}$. The problem of deciding whether there exists an $\mathsf{L}_{\mathsf{Kh}_i}$-formula φ such that $[\![\varphi]\!]^{\mathcal{M}} = X$ is in* P.

Proof. In [4, Prop. 3] it is established that $[\![\varphi]\!]^{\mathcal{M}} = X$ for some $\mathsf{L}_{\mathsf{Kh}_i}$-formula φ iff $[\![\psi]\!]^{\mathcal{M}} = X$, for some propositional formula ψ. Then, by Lemma 3.3, the problem boils down to computing $\mathfrak{C}_{\mathcal{M}}$, and then checking whether $X \cap [s] = \emptyset$ or $[s] \subseteq X$, for each $[s] \in \mathfrak{C}_{\mathcal{M}}$. This computation can be clearly done in polynomial time. □

4 COMPLEXITY OF MODEL-COMPARISON

This section is devoted to a computational analysis of bisimulations in the context of $\mathsf{L}_{\mathsf{Kh}_i}$. These bisimulations depend on the sets of plans specified on $\mathrm{U}(i)$. The main problem we address here consists in checking whether two *finite* models are bisimilar:

Definition 4.1. Define the problem `KhiBisim` as:

$$\texttt{KhiBisim} \overset{\text{def}}{=} \{\langle \mathcal{M}, \mathcal{M}'\rangle \mid \mathcal{M}, \mathcal{M}' \text{ are finite LTS}^{\mathrm{U}}\text{s and } \mathcal{M} \leftrightarrows^* \mathcal{M}'\}.$$

Before we proceed to characterize the complexity of `KhiBisim`, we will discuss some subproblems that are of interest for their own sake. Concretely, `KhiBisim` can be thought of as a "search and verification problem", consisting on finding a potential candidate Z for a bisimulation and checking that Z actually is a bisimulation. Remarkably, the search part is not actually needed, as we can use a very particular relation that works universally. We postpone the definition of such a relation, and start providing an algorithm and a complexity characterization of the problem of checking if a given Z is a bisimulation. Below, $\mathcal{M}$ and $\mathcal{M}'$ are always finite.

Definition 4.2. Define the problem `CheckKhiBisim` as:

$$\texttt{CheckKhiBisim} \overset{\text{def}}{=} \{\langle \mathcal{M}, \mathcal{M}', Z\rangle \mid Z \text{ is an } \mathsf{L}^*_{\mathsf{Kh}_i}\text{-bisimulation between } \mathcal{M} \text{ and } \mathcal{M}'\}.$$

Lemma 4.3. *Deciding* `CheckKhiBisim` *is in* coNP.

Proof. To prove that `CheckKhiBisim` is in coNP we will give a non-deterministic algorithm for its complement. First, we guess a *certificate* that acts as a counterexample. In our case, the certificate will be of the form $x = \langle X, b\rangle$, where X is a subset of states from one of the models, and b is a bit indicating to which model we are referring to. Notice that $|x| \leq f(|\mathcal{M}| + |\mathcal{M}'| + |Z|)$ for some polynominal f. Then, we need to verify that x is a counterexample for `CheckKhiBisim`, i.e., to provide an algorithm $\mathbb{A}$ s.t. Z is not an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ iff $\mathbb{A}(x) = 1$.

Algorithm 1 Counterexample Verifier

1: **function** NotSimulates($\mathcal{M}, \mathcal{M}', Z, X$)
2:   **for all** $i \in \mathrm{Agt}$ **do**
3:     **for all** $\pi \in \mathrm{U}(i)$ **do**
4:       **if** $X \subseteq \mathrm{SE}^{\mathcal{M}}(\pi)$ **then**
5:         $T, X' \leftarrow \mathrm{R}_\pi(X), Z(X)$
6:         $T' \leftarrow Z(T)$
7:         foundWitness ← False
8:         **for all** $\pi' \in \mathrm{U}'(i)$ **do**
9:           **if** $X' \subseteq \mathrm{SE}^{\mathcal{M}'}(\pi')$ and $\mathrm{R}'_{\pi'}(X') \subseteq T'$ **then**
10:             foundWitness ← True
11:         **if** not foundWitness **then return** 1
12:   **return** 0
13: **function** CounterExCheck($\mathcal{M}, \mathcal{M}', Z, \langle X, b\rangle$)
14:   **if** b **then**
15:     **if** exists $[s] \in \mathfrak{C}_{\mathcal{M}}$ s.t. $[s] \nsubseteq X$ and $[s] \cap X \neq \emptyset$ **then**
16:       **return** 0
17:     **return** NotSimulates($\mathcal{M}, \mathcal{M}', Z, X$)
18:   **else**
19:     **if** exists $[s'] \in \mathfrak{C}_{\mathcal{M}'}$ s.t. $[s'] \nsubseteq X$ and $[s'] \cap X \neq \emptyset$ **then**
20:       **return** 0
21:     **return** NotSimulates($\mathcal{M}', \mathcal{M}, Z^{-1}, X$)

Checking conditions (Atom), (A-zig) and (A-zag) can clearly be done in polynomial time w.r.t. the number of states of the models, and any certificate works as a counterexample. Thus, the crucial part is to check the conditions (Kh_i-zig*/zag*). The function CounterExCheck in Alg. 1 checks if (Kh_i-zig*) is satisfied over the set X playing the role of P in Def. 3.4 ((Kh_i-zag*) is obtained by inverting the parameters in line 21, whenever $b = 0$). The condition in line 15 (resp. line 19) requires to computing $\mathfrak{C}_{\mathcal{M}}$ (resp. $\mathfrak{C}_{\mathcal{M}'}$), which can be done in polynomial time on the number of states of the respective model. Then, the rest of the solution relies on invoking the function NotSimulates, which verifies if the given input is a counterexample. The most critical part is in lines 4, 5 and 9, but as stated in [4], all these can be done in polynomial time. Correctness of Alg. 1 follows since it simply verifies the conditions from Def. 3.4. □

To establish a matching lower bound, we reduce the problem of checking whether a propositional formula φ in DNF is a tautology, known to be coNP-complete (see [7]). The strategy is to create two models that are identical, except that an additional action *test* can be potentially executed in only one of them, and such an action can be mimicked in the other model if and only if φ is a tautology. The sets P and P' of Def. 3.4 will encode possible valuations for φ.

Lemma 4.4. *Deciding* `CheckKhiBisim` *is* coNP*-hard.*

Proof. The proof proceeds by reduction of the problem below:

$$\texttt{DNF-TAUT} \overset{\text{def}}{=} \{\varphi \mid \varphi \text{ is a propositional tautology in DNF}\}.$$

Let us define $f : \texttt{DNF-TAUT} \longrightarrow \texttt{CheckKhiBisim}$ as the following mapping between instances of the respective problems. Let $\varphi = \varphi_1 \vee \ldots \vee \varphi_m$ be a propositional formula in disjunctive normal form (DNF), and let $p_1, \ldots, p_n \in \text{Prop}$ those symbols that appear in φ. Then, f maps φ to $\langle \mathcal{M}, \mathcal{M}', Z\rangle$, where $\mathrm{Agt} \overset{\text{def}}{=} \{agt\}$ and:

- $\mathcal{M} \stackrel{\text{def}}{=} \langle \text{S}, \text{Act}, \text{R}, \{\text{U}(agt)\}, \text{V}\rangle$,
- $\mathcal{M}' \stackrel{\text{def}}{=} \langle \text{S}, \text{Act}, \text{R}, \{\text{U}'(agt)\}, \text{V}\rangle$, where
- $\text{S} \stackrel{\text{def}}{=} \{s_i \mid i \in \{1, ..., n\}\} \cup \{t_j \mid j \in \{1, ..., m\}\} \cup \{e\}$,
- $\text{Act} \stackrel{\text{def}}{=} \{in_{\varphi_j} \mid j \in \{1, ..., m\}\} \cup \{out_{\varphi_j} \mid j \in \{1, ..., m\}\} \cup \{test\}$,
- $\text{R}_{in_{\varphi_j}} \stackrel{\text{def}}{=} \{(s_i, t_j) \mid p_i$ does not appear in negative form in $\varphi_j\}$,
- $\text{R}_{out_{\varphi_j}} \stackrel{\text{def}}{=} \{(t_j, s_i) \mid p_i$ appears in positive form in $\varphi_j\} \cup \{(t_j, e) \mid \varphi_j$ does not contain positive symbols$\}$,
- $\text{R}_{test} \stackrel{\text{def}}{=} \{(s_i, s_i) \mid i \in \{1, ..., n\}\} \cup \{(s_i, e) \mid i \in \{1, ..., n\}\}$,
- $\text{R} \stackrel{\text{def}}{=} \{\text{R}_{in_{\varphi_j}}, \text{R}_{out_{\varphi_j}} \mid j \in \{1, ..., m\}\} \cup \{\text{R}_{test}\}$,
- $\text{U}'(agt) \stackrel{\text{def}}{=} \{\{in_{\varphi_j} out_{\varphi_j}\} \mid j \in \{1, ..., m\}\}$,
- $\text{U}(agt) \stackrel{\text{def}}{=} \text{U}'(agt) \cup \{\{test\}\}$, and
- let $q_1, ..., q_{n+m+1} \in$ Prop be $n + m + 1$ different propositional variables, we define: $\text{V}(s_i) \stackrel{\text{def}}{=} \{q_i\}$ for all $i \in \{1, ..., n\}$; $\text{V}(t_j) \stackrel{\text{def}}{=} \{q_{j+n}\}$, for all $j \in \{1, ..., m\}$, and $\text{V}(e) = \{q_{n+m+1}\}$,
- $Z \stackrel{\text{def}}{=} \{(s, s) \mid s \in \text{S}\}$.

We need to prove that φ is a propositional tautology iff Z is an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. Notice that the sets of states of $\mathcal{M}$ and $\mathcal{M}'$ coincide, so whenever we have sZt, s is seen as a member of $\mathcal{M}$ while t is a member of $\mathcal{M}'$. W.l.o.g., we assume that every valuation making φ false assigns 1 to some of the variables (otherwise, f can be easily modified to handle this case). It is clear that $|\mathcal{M}| + |\mathcal{M}'| + |Z|$ is polynomial w.r.t. the size of φ.

⇒) Suppose $\varphi = \varphi_1 \vee \ldots \vee \varphi_m \in$ DNF-TAUT and let $p_1, ..., p_n \in$ Prop those symbols that appear in φ. Let us check that Z is an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$, with $f(\varphi) = \langle \mathcal{M}, \mathcal{M}', Z\rangle$. If sZs', then Z satisfies (Atom), because by definition $s = s'$. Conditions (A-zig/zag) are also directly satisfied. For (Kh_i-zag*) note that every witness in $\mathcal{M}'$ can be mimicked on $\mathcal{M}$, then the condition directly holds. Let us see that Z satisfies (Kh_i-zig*).

Before proceeding, notice that $Z(X) = X$, for all $X \subseteq \text{S}$. Take then $P, T \subseteq \text{S}$ s.t. for all $[s] \in \mathfrak{C}_{\mathcal{M}}$ we have $P \cap [s] = \emptyset$ or $[s] \subseteq P$, and s.t. $P \stackrel{agt}{\Rightarrow} T$. We need to show that exists $T' \subseteq \text{S}$ s.t.: 1) $Z(P) \stackrel{agt}{\Rightarrow} T'$ and 2) $T' \subseteq Z(T)$. It is worth noticing that if $\{\sigma\}$ is a witness of $P \stackrel{agt}{\Rightarrow} T$ of the form $\sigma = in_{\varphi_j} out_{\varphi_j}$, for $1 \leq j \leq m$, we can take $T' = T$ and the same $\{\sigma\}$ as witness of $Z(P) \stackrel{agt}{\Rightarrow} T'$. Then, suppose that $\sigma = test$. By definition of R, $P \subseteq \text{SE}^{\mathcal{M}}(test)$ only if $P \subseteq \{s_1, \ldots, s_n\}$. Since $\text{R}_{test}(P) = P \cup \{e\}$, then necessarily $P \cup \{e\} \subseteq T$ (†).

Consider now the valuation $v : \text{Prop} \to \{0, 1\}$ s.t. $v(p_i) = 1$ for all i s.t. $s_i \in P$, and $v(p_i) = 0$ for all i s.t. $s_i \in \{s_1, \ldots, s_n\} \setminus P$. Since φ is a tautology, there is some φ_j s.t. $v \Vdash \varphi_j$ (here, $\Vdash$ represents the standard satisfiability relation in propositional logic, see e.g. [35]). Thus, if a propositional variable p_i appears positively in φ_j, then $s_i \in P$, whereas if p_i appears negatively in φ_j then $s_i \in \{s_1, \ldots, s_n\} \setminus P$. As a consequence of the latter, $P \subseteq \text{SE}^{\mathcal{M}'}(in_{\varphi_j} out_{\varphi_j})$, and moreover, $t_j \in \text{R}_{in_{\varphi_j}}(P)$. Additionally, it holds that $(t_j, u) \in \text{R}_{out_{\varphi_j}}$ iff, either $u = s_k$ for k s.t. p_k appears in φ_j positively, or $u = e$, if φ_j contains no propositional symbols in positive form. Thus, necessarily $\text{R}_{in_{\varphi_j} out_{\varphi_j}}(P) \subseteq P \cup \{e\} \subseteq T$ (using also †). Then, $T' = \text{R}_{in_{\varphi_j} out_{\varphi_j}}(P)$ works to show that $Z(P) \stackrel{agt}{\Rightarrow} T'$, and furthermore, that $T' \subseteq T = Z(T)$. Hence, Z satisfies (Kh_i-zig*).

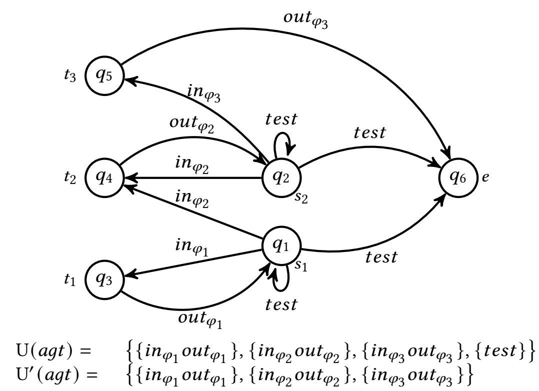

Figure 2: Graphical representation of $\mathcal{M}$ and $\mathcal{M}'$.

⇐) We will prove the contrapositive, i.e., that $\varphi = \varphi_1 \vee \ldots \vee \varphi_m \notin$ DNF-TAUT implies Z is not an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. Let v be the valuation s.t. for all $1 \leq j \leq m$, we have $v \nVdash \varphi_j$. Let $V^+ = \{p_i \mid v(p_i) = 1\}$, and $P^+ = \{s_i \mid p_i \in V^+\}$. As mentioned earlier, we can assume w.l.o.g. that $P^+ \neq \emptyset$. We will check that Z fails to satisfy (Kh_i^*-zig).

Since $P^+ \subseteq \{s_1, \ldots, s_n\}$, we have that $P^+ \subseteq \text{SE}^{\mathcal{M}}(test)$ and $\text{R}_{test}(P^+) = P^+ \cup \{e\}$, thus $P^+ \stackrel{agt}{\Rightarrow} P^+ \cup \{e\}$. We need to show that there is no T' s.t. 1) $Z(P^+) \stackrel{agt}{\Rightarrow} T'$ and 2) $T' \subseteq Z(T)$. Notice that in $\mathcal{M}'$, the only possible witness for 1) is of the form $\{in_{\varphi_j} out_{\varphi_j}\}$, for $1 \leq j \leq m$. Notice that since for all j, $v \nVdash \varphi_j$, then either: (A) exists $p_i \in V^+$ that appears negatively in φ_j, or (B) exists $p_i \in \{p_1, \ldots, p_n\} \setminus V^+$ that appears positively in φ_j. Let us analyze each case. If (A) holds, then that $p_i \in V^+$ appears negatively in φ_j implies that $s_i \notin \text{SE}^{\mathcal{M}'}(in_{\varphi_j} out_{\varphi_j})$. Thus such a plan cannot be a witness for finding a proper T'. Otherwise, if (B) holds, that $p_i \in \{p_1, \ldots, p_n\} \setminus V^+$ appears positively in φ_j implies that $s_i \in \text{R}_{in_{\varphi_j} out_{\varphi_j}}(P^+)$. Since $s_i \notin P^+$, we have $\text{R}_{in_{\varphi_j} out_{\varphi_j}}(P^+) \not\subseteq P^+ \cup \{e\} = Z(P^+ \cup \{e\})$. Again, such a plan is not appropriate to find a suitable T'. As we analyzed all possible cases, we conclude that Z does not satisfy (Kh_i^*-zig), thus Z is not an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. □

For instance, for the formula $\varphi = (p_1 \wedge \neg p_2) \vee (p_2) \vee (\neg p_1)$, the mapping $f(\varphi)$ results in models $\mathcal{M}$ and $\mathcal{M}'$ as indicated in Fig. 2. It is easy to check that $\varphi \in$ DNF-TAUT, and that Z as in Lemma 4.4 is an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation between the obtained models.

Theorem 4.5. *Deciding* CheckKhiBisim *is* coNP*-complete.*

Proof. It follows from Lemmas 4.3 and 4.4. □

The next result states that, to check whether two LTS^{U} are bisimilar, we can run the defined algorithm with a particular Z.

Lemma 4.6. *Let $\mathcal{M}$ and $\mathcal{M}'$ be LTS^{U}s s.t. $\mathcal{M} = \langle \text{S}, \text{Act}, \text{R}, \text{U}, \text{V}\rangle$ and $\mathcal{M}' = \langle \text{S}', \text{Act}', \text{R}', \text{U}', \text{V}'\rangle$. $\mathcal{M} \leftrightarrows^* \mathcal{M}'$ implies that $\mathfrak{B} \stackrel{\text{def}}{=} \{(t, t') \in \text{S} \times \text{S}' \mid \text{V}(t) = \text{V}'(t')\}$ is an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$.*

Proof Sketch. Suppose Z is an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. We need to show that all the conditions from Def. 3.4 holds for $\mathfrak{B}$. By definition, $\mathfrak{B}$ satisfies (Atom). Since Z satisfies (Atom), we get $Z \subseteq \mathfrak{B}$. Then, (A-zig/zag) follows for $\mathfrak{B}$.

To prove (Kh_i-zig*), let us take $P, T \subseteq \mathrm{S}$ as in Def. 3.4, and show that there exists T' s.t.: 1) $\mathfrak{B}(P) \overset{i}{\Rightarrow} T'$ and 2) $T' \subseteq \mathfrak{B}(T)$. Since Z is an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation, there is T'' s.t. 1) $Z(P) \overset{i}{\Rightarrow} T''$ and 2) $T'' \subseteq Z(T)$. We will show that we can take $T' = T''$. It is clear that since $Z \subseteq \mathfrak{B}$ and $T'' \subseteq Z(T)$, then $T'' \subseteq \mathfrak{B}(T)$. Thus, we only need to prove that $\mathfrak{B}(P) \overset{i}{\Rightarrow} T''$. But by definition of $\mathfrak{B}$, and since Z is a bisimulation, we can show that $\mathfrak{B}(P) = Z(P)$, concluding the proof. The condition (Kh_i-zag*) can be shown analogously. □

Corollary 4.7. *Deciding* KhiBisim *is* coNP*-complete.*

Proof. The problem can be decided in coNP by Lemmas 4.3 and 4.6, invoking CounterExCheck with $Z = \mathfrak{B}$. Furthermore, coNP-hardness follows by Lemma 4.4. □

As a by-product, it can be shown that the problem of checking whether two pointed $\mathrm{LTS}^{\mathrm{U}}$ are $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimilar is also coNP-complete.

Corollary 4.8. *Deciding* PKhiBisim *is* coNP*-complete, where*

$$\texttt{PKhiBisim} \overset{def}{=} \{\langle \mathcal{M}, s, \mathcal{M}', s' \rangle \mid \mathcal{M}, s \,\underline{\leftrightarrow}^* \mathcal{M}', s'\}.$$

Proof. The result follows by reducing the problem KhiBisim, and by running Alg. 1 and checking if two states belong to $\mathfrak{B}$. □

We finish this section with a last result that is a consequence from our previous developments. In the finite case, two models are not $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimilar *iff* there is a formula that distinguishes them. Actually, we can show that Alg. 1 enables us to build such a formula. Let 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 $\varphi_X \overset{\text{def}}{=} \bigvee_{[s] \in \mathfrak{C}_{\mathcal{M}}, [s] \subseteq X} \varphi_{[s]}$, where $\varphi_{[s]} \overset{\text{def}}{=} \bigwedge_{p \in \mathrm{V}(s)} p \wedge \bigwedge_{p \in \mathsf{Prop}' \setminus \mathrm{V}(s)} \neg p$. With this at hand, we can build a formula distinguishing the models.

Theorem 4.9. *For finite $\mathcal{M}, \mathcal{M}'$, if $\mathcal{M}, s \,\not\underline{\leftrightarrow}^* \mathcal{M}', s'$, then there is a polynomial size $\mathsf{L}_{\mathsf{Kh}_i}$-formula φ s.t. $\mathcal{M}, s \models \varphi$ and $\mathcal{M}', s' \not\models \varphi$.*

Proof sketch. We can use Alg. 1 with $Z = \mathfrak{B}$ 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_i-zig*) fails ((Kh_i-zag*) is analogous), take $\varphi = \mathsf{Kh}_i(\varphi_X, \varphi_{T^+})$ where X is the input of the algorithm and $T^+ = \bigcup_{t \in T} [t]$, for T as in line 5. The respective φ does the job on each case. □

5 BISIMULATION CONTRACTION

One of the most common applications of bisimulation is the one of model minimization or model contraction. The idea is, given a model, to obtain one that is smaller than the original one but preserving its properties. Bisimulation is usually a right way to guide this preservation condition. Model contraction is particularly important for model-checking, since the complexity of model-checking algorithms relies on the size of the model. In turn, it is important to determine how to measure the relevant parts of the model and minimize them. We will present some proposals for model minimization, and discuss how they improve on model-checking for $\mathsf{L}_{\mathsf{Kh}_i}$.

5.1 Contraction Via Propositional Valuations

Our first proposal bears resemblance to the ideas introduced to characterize bisimulations between two different models using the relation $\mathfrak{B}$ in Sec. 4. Concretely, we simply collapse those states that share the same propositional valuation. Still, we need to decide how to connect the states on the new model.

Lemma 5.1. *Let $\mathcal{M}$ be an $\mathrm{LTS}^{\mathrm{U}}$, the relation $\mathcal{Z}_{\mathcal{M}}$ is an auto-bisimulation. Moreover, $\mathcal{Z}_{\mathcal{M}}$ is the maximal auto-bisimulation of $\mathcal{M}$.*

Interestingly, by a simple inspection of the valuation function on a model one can obtain already a maximal bisimulation. Maximality holds since any other bisimulation cannot relate more states, as states not related by $\mathcal{Z}_{\mathcal{M}}$ are distinguishable by some propositional symbol, thus violating (Atom). This property is a consequence of the globality in these particular bisimulation conditions. Below, we introduce a model contraction using such a bisimulation.

Definition 5.2. Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V} \rangle$ be an $\mathrm{LTS}^{\mathrm{U}}$, we define its **valuation contraction** as $\mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}} = \langle \mathrm{S}', \mathrm{Act}', \mathrm{R}', \mathrm{U}', \mathrm{V}' \rangle$, where:

- $\mathrm{S}' \overset{\text{def}}{=} \mathfrak{C}_{\mathcal{M}}$, $\mathrm{Act}' \overset{\text{def}}{=} \{a_\pi \mid \pi \in \bigcup_{i \in \mathrm{Agt}} \mathrm{U}(i)\}$,
- $([s], [t]) \in \mathrm{R}'_{a_\pi} \overset{\text{def}}{\Leftrightarrow}$ 1) $[s] \subseteq \mathrm{SE}^{\mathcal{M}}(\pi)$ and 2) $(v, u) \in \mathrm{R}_\pi$, for some $v \in [s]$ and $u \in [t]$,
- $\mathrm{U}'(i) \overset{\text{def}}{=} \{\{a_\pi\} \mid \pi \in \mathrm{U}(i)\}$ (for each $i \in \mathrm{Agt}$), and
- $\mathrm{V}'([s]) \overset{\text{def}}{=} \mathrm{V}(s)$.

There are some interesting aspects of $\mathcal{Z}_{\mathcal{M}}$ to be discussed. Since this bisimulation collapses the maximum number of states in $\mathcal{M}$, it is optimal with respect to this measure. Moreover, the size of $\mathrm{U}'(i)$ above equals the size of the original $\mathrm{U}(i)$, except that now each $\pi \in \mathrm{U}'(i)$ is a singleton containing a plan of length 1. As a side effect, the number of accessibility relations may grow, as they compact the information concerning complex plans that are strongly executable in $\mathcal{M}$. This trade-off is convenient for performing model-checking, since not only the number of states obtained is minimal, but also is the number of visited states needed to execute a plan.

Theorem 5.3. *Let $\mathcal{M}$ be an $\mathrm{LTS}^{\mathrm{U}}$ with $\mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}$ as in Def. 5.2. Then, for all states s, we have $\mathcal{M}, s \,\underline{\leftrightarrow}^* \mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}, [s]$. Moreover, for a finite $\mathcal{M}$, $\mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}$ can be computed in polynomial time.*

Proof. To prove that $\mathcal{M}, s \,\underline{\leftrightarrow}^* \mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}, [s]$, we need to find an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation Z s.t. $sZ[s]$. Define $Z \overset{\text{def}}{=} \{(t, [t]) \mid t \in \mathrm{S}\}$. First, notice that $Z \neq \emptyset$ since $\mathrm{S} \neq \emptyset$. Second, Z satisfies (Atom) directly since $\mathrm{V}'([t]) = \mathrm{V}(t)$. Third, Z also satisfies (A-zig/zag), because it is defined for all t and $[t]$. Then, we focus on (Kh_i-zig*), whereas (Kh_i-zag*) follows by a similar argument. For checking (Kh_i-zig*) let us take $P, T \subseteq \mathrm{S}$ such that for all $[t] \in \mathfrak{C}_{\mathcal{M}}$ either $[t] \cap P = \emptyset$ or $[t] \subseteq P$. Assume also that $P \overset{i}{\Rightarrow} T$. We need to find $T' \subseteq \mathrm{S}'$ s.t. 1) $Z(P) \overset{i}{\Rightarrow} T'$, and 2) $T' \subseteq Z(T)$.

Let us take $T' \overset{\text{def}}{=} \{[s] \mid s \in T\}$. By definition of Z, it is clear that $Z(T) = T'$, which implies 2. We need to show that it satisfies 1. Since $P \overset{i}{\Rightarrow} T$, there is $\pi \in \mathrm{U}(i)$ s.t. $P \subseteq \mathrm{SE}^{\mathcal{M}}(\pi)$ and $\mathrm{R}_\pi(P) \subseteq T$. By definition of $\mathcal{M}_{/\mathcal{Z}_{\mathcal{M}}}$, there is a_π s.t. $\{a_\pi\} \in \mathrm{U}'(i)$. Let $[s] \in Z(P)$. Then, exists $t \in [s]$ s.t. $t \in P$. By hypothesis, $[s] = [t] \subseteq P$, and since $P \subseteq \mathrm{SE}^{\mathcal{M}}(\pi)$, $[s] \subseteq \mathrm{SE}^{\mathcal{M}}(\pi)$. Then, there exists u s.t. $(t, u) \in \mathrm{R}_\pi$. Thus, by definition of R', $([t], [u]) \in \mathrm{R}'_{a_\pi}$, and since a_π

is a single action, $[t] \in \mathrm{SE}^{\mathcal{M}/Z_{\mathcal{M}}}(\{a_\pi\})$, for all $[t] \in Z(P)$. Hence, $Z(P) \subseteq \mathrm{SE}^{\mathcal{M}/Z_{\mathcal{M}}}(\{a_\pi\})$. It remains to be proved that $\mathrm{R}'_{a_\pi}(Z(P)) \subseteq T'$, but this can be directly checked by analyzing the definition of R'. Therefore, we can conclude that $Z(P) \overset{i}{\Rightarrow} T'$ (with witness $\{a_\pi\}$), proving condition 1. To see that $\mathcal{M}_{/Z_{\mathcal{M}}}$ can be computed in polynomial time, it is enough to notice that e.g. $\mathfrak{C}_{\mathcal{M}}$ and $\mathrm{SE}^{\mathcal{M}}(\pi)$ (and thus R'_{a_π}) can be computed in polynomial time. □

Notice also that we can still reduce the number of actions and the size of each $\mathrm{U}(i)$. This can be achieved by collapsing any pair of actions a_π and $a_{\pi'}$ such that $\mathrm{R}'_{a_\pi} = \mathrm{R}'_{a_{\pi'}}$ into a single action. As a consequence only one of these equal relations is maintained, as well as a single class containing the new single action.

5.2 Contractions Via Classical Bisimulations

Here we briefly discuss another possibility to construct a model contraction. The definition we will introduce relies on the notion of bisimulation for the Basic Modal Logic (BML) (see, e.g., [13]). This is remarkable since BML-bisimulation between two models do not imply they satisfy the same $\mathsf{L}_{\mathsf{Kh}_i}$-formulas. However, using BML-bisimulation works fine for auto-bisimulations.

Definition 5.4. Let $\mathcal{M} = \langle \mathrm{S}, \mathrm{Act}, \mathrm{R}, \mathrm{U}, \mathrm{V}\rangle$ be an $\mathrm{LTS}^{\mathrm{U}}$, and let Z its maximal auto-bisimulation for BML, its **BML contraction** is defined as $\mathcal{M}_{\mathrm{BML}} = \langle \mathrm{S}', \mathrm{Act}, \mathrm{R}', \mathrm{U}', \mathrm{V}'\rangle$, where:

- $\mathrm{S}' \overset{\text{def}}{=} \mathrm{S}_{/Z}$, $\mathrm{V}'([s]) \overset{\text{def}}{=} \mathrm{V}(s)$, $\mathrm{U}'(i) \overset{\text{def}}{=} \mathrm{U}(i)$ (for each $i \in \mathrm{Agt}$).
- $\mathrm{R}'_a \overset{\text{def}}{=} \{([s],[t]) \mid \text{ exist } s' \in [s], t' \in [t] \text{ s.t. } (s',t') \in \mathrm{R}_a\}$.

The auto-bisimulation through this notion is not maximal for $\mathsf{L}_{\mathsf{Kh}_i}$, but still useful since efficient methods to compute it already exists, see, e.g., [18, 29]. Moreover, using this notion for model contraction avoids action renaming, since the set of actions of the contracted model is the same as in the original model.

Theorem 5.5. *Let $\mathcal{M}$ be an $\mathrm{LTS}^{\mathrm{U}}$ with $\mathcal{M}_{\mathrm{BML}}$ as in Def. 5.4. Then, for all states s, we have $\mathcal{M}, s \overset{*}{\leftrightarrows} \mathcal{M}_{\mathrm{BML}}, [s]$.*

Proof. To prove this result, it is enough to show that $Z' = \{(s,[s]) \mid s \in S\}$ is an $\mathsf{L}^*_{\mathsf{Kh}_i}$-bisimulation. It is clear that Z' satisfies (Atom), by definition of BML-bisimulations. For checking (Kh_i-zig*) let us take $P, T \subseteq \mathrm{S}$ s.t. for all $[t] \in \mathfrak{C}_{\mathcal{M}}$ either $[t] \cap P = \emptyset$ or $[t] \subseteq P$. Assume also that $P \overset{i}{\Rightarrow} T$. We need to find $T' \subseteq \mathrm{S}'$ s.t. 1) $Z'(P) \overset{i}{\Rightarrow} T'$, and 2) $T' \subseteq Z'(T)$. We can show that $T' = Z'(T)$ does the job. Since $P \overset{i}{\Rightarrow} T$, there is $\pi \in \mathrm{U}(i)$ s.t. $P \subseteq \mathrm{SE}^{\mathcal{M}}(\pi)$ and $\mathrm{R}_\pi(P) \subseteq T$, thus it can be shown that $Z'(P) \subseteq \mathrm{SE}^{\mathcal{M}_{\mathrm{BML}}}(\pi)$ and $\mathrm{R}'_\pi(Z'(P)) \subseteq T'$. Hence, $Z'(P) \overset{i}{\Rightarrow} T'$, as wanted. ($\mathsf{Kh}_i$-zag*) follows similarly. By definition of Z', (A-zig/zag) hold. □

Notice that the result above does not hold in general for other modal logics, for instance for *sabotage logic* (see e.g. [2, 8]). It thus reflects an interesting expressivity feature of $\mathsf{L}_{\mathsf{Kh}_i}$. Even though $\mathcal{M}_{\mathrm{BML}}$ is not the minimal contraction w.r.t. the number of states, Thm. 5.5 enables us to use efficient methods to compute a model that is considerably smaller than the original one. For instance, [29] establishes that for a finite $\mathcal{M}$, computing $\mathcal{M}_{\mathrm{BML}}$ can be done in $O(m * \log(n))$-time, where n and m are resp. the number of states and edges of $\mathcal{M}$.

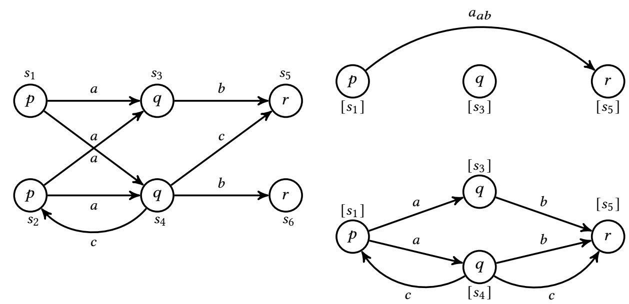

Figure 3: $\mathcal{M}$ on the left, its contractions $\mathcal{M}_{/Z_{\mathcal{M}}}$ (top) and $\mathcal{M}_{\mathrm{BML}}$ (bottom) on the right.

As an example, let $\mathcal{M}$ be the leftmost $\mathrm{LTS}^{\mathrm{U}}$ in Fig. 3, where $\mathrm{U}(i) = \{\{ab\},\{ac\}\}$. The model $\mathcal{M}_{/Z_{\mathcal{M}}}$ is shown at the top right, where $\mathrm{U}'(i) = \{\{a_{ab}\},\{a_{ac}\}\}$, while $\mathcal{M}_{\mathrm{BML}}$ is displayed at the bottom right of Fig. 3. It is worth noticing that $\mathcal{M}_{/Z_{\mathcal{M}}}$ always contains a minimal number of states, whereas this is not the case for $\mathcal{M}_{\mathrm{BML}}$.

6 FINAL REMARKS

We investigated computational aspects of different problems related to bisimulations for the logic $\mathsf{L}_{\mathsf{Kh}_i}$ from [3, 4]. We start by redefining the previously introduced notion of bisimulation, using structural properties of the models only, and provide adequacy results. As a by-product, we show that the definability problem for $\mathsf{L}_{\mathsf{Kh}_i}$ is in P. We then analyze the complexity of different instances of the model-comparison problem, all shown to be coNP-complete. Finally, we provide suitable alternatives for model contraction using bisimulations that can be computed efficiently.

One promising direction for further research involves characterizing the complexity of alternative model-equivalence relations for $\mathsf{L}_{\mathsf{Kh}_i}$. Possible candidates include *simulation equivalence* or *trace equivalence*, adapted to incorporate the notions of plan dependence and strong executability that are central to $\mathsf{L}_{\mathsf{Kh}_i}$. Furthermore, we could explore more advanced model minimization techniques aimed at preserving specific structural characteristics of the model, such as the branching factor or the maximum length of plans.

Finally, we would like to investigate the computational properties of bisimulations for other knowing-how logics. Some of the results presented in this article extend naturally to the setting of *regular* $\mathrm{LTS}^{\mathrm{U}}$ from [16], in which each $\pi \in \mathrm{U}(i)$ can be infinite, but finitely represented by a finite-state automaton. Also, we would like to adapt the general methodologies used herein to handle the linear-plans logic from [36, 37], and other variant logics as in [14, 19, 25].

ACKNOWLEDGMENTS

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, SeCyT-UNC grant 33620230100178CB, and as part of France 2030 program ANR-11-IDEX-0003. We also thank the participants of the DL(R) workshop in Santiago (Chile) for their insightful comments that motivated the investigation of a question leading to Thm. 4.9.

REFERENCES

[1] E. Ábrahám, N. Jansen, R. Wimmer, J.-P. Katoen, and B. Becker. 2010. DTMC Model Checking by SCC Reduction. In *QEST 2010, Seventh International Conference on the Quantitative Evaluation of Systems*. IEEE Computer Society, 37–46.

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

[3] C. Areces, R. Fervari, A. R. Saravia, and F. R. Velázquez-Quesada. 2021. Uncertainty-Based Semantics for Multi-Agent Knowing How Logics. In *Proceedings Eighteenth Conference on Theoretical Aspects of Rationality and Knowledge, TARK 2021 (EPTCS, Vol. 335)*. 23–37.

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

[5] C. Areces, S. Figueira, and D. Gorín. 2011. Using Logic in the Generation of Referring Expressions. In *Logical Aspects of Computational Linguistics - 6th International Conference, LACL 2011 (LNCS, Vol. 6736)*. Springer, 17–32.

[6] M. Arenas and G. I. Diaz. 2016. The Exact Complexity of the First-Order Logic Definability Problem. *ACM Transactions on Database Systems* 41, 2 (2016), 13:1–13:14.

[7] S. Arora and B. Barak. 2009. *Computational Complexity: A Modern Approach* (1st ed.). Cambridge University Press, USA.

[8] G. Aucher, J. van Benthem, and D. Grossi. 2018. Modal logics of sabotage revisited. *Journal of Logic and Computation* 28, 2 (2018), 269–303.

[9] Ch. Baier and J.-P. Katoen. 2008. *Principles of model checking*. MIT Press.

[10] F. Belardinelli, R. Condurache, C. Dima, W. Jamroga, and A. V. Jones. 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*. ACM, 1286–1295.

[11] F. Belardinelli, R. Condurache, C. Dima, W. Jamroga, and M. Knapik. 2021. Bisimulations for verifying strategic abilities with an application to the ThreeBallot voting protocol. *Information and Computation* 276 (2021), 104552.

[12] F. Belardinelli, C. Dima, and A. Murano. 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*. AAAI Press, 425–434.

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

[14] P. F. Castro, P. R. D'Argenio, and R. Fervari. 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*. ijcai.org, 229–239.

[15] M. Dastani and W. Jamroga. 2010. Reasoning about strategies of multi-agent programs. In *9th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2010*. IFAAMAS, 997–1004.

[16] S. Demri and R. Fervari. 2023. Model-Checking for Ability-Based Logics with Constrained Plans. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*. AAAI Press, 6305–6312.

[17] A. Ehrenfeucht. 1961. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae* 49, 2 (1961), 129–141.

[18] J.-C. Fernandez. 1989. An Implementation of an Efficient Algorithm for Bisimulation Equivalence. *Science of Computer Programming* 13, 1 (1989), 219–236.

[19] R. Fervari, A. Herzig, Y. Li, and Y. Wang. 2017. Strategically knowing how. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*. ijcai.org, 1031–1038.

[20] R. Fervari, F. R. Velázquez-Quesada, and Y. Wang. 2022. Bisimulations for Knowing How Logics. *Review of Symbolic Logic* 15, 2 (2022), 450–486.

[21] R. Fraïssé. 1954. *Sur quelques classifications des systèmes de relations*. Ph.D. Dissertation. University of Paris.

[22] S. Ginsburg. 1975. *Algebraic and Automata-Theoretic Properties of Formal Languages*. Elsevier Science Inc., USA.

[23] V. Goranko and S. Passy. 1992. Using the Universal Modality: Gains and Questions. *Journal of Logic and Computation* 2, 1 (1992), 5–30.

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

[25] Y. Li, E. Lorini, and M. Mittelmann. 2025. Rational Capability in Concurrent Games. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 1309–1317.

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

[27] R. Milner. 1989. *Communication and Concurrency*. Prentice Hall.

[28] R. Nissim, J. Hoffmann, and M. Helmert. 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*. IJCAI/AAAI, 1983–1990.

[29] R. Paige and R. E. Tarjan. 1987. Three Partition Refinement Algorithms. *SIAM J. Comput.* 16, 6 (1987), 973–989.

[30] D. Park. 1981. Concurrency and automata on infinite sequences. In *Theoretical Computer Science*, Peter Deussen (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 167–183.

[31] D. Sangiorgi and J. J. M. M. Rutten (Eds.). 2012. *Advanced Topics in Bisimulation and Coinduction*. Cambridge tracts in theoretical computer science, Vol. 52. Cambridge University Press.

[32] J. Urdahl. 2016. *Path Predicate Abstraction: for Sound System-Level Modeling of Digital Circuits*. Ph.D. Dissertation. Kaiserslautern University of Technology, Germany.

[33] J. Urdahl, D. Stoffel, J. Bormann, M. Wedler, and W. Kunz. 2010. Path predicate abstraction by complete interval property checking. In *Proceedings of 10th International Conference on Formal Methods in Computer-Aided Design, FMCAD 2010*. IEEE, 207–215.

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

[35] D. van Dalen. 2012. *Logic and Structure*. Springer London.

[36] Y. Wang. 2015. A Logic of Knowing How. In *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 (LNCS, Vol. 9394)*. Springer, Berlin, Heidelberg, 392–405.

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